

HAKING

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

ANALYZING MALWARE AND MALICIOUS CONTENT

HACKING TRUST RELATIONSHIPS
VIDEOJAKING: HIJACKING IP VIDEO CALLS
AUTOANALYSIS PROBLEMS
SYMMETRIC SECRETS
ICQ FORENSICS
CRACKING SIP
STUDY OF MITM ATTACKS AGAINST SMARTPHONE DEVICES

APPLICATIONS ON THE CD



**CERTIFIED WIRELESS NETWORK ADMINISTRATOR
TRAINING BY SEQRIT.ORG**

DOUBLE ANTI-SPY PRO TRIAL



Vol.5 No.3
Price USD 14.99
Issue 3/2010(28)
ISSN: 1733-7186



PLUS

**THE EVIL TWINS
IDENTITY FRAUD AND PHISHING**
BY JULIAN EVANS

SAINT®

Announcing SAINT 7

Securing your network
just got easier!



SAINT's crisp new interface makes it even easier to use.

- ✓ Integrated vulnerability scanning and penetration testing
- ✓ Payment Card Industry (PCI) Approved Scanning Vendor (ASV)
- ✓ Heterogeneous exploit and vulnerability coverage
- ✓ Security tools module includes e-mail harvesting, social engineering trojan, e-mail forgery, and more

Download a free white paper about integrated vulnerability assessment and penetration testing at www.saintcorporation.com/Hakin9

Contact SAINT's sales team at 1-800-596-2006 x0119 or sales@saintcorporation.com

Dear Readers,

In the times of the growing number of malwares and malicious content not only are our computers at risk but also our mobiles, smartphones and other devices used for accessing the Internet.

That is why hakin9 magazine decided to give a closer look at these attacks. Jeremiah Brott presents a great article on analyzing malware and malicious content. Mayank Aggarwal gives us the study of MITM (Man in the Middle Attack) attacks against Smartphones Devices.

Florian Eichelberger also touches the problem of malwares with the commonly used Auto-Analysis Systems. Ric Messier talks about Session Initiation Protocol and its vulnerability of password cracking.

Thomas Wilhelm starts a series of articles on hacking trust relationships with the focus on information gathering against a target system in order to identify potential trust relationship.

Another interesting article written by Abhijeet Hatekar is devoted to Videojacking and specifically hijacking IP video calls.

As in each issue, Julian Evans discusses the most recent issues from the IT security world. This time he discusses the evil twins: identity fraud and phishing.

Also, I want to thank you for your help and feedback during works on each issue. We value your opinion so keep the mails coming in!

best regards
Karolina Lesinska
Editor-in-Chief

If you are not a HACKER, wanna be HACKER or SECURITY PROFESSIONAL DO NOT READ THIS AD!

LIGATT Security Suites can turn anyone into a computer hacker with out them knowing anything about computer hacking or network security.

There are 5 steps of computer hacking:

Reconnaissance – Where one tries to find out as much information about their target as possible. This usually includes public information. The more information you have, the more you will be able to find and target weaknesses such as: other IP addresses, phone numbers or an email address that could be used for social engineering attacks.

Scanning / Vulnerability – Where the hacker checks for weaknesses (open ports) on your network.

Penetration – Where you will exploit one of the open ports found on your computer or firewall.

Advance – Gaining more access. For instance, the attacker can break into more sensitive administrator root accounts, install backdoors or Trojan horse programs, and install network sniffers to gather additional information.

Covering Tracks – This is the stage where a hacker eliminates any records or logs showing his malicious behavior.



PORT SNITCH

PORTSNITCH takes care of the first two stages of computer hacking, with a few quick mouse clicks. PORTSNITCH not only looks for vulnerabilities on your computer or network, it will perform a public information search for the "Target." The public search includes, but is not limited to:

Facebook.com
Amazon.com
News Searches
Email Name Searches

MySpace.com
Google.com
Blog
Criminal Searches

Youtube.com
Yahoo.com
IP Searches
Pictures Searches



IPSNITCH

IPSNITCH consists of two powerful programs in one. The first powerful program is email spoofing. This allows you to send an email to anyone you'd like and make it appear to have come from someone else.

The second powerful program allows you to get anyone's IP address. With IPSNITCH all you need is an email address of the person in which you are targeting. IPSNITCH lets you send that person an email making the email look like it came from someone else. When a person opens the email, it will automatically text your cell phone and/or email you the person's personal IP address and the ISP that owns the IP address.

* SPOOFNET

SPOOFNET allows you to surf the internet totally anonymously by hiding your IP Address and displaying an IP Address that can't be traced back to you. SPOOFNET is a sophisticated proxy server. Although there are thousands of free Proxy Servers on the market today, they all can't be trusted. As an example, some free proxy servers will capture all the websites you visit as well as all the keys that you type. In other words, some proxy servers can be used as spyware.

TattleTell

TattleTell will notify you by email or text message when an IP address is online or offline. This includes: if the IP address is online or offline, the ISP, and will get a fingerprint of the computer to help identify the suspect's computer.

RECON

RECON is the most advance network security auditing program on the market today. RECON is an active scanner, featuring high-speed discovery, configuration auditing, asset profiling, sensitive data discovery and vulnerability analysis of your security posture. RECON performs network scans using vulnerability check databases based on over 15,000 vulnerabilities. Security audits can take hours to perform. With RECON you can start the audit and move on to other projects or personal time. When the audit is complete it will text you or email you to let you know that the audit is complete.

PC-211

Hand down and thumbs up PC-211 is the most advance penetration testing program on the market. Like other LIGATT Security Suites products, you don't need to know anything about penetration testing. PC-211 uses different techniques to by pass a firewall, IDS and IPS systems. Just like with RECON when the penetration test is complete it will text you or email you to let you know that the audit is complete.

*SPOOFEM

Allows you to call any number in the United States or Canada (other countries coming soon) and have any number show up in the persons caller ID. You can change your voice to male or female, record telephone calls, spoof text messages and spoof emails.

NO SOFTWARE TO DOWNLOAD AND INSTALL

All of the LIGATT Security Suites products and services are web base. That means no matter what operating systems you choose Windows, Mac, Linux or even your web base cell phone, you can use any of our services.

WE PUT OUR MOUTHS WHERE OUR MONEY IS

Unless indicated by a "***", we do not charge you for using any of our services if you do not get any results. As an example, if you use IPSNITCH and we do get the persons IP address you do not pay. If you use PC-211 and it is unable to hack in, you do not pay. You only pay AFTER we get you your results.

LIGATT Security is always adding new services and features.

LIGATT Security International
www.LIGATT.com

* - SPOOFEM is a per minute charge. A Spoof text and email messages are free with an account. SPOOFNET is a pay as you go service.

CONTENTS

HAKIN9 team

Editor in Chief: Karolina Lesińska
karolina.lesińska@hakin9.org
Advisory Editor: Ewa Dudzic
ewa.dudzic@hakin9.org

Editorial Advisory Board: Matt Jonkman, Rebecca Wynn, Rishi Narang, Shyaam Sundhar, Terron Williams, Steve Lape, Peter Giannoulis, Aditya K Sood, Donald Iverson, Flemming Laugaard, Nick Baronian, Tyler Hudak, Michael Munt

DTP: Ireneusz Pogroszewski
Art Director: Agnieszka Marchocka
agnieszka.marchocka@hakin9.org
Cover's graphic: Łukasz Pabian
CD: Rafał Kwaśny
rafal.kwasny@gmail.com

Proofreaders: James Broad, Ed Werzyn, Neil Smith, Steve Lape, Michael Munt, Monroe Dowling, Kevin McDonald, John Hunter, Michael Paydo, Kosta Cipo, Lou Rabom

Contributing editor: James Broad

Top Betatesters: Joshua Morin, Michele Orru, Clint Garrison, Shon Robinson, Brandon Dixon, Justin Seitz, Matthew Sabin, Stephen Argent, Aidan Carty, Rodrigo Rubira Branco, Jason Carpenter, Martin Jenco, Sanjay Bhalerao, Avi Benchimol, Rishi Narang, Jim Halfpenny, Graham Hill, Daniel Bright, Conor Quigley, Francisco Jesús Gómez Rodríguez, Julián Estévez, Chris Gates, Chris Griffin, Alejandro Baena, Michael Sconzo, Laszlo Acs, Benjamin Aboagye, Bob Folden, Cloud Strife, Marc-Andre Meloche, Robert White, Sanjay Bhalerao, Sasha Hess, Kurt Skowronek, Bob Monroe, Michael Holtman, Pete LeMay

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

Senior Consultant/Publisher: Paweł Marciniak
CEO: Ewa Łozowicka
ewa.łozowicka@software.com.pl

Production Director: Andrzej Kuca
andrzej.kuca@hakin9.org

Marketing Director: Karolina Lesińska
karolina.lesińska@hakin9.org

Circulation Manager: Iłona Lepieszka
ilona.lepieszka@hakin9.org

Subscription:
Email: *subscription_support@hakin9.org*

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.hakin9.org/en

Print: ArtDruk *www.artdruk.com*

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


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

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


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

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

To create graphs and diagrams

we used  SmartDraw

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

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

ATTENTION!

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

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.



BASICS

14 Datacenter Storage

RICHARD C. BATKA

One of the biggest challenges storage managers face is to try and calculate just how much storage is needed by that ultra important, mission critical application in the datacenter.



ATTACK

18 Hacking Trust Relationships

THOMAS WILHELM

This article is the first in a series of six, which covers the topic of hacking trust relationships. This first article focuses specifically on Information Gathering against a target system, in order to identify potential trust relationships.

24 Videojacking: Hijacking IP Video Calls

ABHIJEET HATEKAR

Have you ever wondered about the hacking technologies used in Hollywood movies like Ocean's Twelve or The Thomas Crown Affair to steal valuables like diamonds or sculptures? This article will turn you into James Bond for a while by allowing you to hack your own IP Video phones or Cisco Surveillance camera streams. Readers are expected to have working knowledge of VoIP network and their configurations.

30 AutoAnalysis Problems

FLORIAN EICHELBERGER

With thousands of malware samples appearing every day, even Anti-Virus companies no longer have the resources to figure out what each new family is doing, let alone every sample. Although auto-update functionality is wide-spread, having the behavioral pattern of a malware sample aids in its removal. In this article I want to cover commonly used Auto-Analysis Systems and how they can be fooled to raise awareness on this topic and how to treat their output.

34 Analyzing Malware & Malicious Content

JEREMIAH BROTT

Malware, short for malicious software, is a piece of software that's sole purpose and design is to infiltrate or cause damage to a computer system without the owner's well informed consent.

44 **Study of MITM Attacks Against Smartphone Devices**

MAYANK AGGARWAL,
SMOBILE GLOBAL THREAT CENTER RESEARCH ENGINEER

We can foresee a huge growth in Wi-Fi enabled smartphone's in the future. The important question to ask is whether the general public understands the security risks associated with using a smartphone device in an unencrypted wireless hotspot.



DEFENSE

50 **Symmetric Secrets**

TAM HANNA

An article in the last issue of this magazine introduced you to the various forms and applications of cryptography and cryptology. The first topic which usually gets covered is symmetric cryptography – if maths mavens like the JKU Linz's Wolfgang Windsteiger follow this structure, why not adapt it for our personal usage?

56 **ICQ Forensics**

FLORIAN EICHELBERGER

Being around for 14 years, the Mirabilis (now AOL) ICQ Client is probably one of the more widely used Instant Messenger Clients. Having been used by that many people also includes malevolent people and often forensic examiners face the challenge of having to reconstruct the activity's of a user.

62 **Cracking SIP**

RIC MESSIER

The Session Initiation Protocol has been gaining considerably more use in recent years. Voice Over IP has demonstrated both cost-effectiveness and quality such that both consumers and businesses are using it to connect with friends and colleagues worldwide.

66 **Faith in the Format: Unintentional Data Hiding in PDFs**

MATT DAVIS

Adobe's Portable Document Format (PDF) has gained a prominent foothold as a method of distributing text-based and graphic-based information. Its use has become ubiquitous across academic, technical, and governmental institutions and has become one of the forerunners of information dissemination.

REGULARS

08 In Brief

Selection of short articles from the IT security world
ID Theft Protect
Armando Romero &
www.hackerscenter.com

10 ON THE CD

What's new on the latest Hakin9 CD
Hakin9 team

12 TOOLS

CodeScan
HDD Mechanic
Michael Munt

70 Emerging Threats

APT, Google, China and YOU!
Matthew Jonkman

72 ID fraud expert says...

The Evil Twins – Identity Fraud and Phishing
Julian Evans

78 Special Report

Shmoocon 2010 Round-Up
Chris John Riley
Black Hat 2010
James Broad
Security at the Mobile World Congress
Tam Hanna

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 on the Hakin9 website (<http://www.hakin9.org/en>).

48% OF PCS FOUND TO HAVE MALWARE

PWG Phishing Activity Trends Report for Q3 of 2009 has just been released (January 2010). It shows record highs in multiple phishing vectors, but also offers an interesting observation on desktop crimeware infections.

According to the report, the overall number of infected computers (page 10) used in the sample decreased compared to previous quarters, however, 48.35% of the 22,754,847 scanned computers remain infected with malware.

And despite that the crimeware/banking trojans infections slightly decreased from Q2, over a million and a half computers were infected.

DECEMBER MALWARE HITS ONLINE SHOPPING

A leading network security company has highlighted that there is a rise in online fraud driven by software designed to steal financial account data on web sites was timed to hit web sites during the surge in December online shopping. They claimed that malicious software, or malware, activity overall slowed in December to the lowest levels since October.

However, one malware variant – despite its threat only spanning a few days – bucked the trend, delivering 66.5% of total malware activity for December. The Bredolab downloader, a malicious software program that can infect computers by downloading other malicious software designed to steal financial account information, loaded ZBot malware onto infected machines. ZBot malware variants are often configured to steal online banking account information.

In overall malware activity for the month of December, Fortinet detected 157 new vulnerabilities, one-third of which were in active attack mode. It also reported that there were 10 zero-day vulnerabilities.

A zero-day vulnerability refers to a security hole in a software application

for which the software vendor has yet to provide a fix.

Fortinet's report is based on data collected from the company's FortiGate network security appliances and intelligence systems.

Source: Fortinet / ID Theft Protect

NEW PRIVACY LAW FOR FRANCE

The first effects of France's new law against internet piracy will begin to be felt in 2010. The law was passed after a long struggle in parliament, and in the teeth of bitter opposition from groups opposed to internet restrictions. Illegal downloaders will be sent a warning e-mail, then a letter if they continue, and finally must appear before a judge if they offend again.

The law was backed by President Nicolas Sarkozy and the entertainment industry. Its supporters say it is a model for other countries around the world that want to protect their creative industries and make clear to ordinary web-users that not everything is for free.

The law has many opponents, who say either that it is too draconian, or that it has already been overtaken by technology and that serious downloaders will simply sidestep it. But for supporters, it is a long-overdue necessity.

FACEBOOK GROUP SPREADING MALWARE

A false rumour suggesting that Facebook is to start charging is being used to bait malware traps. Thousands of disgruntled punters, angry at the \$4.99 a month charge for using the social networking site that will supposedly kick in from June (or July, according to other false reports) have been induced to visit *protest group* sites in response to spam emails. However, in reality, there is no such plan and the protest pages often contain malware, as urban myth debunking site Snopes warns:

The protest page was a trap for the unwary; clicking on certain elements of

it initiated a script that hijacked users' computers. Some of those who did venture a click had their computers taken over by a series of highly objectionable images while malware simultaneously attempted to install itself onto their computers.

Snopes published its warning on 31 December, but groups on Facebook itself protesting the supposed upcoming charges remain active almost two weeks later. A quick check on one such UK group contains no scripting unpleasantness directly, but it does link to numerous third-party sites whose provenance remains suspect.

Searching for *Facebook charges July 2010* leads to fake blog entries as well as some legitimate results, evidence of an ongoing black hat SEO campaign of a type commonly used to punt rogue security scanner software over recent months.

SOCIAL-NETWORKING SPAM HITS BUSINESS

A survey by a leading UK security vendor has found a 70 per cent jump in spam and malware attacks using social-networking websites in 2009. The survey by UK vendor Sophos found that Facebook topped the list as the perceived riskiest of the major social-networking sites, followed by MySpace, Twitter, and finally LinkedIn.

The Sophos report said that more than 50 percent of the companies surveyed were spammed through a social-networking site last year and that 36 percent were hit by malware from such a site.

Almost half of all the companies surveyed now allow employees open access to Facebook, compared with just 13 percent a year ago. The trick, Sophos said, is not to ban social networking sites but to secure and monitor them to minimize their risks.

MOZILLA PULLS TWO ADD-ON PROGRAMS FROM FIREFOX

Mozilla pulled two programs from its Firefox browser add-on site in January

2010 for containing malware. Sothink Web Video Downloader 4.0 and all versions of Master Filer were found to contain Trojan horse code aimed at Windows users.

In a blog post, Mozilla stated that the Master Filer add-on was able to bypass AMO's security tests. The threat was discovered by Mozilla user CatThief, according to the blog post.

When Mozilla added two more security checks to its vetting process and rescanned its entire catalogue, it discovered that version 4 of the Sothink Web Video Downloader also contained a Trojan horse program.

Sothink Web Video Downloader contained Win32.LdPinch.gen, and Master Filer contained Win32.Bifrose.32.Bifrose.

Source: ID Theft Protect

CHUCK NORRIS BOTNET HITS ROUTERS AND DSL ADAPTERS

Chuck Norris, the famous actor, has unwillingly given his name to a very malicious botnet uncovered by Czech researchers of Masaryk university.

The researchers chose this name after uncovering a comment within the code: *in nome di Chuck Norris*, the Italian version of *in the name of Chuck Norris*.

The botnet is responsible for the theft bank accounts and email credentials.

The botnet spreads by guessing default passwords in DSL Adapters and Routers but also exploiting known vulnerabilities into D-Link devices.

Infection seems to spread from US to China and the authors and their region is still unknown.

BLADE – THE SOLUTION AGAINST DRIVE BY EXPLOITS

Drive by exploits hit the headlines since the Internet Explorer 4-5 vulnerabilities allowing silent download and execution of executables.

Since then, web browsers got more sophisticated at protecting their users.

However drive by exploits are still present on over 5 million web pages on the internet, according to a 2009 research.

Modern drive by exploits target not only browsers but also their addons such as Adobe Flash and Adobe PDF.

In order to stop the plague, that is one of the causes of the quick spread of botnets, researchers at SRI International and Georgia Tech are about to release BLADE (*Block All Drive-By Download Exploits*).

The tool is aimed at blocking any kind of download and warn the user about it. In the tests results, showed by Phil Porras, SRI's program director, Adobe PDF was the most targeted followed by Sun's Java and Internet Explorer.

THE INTERNET UNDER ATTACK

Over 2,500 companies around the world have been compromised according to NetWitness. The cyber attack, one of the largest in internet history, targeted proprietary corporate data, e-mails, credit-card transaction data and login credentials at companies in the health, financial and technology industries in 196 countries.

While the attack is still under study, more and more companies are found out to be fallen victim. Nasa and Pentagon have not been saved.

Once again, corporations and governments prove themselves unable to protect their assets and their vital information.

Malicious software installed through phishing seems to be the root cause of the infection of over 75,000 computers with agent of the Zeus botnet.

This breach, follows the very famous Google breach, a different kind of attack that has caused the jeopardy of intellectual property of over 30 Fortune500.

The latter is in fact a targeted attack imputed to Chinese hackers willingly hacking into U.S. companies.

METASPLOIT NOW INTEGRATES WITH CORE IMPACT

Core Security Technologies has announced the immediate integration of

its commercial automated penetration testing solutions with the most famous open source counterpart.

The integration will allow the IMPACT Pro customers to exploit a system using Metasploit installing the IMPACT Pro Agent and walk the exploitation process from there. Less advanced users will also have the possibility of running Metasploit db-autopwn managing the exploitation process from within Core GUI.

Although there is nothing changing from the Metasploit user's point of view, this news surely highlights that something is going on in the Penetration testing industry. Pentesting tools are now considered necessary auditors means to proof a system security. Metasploit, that was previously acquired by Rapid7, has gained another attestation from the industry and we may see a completely different scenario for this kind of tools in the near future.

NEW CYBER-FRAUDSTERS TO BE JAILED

Edwin Pena, the mastermind behind the largest Voip scam in history has pled guilty and will wait his sentence in jail. Pena had used the help of other hackers, including Robert Moore, who was sentenced 24 months in prison on July 2007.

Another cyber criminal is about to spend four years and eight months in prison. It's the 33-year-old Renukanth Subramaniam, creator of DarkMarket, a website for criminals and fraudsters to exchange credit cards numbers and advices on how to install stealth spywares. The man, coming from Sri-Lanka, is just the last of a series of arrests that now add up to 60 people as part of the investigation on this cybercrime network.

Source: Armadno Romeo

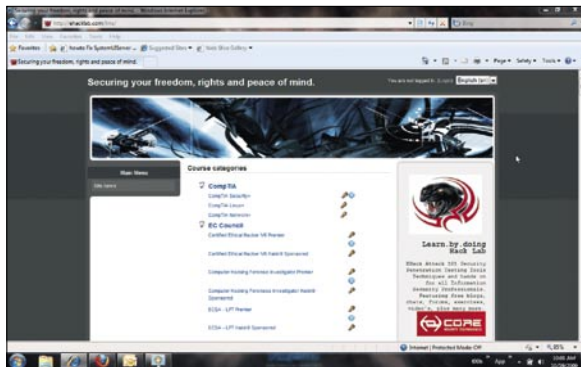
HACKIN9.LIVE

EHACK LAB – LMS ACCESS TUTORIAL

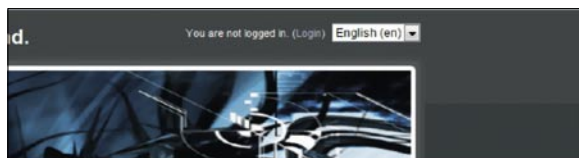
This short tutorial will guide you through creating a new account on the eHack LMS (Learning Management System), show you how to enroll in your courses, and download the prep material.

1. Watch the LMS instruction video on www.tinyurl.com/ehacklablms
2. Using your web browser Navigate to <http://www.ehacklab.com/lms>

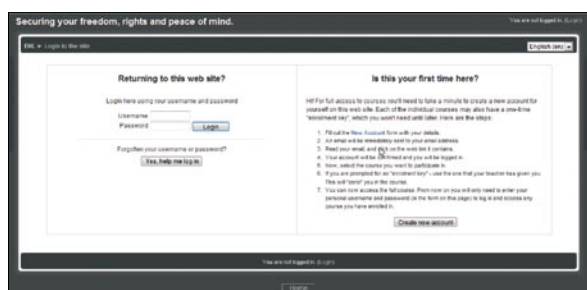
This is the homepage where you can see all the courses we offer, our sponsors, website calendar, as well as news and blog updates.



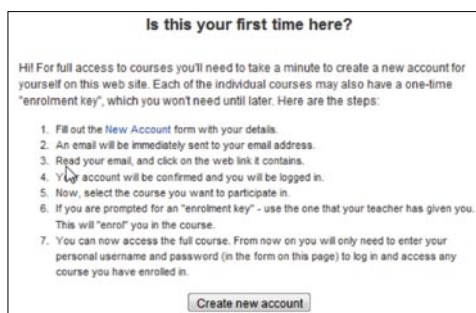
As it states in the top right of the page next to language choice. As you are currently "not logged in."



Proceed to the login page by clicking the (Login) link. This will take you to a new page where you may either login to an existing account or create a new account. In this tutorial, we will be creating a new account to later be enrolled in courses.

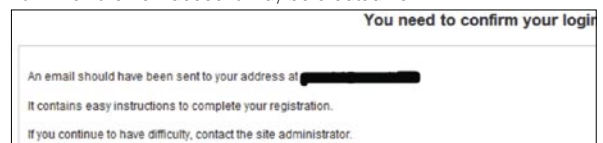


This page will look like this:

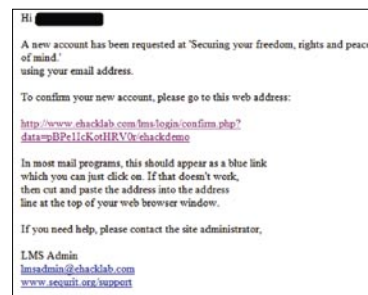


Begin the creation of a new account. Click the "Create new account" button

You are now prompted with a screen asking for some basic account information with which the new account may be created from.



A confirmation link will be sent to the email you specified in your account. You must follow this link and confirm your email before you can access the LMS.



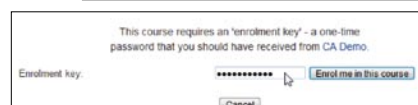
The email looks like this:

Follow the link and you will then be logged into the LMS.

You must now use your enrollment key to register yourself with the course.



Click on the course with which you are to be enrolled. In this case, the CHFI



Enter the enrollment key that was provided by your instructor, and click the "Enroll me in this course" button. You are now successfully enrolled in your course.

You now have access to the LMS, your course, and all the tools and resources you need to begin your course. This tutorial will be followed up with another short tutorial covering the connection to remote attack lab.



IF THE CD CONTENTS CAN'T BE ACCESSED AND THE DISC ISN'T PHYSICALLY DAMAGED, TRY TO RUN IT ON AT LEAST TWO CD DRIVES.



IF YOU HAVE EXPERIENCED ANY PROBLEMS WITH THE CD, E-MAIL:
CD@HAKIN9.ORG

NTFS Mechanic Disk & Data Recovery for NTFS Drives



Items Tested:

40GB External USB HDD that has had an extensive amount of files written to it, and then randomly deleted, approximately 16GB in total and has intermittent connection issues to the point that the local machine doesn't actually register the drive is there.

Once I had the software installed it was time to see how it performs. I plugged the external drive in and then powered up the software. It saw my drive straight away, but it didn't actually state what disk format the drive actually was. This might be due to the fact that the operating system didn't actually find the drive itself, so it was a pleasant surprise that this program did indeed find it.

You are able to configure what types of files you actually want the program to be searching for during the recovery process, for this test I just left everything as default which means everything was selected.

I selected my external USB Drive and it scanned the partitions first to ensure that it can actually see the drive correctly. Once this part of the process has been completed it then requests that you allow it to scan the whole partition that you have selected, this

appears to be a very cpu intensive program so I would suggest to just leave it running on its own if possible. It took just over an hour to scan through a 40GB hard drive. Once it was finished NTFS Mechanic provides all the data that's on the drive, deleted and non-deleted files. You can select in the right hand menu to only see the recovered files, which makes it a lot easier to see what the program has actually found.

If you look at the properties of the files and folders that have been listed as being recovered, you can actually see the prognosis of each file if you decided to proceed and recover the file completely.

The process for recovery couldn't be much easier, it's simply a case of going through the folder list and selecting the files you want to recover and then just say where you want them to be stored.

The program performs really well and managed to recover data from a disk that hasn't been seen by any of my machines for a little while now which quite impressed me.

I noticed that there were a few areas within the program that could do with some QA work as there were non english characters in use and some screens weren't actually needed in my opinion but they aren't detrimental to the product.

I would gladly have this tool in my toolbox.

http://recoverymechanic.com/ntfs_recovery/ntfs_mechanic.php

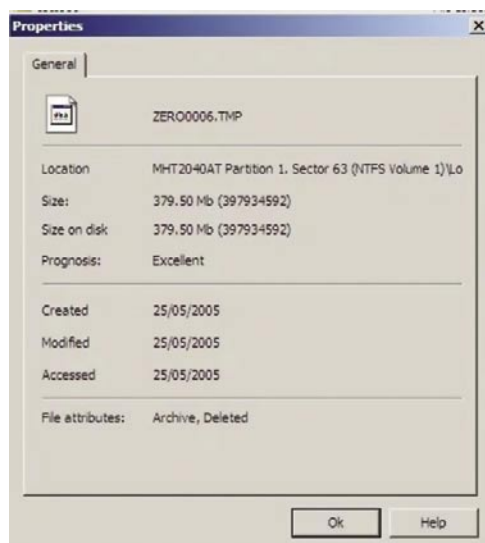
Pricing

Standard \$99.95

Business \$199.95

Professional \$299.95

Prices are in US Dollars



CodeScan



CodeScan is a source code analysis tool, that will allow you to scan your code and then produce detailed reporting on all the vulnerabilities that are found in your code. By scanning and repairing your code throughout your project, so long as you follow the recommendations from the reports, you should be releasing secure code from the outset. Not as a bugfix or an update *after* someone has found the vulnerability.

Once it was installed, you need to ensure that you have a permanent connection to the Internet when your trying to use it, as it is based on the SaaS (*software as a service*) system, and will call home each time it is used. This is so that it can download any updates that may have been released, you dont have a choice on this as I found out, the program refuses to run without it.

Configuring a scan couldnt be easier, just enter in the details for the project and then select the location of the code to be scanned. As you browse to the code's location, you can specify exactly how many pages you wish to have scanned. In using the CodeScan Combo, I am able to select the actual page type I want to use, in this instance I choose php.

Its then a simple case of clicking on the scan button to start the process off, be aware that if you select are large amount of pages to scan it will take a long time to complete as all the scans are run against each page in succession. One annoying quirk I found was that even though I had specified to scan php pages in the setup of the project scan, it still asks me to confirm this before commencing the scan. However, the producer claims it is an intended step that allows you to select which components to scan.

The reporting provided by CodeScan is very clear, concise and very easy to read (which is ideal for managers). The first page is a summary of the scan, and clearly shows exactly the vulnerabilities by severity in a



piechart format. You are also presented with a table showing how many of each vulnerability have been found in the code. As you go further into the report you are shown which page and which line actually has the error, it also shows the syntax that is at fault.

Not being a coder myself, I decided to scan a well known local webserver and I was very very surprised at the results as it found a lot of sql injection vulnerabilities within the pages. I can see how this application would be of great use to any software development house, as it wouldnt take much to run a scan at the end of each week to ensure that all the code to date is as secure as it can be and if there is any requirements to be changed they can be made before release.

I did have a need to contact the support, as I was getting some weird crashing but this was due to the low specification of my machine and not down to the application itself. The support was quick and helpful, so your in good hands if you ever get stuck!

Update from Manufacturer

From March 2010, We're releasing a free version of CodeScan. This will be called CodeScan Community Edition and will include all languages and vulnerabilities. This will coincide with the launch of CodeScan Developer 1.9.1 which will become our payware version available for sale on our website at the same price as currently advertised (ASP – \$197, PHP – \$397, ASP.NET – \$797, Combo – \$997). CodeScan 1.9.1 features a Visual Studio Plugin interface for ASP.NET, as well as advanced reporting against OWASP 2007, OWASP 2010, CWE (*Common Weakness Enumeration*) and ASVS Level 1B for all languages. Users who purchase CodeScan Developer 1.9.1 will also be entitled to all future product releases for the duration of their subscription period.

<http://www.codescan.com/>

CodeScan PHP: \$397
CodeScan ASP.NET c#: \$797
CodeScan ASP: \$197
CodeScan Combo: \$997

Version tested 1.9.0



RICHARD C. BATKA

Datacenter Storage

Difficulty



WHAT YOU WILL LEARN

Basics of storage design and implementation strategies associated with data center virtualization

How to deliver innovative storage solutions such as virtual provisioning for storage problems where no clear guideline or policy exists

Enterprise storage possibilities on EMC Symmetrix and EMC Clarion storage platforms.

WHAT YOU SHOULD KNOW

Multi Terabyte [RAW] distributed data storage environments

Hierarchical storage management systems in distributed storage environments

Ability to build/destroy/manage LUNs and Meta devices, map and mask devices and device groups , build and manage zones and zone sets using EMC tools

EMC Control Center, Symmetrix Management Console, SRDF/TimeFinder Manager, Performance Manager, Storage Scope (FLR), and SYMCLI (command language)

Project teaming with cross-functional IT Departments and Business Units

Disaster preparation planning, design, testing and recovery

One of the biggest challenges storage managers face is to try and calculate just how much storage is needed by that ultra important, mission critical application in the datacenter.

Question: True/False. You should allocate space based on anticipated storage growth, mostly in the Terabyte range and likely because you *believe* it will reduce the management expense and application downtime required to add additional storage later on down the road.

Answer: False. This approach almost always results in over-provisioning of storage capacity and increased cost – all the things we want to avoid.

The Top 7 things storage managers want to avoid:

- 1) Small loading docks
- 2) Higher storage cost
- 3) Increased power consumption/cost
- 4) Expensive short term hardware storage cost because of #1
- 5) More cooling & electricity cost
- 6) Additional floor tile/space requirements
- 7) Lower net storage capacity utilization rates

Solution: Virtual Provisioning

The industries response to the common problem of storage allocation is called Virtual Provisioning. Virtual Provisioning can help at every level of the technology enabled enterprise.

The most dramatic effects can be seen at the storage and host level. For now, let's focus on storage size. I will address the host side in another article.

What Is Virtual Provisioning?

Virtual provisioning allows you to improve storage capacity utilization and simplify storage management by presenting an application with more storage than is physically available.

Example: Google Gmail

You have probably already experienced all the benefits of Virtual Provisioning without even knowing it. Have you ever used Google G-Mail? The Gmail application is a classic example of Virtual Provisioning technology in action. Google actually leads all other email service providers by offering almost unlimited email storage to end users.

Storage Hack

Some forward thinking people have even take it to the next level (with 3rd party utilities of course) creating what I call a *smart man's free network attached storage* – SMFNAS by mapping a Gmail account as a pseudo-drive in the O/S (lets say Windows 7) and using the account as a drag-and-drop file system. You can even chain the accounts together to form an effectively unlimited amount of free network storage space.

TIP

Microsoft's Cloud Computing Service and Google's Chrome Operating System will continue to gain momentum in the marketplace – so to will the associated storage and reliability

requirements. We will see the emergence of new solid state storage platforms that support Virtual Provisioning.

Virtual Provisioning: Up Close

In a Virtual Provisioning environment, as storage demands increase, additional physical storage units called blocks are dynamically allocated from a common (shared) pool of drives. This approach reduces management overhead, but also means more efficient use of available storage.

Historically, a host would report actual capacity (physical/allocated) which would mean that the entire amount of physical storage (capacity) would need to be present on day one. This resulted in low levels of utilization.

Fact

Storage Objects in Mirror Are Further Away Than They Appear.

Business groups always over-inflate storage requirements resulting in a lot of waste. This waste occurs at two levels. First in the aggregate gross amount of the storage request and second in the storage tier type required (tier-1 / tier-2 / tier-3). With virtual provisioning, the host's reported capacity is larger than the actual storage space installed on the storage system.

Storage can now be easily allocated without regard for the number of currently available drives.

Tip

Physical storage is assigned to the server in a storage-on-demand fashion from a shared resource pool. It is this pool of storage that is managed by the storage administrator and not the Logical Unit Number (LUN)'s as had previously been the case.

A New Era

Recently three of the biggest players in the enterprise storage/technology space got together (November 3, 2009) and announced the *Virtual Computing Environment (VCE)* coalition with a commitment to help increase

business agility and lower IT, energy and real estate costs through datacenter virtualization. (In other words, they are going to make it easier for you to introduce Virtual Provisioning at the infrastructure, storage and host levels).

Fact

Worldwide spending on datacenter technology infrastructures exceeded \$350 billion last year.

Emc Symmetrix & Clariion

Let's take a look at how two of the most popular storage platforms EMC Symmetrix & EMC Clariion allow you to tackle the datacenter storage problem today.

Emc Clariion

In the Clariion world, a shared storage pool is called a *thin pool*. This is similar to standard RAID groups in that it simply contains a group of disks. There are limitations on the use of thin pools such as the amount per array which is dependant on the array type.

As user capacity within a thin pool increases there is a tipping point – a point where additional capacity needs to be added. Adding a new disk to a storage pool is non-invasive and as thin LUN's are deleted their availability is automatically returned to the pool without administrator intervention.

Fact

Thin LUN's use less physical storage than traditional LUN's.

A storage administrator needs to configure the total storage that the system should see, but the actual physical disk allocated will be a fraction of that amount.

Performance Notes

What are four things we should know to achieve improved performance?

- Thin LUN's are internally striped so performance will frequently fluctuate
- Use thin LUN's for applications that can tolerate some variation in performance [more on that below].

- Think about the number of spindles, types, and sizes of drives used, regardless of the governing storage management platform in place.
- Disk contention is always a problem. If you want predictable performance, set strict performance parameters by using RAID groups and traditional meta-LUN's striped over multiple RAID groups.

Capacity

The systems reported capacity and actual physical storage allocation reported are very different. When thin LUN's are provisioned to a server the actual physical storage used will be shown as 2GB. The first GB is used for metadata associated with the LUN and the remaining 1GB is available for incoming host writes.

Storage Algorithm

The Mapping Service (storage-on-demand algorithm) guarantees that at least a 1GB slice is available at all times. Data is written in 8K chunks and is optimally written to a location on disk (as determined by the service).

Fact

- EMC's current release supports basic thin LUN structures and the ability to expand a pool by adding disk drives.
- Thin LUN's are supported for LUN migration and local replication using SnapView (snapshots and clones).
- Navisphere Analyzer and Navisphere Quality of Service Manager are supported but replication is only supported with Recoverpoint.

I believe we will see additional support for synchronous and asynchronous MirrorView replication as well as SAN Copy. Then we will see support for shrinking thin pools by removing disks, the ability to create and track thin LUN space reservations and the ability to expand and shrink thin LUN's.

Emc Symmetrix

Similar in principle to Clariion thin provisioning, Symmetrix virtual provisioning uses logical devices called

BASICS

Thin Devices which can be used in many of the same ways as Symmetrix devices have been used in the past.

Thin devices do not need to have physical storage completely allocated from the start. Physical storage comes from a storage pool of multiple data devices. As writes come in, a minimum distribution of physical storage is assigned from a pool. When data is read, it is retrieved from the data device that the thin device is bound to. When the storage pool requires more space, additional data devices can be added to existing thin pools.

Tip

Data devices do not equate to physical disks.

A Symmetrix thin storage pool is made up of many data devices which in turn consist of many physical disks.

Tip

Storage allocation operations are performed in small units of storage called *thin device extents* and round robin schemes exist to balance the allocation of data device extents across all of the data devices in the pool that are enabled and have available capacity.

Fact

A thin device extent size is typically 12 tracks, or 768KB.

The initial bind of a thin device to a pool causes one thin device extent to be allocated per thin device. A read from a thin device results in the data being retrieved from the appropriate data device in the storage pool to which the thin device is bound.

It is possible for a thin device to be presented for host-use before all of the reported capacity of the device has been mapped. In other words, the sum of the reported capacity of the thin device can exceed the available storage capacity of the pool – such a configuration is defined as – *oversubscribed*.

Thin devices appear to the host exactly the same as standard devices and are masked to hosts in the same way.

Tip

The most important aspect to thin device performance is around their spread across the backend.

WARNING

ON STARTUP: NO KEYBOARD DETECTED. PRESS ANY KEY TO CONTINUE.

It's important to consider performance and situations where virtual provisioning is not a good solution. Evaluation & testing is critical.

On a Symmetrix, virtual provisioning data becomes simpler as automated wide

striping provides equivalent or potentially better performance with less planning and effort than is required with standard provisioning.

How To Get The Most Out Of Your Storage Environment

Here is a list of some general datacenter storage guidelines to follow. This list is applicable to virtual provisioning technology as a whole and is not specific to any one specific platform.

Guideline #1: Thin Provision Selectively

To improve capacity utilization within file systems, use thin provisioning only when files are not frequently deleted. Many file systems do not efficiently reuse the space associated with deleted files, which reduces the capacity utilization benefits of the technology.

Guideline #2: Performance Requirements

Virtual Provisioning is desirable for applications that can tolerate some performance variability. Some workloads may see performance benefits from wide striping with thin provisioning; however, when multiple thin devices contend for shared spindle resources in a given pool the performance for a given application can be more variable when utilization reaches higher levels.

Guideline #3: Development & Testing Storage Tiers

Thin provisioning can be an effective way to improve ease of use and capacity utilization for lower storage tiers such as test and development.

Guideline #4: Document Storage

Document repositories with rapidly rising capacity requirements can benefit greatly from the improved capacity utilization offered by thin provisioning.

Alternatives

Take time to evaluate alternative storage platforms. While not in the same class, they do exist. Here are some solutions to keep an eye on:

TERMS DEFINED

- **FRAME:** A frame is a large storage system such as the EMC Symmetrix DMX3000, Clarion, or V-MAX (high-end storage system which can now scale up to 2 PB of usable protected storage capacity- amazing!)
- **ARRAY:** A disk storage system which contains multiple disk drives
- **STORAGE ALGORITHM:** A mathematical formula used to guarantee availability of space in a virtually provisioned environment as well as the optimal writing of data to a physical disk. The algorithm is commonly captured in the form of a program running as a service.
- **LUN:** Logical Unit Number
- **SPINDLE:** Inside a hard drive, a spindle holds one or more flat circular disks called platters, onto which the data is recorded

On the 'Net

- Enterprise Storage <http://www.emc.com>
- FreeNAS <http://www.freenas.org>
- NASLite <http://www.serverelements.com>
- OpenFiler <http://www.openfiler.com/>
- NotSoJankJobs <http://notsojankjobs.com>
- Hak5 <http://www.hak5.org> (Episode 515 – Build your own SAN)



eLearnSecurity
Forging security professionals



Storage Alternatives: Freenas

FreeNAS is a free *Network-Attached Storage* (NAS) server that supports all the common formats. For those of us that need to work within the confines of the most restrictive budget \$0.

Storage Alternatives: Naslite

NASLite is a *Network Attached Storage* (NAS) Operating System which was designed to boot from any device (DRIVE/USB). NASLite can turn a dedicated x86 box into NAS server supporting SMB/CIFS, NFS, AFP, FTP, HTTP, and RSYNC. Most common drives and interfaces are supported.

Storage Alternatives: Openfiler

Openfiler is open source storage appliance software offering Unified SAN & NAS protocol suite, high availability cluster failover capability, block-level remote replication for disaster recover all of which is managed via a web-based gui.

Conclusion

In this article we looked at virtual provisioning and some of the most common storage platforms (CLARION & SYMMETRIX) seen in datacenters around the world. The way people think about storage and the governing financials have evolved to a point where implementing virtual provisioning has clear advantages. For those under very tight budgets -fret not- because alternatives do exist. We will see increased enterprise functionality incorporated across all solution classes as the industry evolves. Take time to review your list of applications, associated storage needs, and select good candidates for virtual provisioning.

Richard C. Batka

Richard C. Batka has held various management and engineering positions with Microsoft, PricewaterhouseCoopers, Symantec, Thomson Reuters, and JPMorgan Chase. He has spent the last 17 years devoted to the complex issues of enterprise strategy, application development, security, infrastructure, data management and regulatory compliance. A graduate of New York University he holds numerous industry certifications. Currently, Mr.Batka is the CEO of a privately funded firm that provides strategy and engineering services to select clients. Mr. Batka can be reached at rbusa1@gmail.com.

Online Penetration Testing Course for Professionals

Interactive elearning system
1600 slides
4 hours videos
Hacking Labs on DVD
Reporting & Methodology
Certification

3 domains - 18 modules
Web Application Security
Network Security
System Security
Web 2.0 attacks
SSL Sniffing
Writing Rootkits
Privilege escalation
Advanced Buffer Overflows

www.elearnsecurity.com



THOMAS WILHELM

Hacking Trust Relationships

Difficulty



This article is the first in a series of six, which covers the topic of hacking trust relationships. This first article focuses specifically on Information Gathering against a target system, in order to identify potential trust relationships.

Over the last decade, there has been a visibly increase in interest in hacking and professional penetration testing; methodologies have been refined, books published, and articles written – all done to expand the professionalism and knowledge of how to conduct effective attacks against computer systems and networks.

The number of people performing effective penetration tests have grown over the years, and the profession overall has improved dramatically because of our ability to share knowledge. Most companies have increased their security posture because of a concerted effort within the hacker community to standardize and improve the way risk assessments and penetration tests are performed. There are those who have resisted standardization of how we do business through the development of methodologies, but overall methodologies have moved us away from ad hoc assessments to repeatable processes.

The problem with methodologies, however, is that they do not provide direction when dealing with unknowns. In general, methodologies focus on specific vulnerabilities and guide the penetration test engineer to find them using *best practices*, such as using vulnerability scanners, specific tools, and generalized tests. By following a methodology, we can easily identify both low-hanging fruit and more complex vulnerabilities; however, each step along the way we are

provided *road signs* on how to progress onto the next step. For organizations interested in identifying the more obvious vulnerabilities, current methodologies are a perfect fit.

But what about those vulnerabilities that exist that cannot be detected by scanners and hacker tools? There are two different categories of vulnerabilities that are more difficult to detect and attack – misconfiguration and trust relationships. Misconfiguration can often be attributed to human error, and requires the penetration test engineer to think outside the box, allowing them to try approaches that don't fit the normal way of conducting an attack. Exploitable misconfigurations include improper routing and switching, wrong file or directory permissions, weak passwords, typos in firewall rules, et cetera. When checking for misconfigurations, the penetration test engineer has to assume that something is amiss and check for it.

Hacking trust relationships requires a different approach than hacking misconfigurations or other vulnerabilities. When dealing with trust relationships, the penetration test engineer has to assume the guise of someone (or something) else, and impersonate their actions exactly, and you must out-think your target to infiltrate the target – not just take advantage of mistakes. Therefore, hacking trust relationships can be considered the pinnacle of hacker skills.

WHAT YOU WILL LEARN

How to conduct information Gathering against a single server

Why it is critical to verify findings using multiple hacker tools

WHAT YOU SHOULD KNOW

Basic Linux commands

Traditional hacking tools

How to set up a hacking lab to recreate the scenario

Trust Relationships – An Explanation

In the most generic term, trust relationships involve increased levels of access between two entities. The level of trust can vary, but exists to provide improved functionality and communication between the two entities. The interesting thing about trust relationships is that after they have been exploited, it is difficult for system and network administrators to detect malicious activity.

Let's look at some examples. The trust relationship most of us are exposed to everyday is access to web servers located on the Internet. Whenever you visit a web site, the administrators have set up the server so that you have access to their web application, which serves you their web pages. Access to the server is often strictly limited to the web service only – we do not have access to anything else. Trust may also extend to forms on the page, documents provided, ability to post data, and so on. And since exploited trust relationships hide malicious activity better, we may be able to post malicious code without detection if the system has granted us permission to post data in the first place. The greater the trust relationship, the greater chance a hacker has to exploit the service.

Web servers are the more obvious example, but other trust relationships exist. Access to the inner-workings of a computer through the use of usernames and passwords is another example. For hackers attempting to break in to a system through exploitation of the login trust relationship, there are numerous brute-force tools that are quite effective. However, exploiting web servers and brute force passwords often create a lot of *noise* on the network; with an alert network security team, those two types of remote attacks are easy to detect and stop.

During this series of articles, we will expand on trust relationship attacks to include more than just web-based and remote password attacks.

What we really want to examine is trust relationships between multiple computer systems and users; if we can exploit those types of trust relationships, we have a greater chance of exploiting multiple systems.

The Practice Target

To illustrate and practice techniques of hacking trust relationships, we will use a LiveCD, obtainable at <http://heorot.net/hakin9>, which contains links for both an ISO file and a VMX file. We can burn the ISO image to a disk and boot it in a system within a lab network, or use both files within a virtual network. For this series of articles, I will use VMware;

feel free to use whichever method is most convenient for you. Figure 1 illustrates the network configuration for our lab network. Notice the IP address for our target system – 192.168.2.101; the IP address has been preconfigured, which means the network will need to be modified accordingly to provide connectivity between the attack platform and the target system (see Figure 1).

In Figure 2, I set up my backtrack system with IP address 192.168.2.10, and made sure I had connectivity with my target (192.168.2.101). For safety reasons, the network I use for my hacking lab does not have connectivity to the Internet; any reference or examples

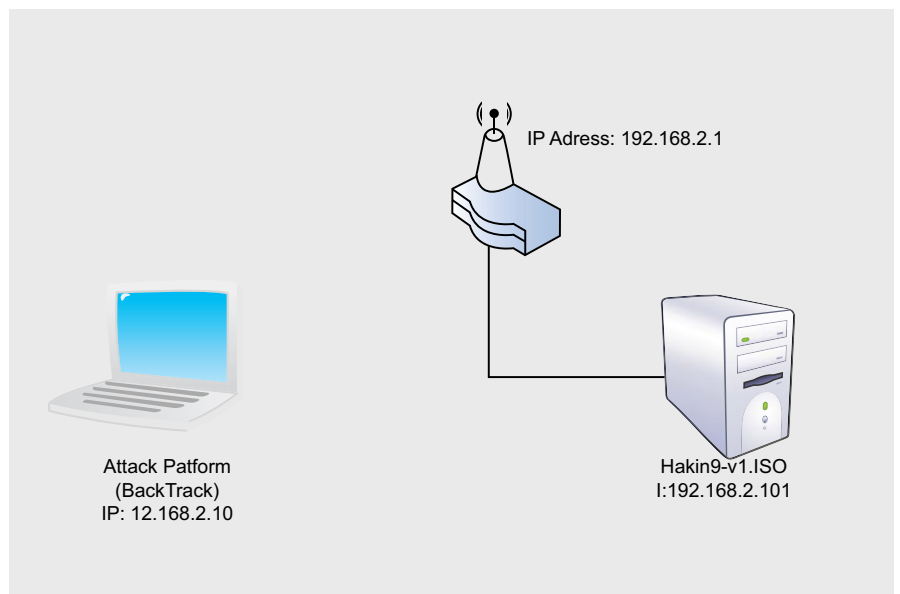


Figure 1. Lab Network Configuration

```

root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@bt:~# ifconfig eth0 192.168.2.10
root@bt:~# ping 192.168.2.101
PING 192.168.2.101 (192.168.2.101) 56(84) bytes of data:
64 bytes from 192.168.2.101: icmp_seq=1 ttl=64 time=2.15 ms
64 bytes from 192.168.2.101: icmp_seq=2 ttl=64 time=0.534 ms
^C
--- 192.168.2.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.534/1.346/2.158/0.812 ms
root@bt:~#
    
```

Figure 2. Configuration of Attack System

where information is gathered off of the Internet is obtained on a separate system connecting to a disparate network.

Once configured, we can begin start Information Gathering of our penetration test. In this stage, all we are attempting to do is understand our target system and gather enough information so that we can identify potential vulnerabilities later. In a real-world penetration test, we would be able to use the Internet to gather information about the corporation, the network, employees and more; however, in our scenario, the only information we can gather is directly off of the target system.

Information Gathering

Now that we have connectivity we can perform a scan. Our first step is to identify potential targets on the network.

If we run the *netdiscover* application, we will see that there is one other system on the network – 192.168.2.101. This confirms our network configuration information as seen in Figure 1. Once we confirmed our network, we can look at our target closer.

I used Nmap to examine which ports were open on our target system, and received the following results: see Listing 1.

I used the `-A` flag during the scan, in order to obtain version information of all running applications and the underlying Operating System. Capturing application data (name and version), and the OS kernel version, will allow us to look for exploits later in the penetration test.

Our Nmap scan provided a significant amount of information for everything except *netbackup* on port 12782. The scan also identified that the FTP service can be

connected to anonymously. At this point in the Information Gathering stage, we can focus our efforts on these two services, since we have sufficient information on the other services to later find exploitable vulnerabilities.

If we attempt to connect to port 13782 using the *telnet* application, we see that the target system refuses our request to connect, as seen in Figure 3. To verify that the port is truly unreachable, we can try and connect with *netcat*; however, the system also refuses that attempt as well.

Other attempts to connect to the system on port 13782 will fail as well – for some reason, the port was available in our initial scan, but is no longer open. At this point, we will have to move on. If we try and connect anonymously to the FTP service on our target system, we meet with success, as demonstrated in figure 4. If we use *anonymous* for both the username and password, we are permitted access to the FTP file system (we can also leave password blank and gain access as well).

If we examine the files in the download section, we are presented with the following file list: see Listing 2.

We can download each document and see if there is any information that would be usable in exploiting this system, such as usernames and passwords. When we do examine them, however, we find that they are useless (please feel free to examine them for yourselves). They are simply security policies – nothing more.

At this point, according to most methodologies, we are probably done with this stage of our penetration test. As a recap, we know:

- Application names and versions (except for port 13782)
- Operating System type and kernel version
- FTP is accessible anonymously and contains PDF files

Anyone with any experience would quickly recognize that we really don't have much to work with. At this point, we are left hoping that one of the services

Listing 1. Nmap Scan of Target System

```
root@bt:~# nmap -A 192.168.2.101

Interesting ports on 192.168.2.101:
Not shown: 996 closed ports
PORT      STATE SERVICE  VERSION
21/tcp    open  ftp      vsftpd 2.0.4
|_ ftp-anon: Anonymous FTP login allowed

22/tcp    open  ssh      OpenSSH 4.3 (protocol 1.99)
|_ sshv1: Server supports SSHv1
| ssh-hostkey: 2048 52:70:f5:39:43:94:6d:62:c5:a0:75:28:d2:55:d1:21 (RSA1)
| 1024 8c:ce:da:b2:29:95:f1:df:48:ac:e3:60:e7:b3:bc:8d (DSA)
|_ 2048 fb:6a:98:30:1d:22:12:31:9f:ab:4a:37:c7:6e:c5:d1 (RSA)

631/tcp   open  ipp      CUPS 1.1
13782/tcp open  netbackup?

MAC Address: 00:0C:29:CD:DA:95 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.24
Network Distance: 1 hop
Service Info: OS: Unix
```

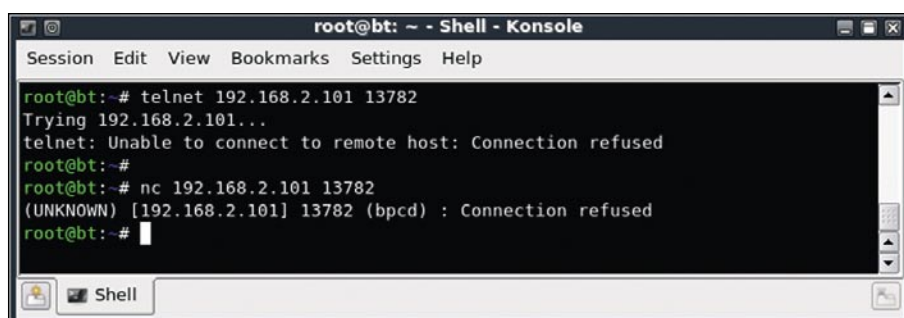


Figure 3. Failed Port 13782 Connection Attempt

have a known exploit that we can take advantage of in the later stages of our penetration test. If this system were part of a large network with a multitude of other servers available, we would move on and try find a system with a higher level of exploitability. In fact, if we ran a Nessus scan against the target, the worse we would receive is security warnings – no identified security holes. The warnings are things we already know, such as FTP allows anonymous access, and SSH is running an older version. Again, perhaps there are some exploits out there; but if there aren't, we would probably move onto other targets.

Since no additional targets are included in this scenario, we have obviously missed something. However, everything we have done is considered standard procedures by most published methodologies. What should we be doing different, then?

Looking for Trust Relationships

Every system has trust relationships, both internally and externally – a system that does not communicate with any other system or its users is worthless. Armed with that knowledge, let's start looking at what trust relationships already exist, and expand on them.

External Trust Relationships

The most obvious external trust relationships are SSH and FTP. FTP may actually have internal trust relationships that we need to explore; but unless there's an exploit, it may be impossible to elevate privileges through FTP. That leaves us with SSH as the only other solid trust relationship we can attack. Later in our penetration test we may conduct a brute force attack against the target system, but until then, we will note it for now since we are still in the information Gathering phase of our penetration test.

When looking for external trust relationships, we have to expect both the obvious and the non-obvious. As mentioned earlier, trust relationships are more difficult to detect and attack; we need to expand our mindset in order to

see trust relationships that others would miss. Honestly, looking for – and exploiting – trust relationships requires a cynical mind; we need to assume that the system administrator is a flawed individual and cannot properly secure their own system. I would not advice saying that to any system administrators directly (especially since I personally started off as a system administrator and know how hard they work), but if we can maintain the mindset that administrators are flawed, we can begin to see things that are often skipped by others.

Cynicism also extends to the tools we use. Which means we must assume Nmap is flawed as well; I know that doesn't sound right, but let's just stick with that thought. In Figure 5, I used netcat to scan the target network. As we can see, there's a new port open – port 6666. Nmap didn't detect it earlier, so either Nmap is flawed or netcat is flawed... or there's something else going on.

If we use the Internet to find out what applications are associated with port 6666, we find everything from IRC to Trojans. Now that we found a new port, we need to try and gather additional information about it. Previously, we

used telnet to try and connect with port 13782; let us see what type of results if we connect to port 6666 instead. Figure 6 indicates that telnet was not able to connect to the port as well. Things are definitely getting curious at this point.

At this point, we have a lot of usable information about our target system, but none of it seems reliable. There are some additional tasks we could perform, now that we have come to the conclusion that the Nmap results earlier were incorrect – or at least not all-inclusive. For example, we may want to use another tool to validate all the version information we received. We may not be able to trust that the Operating System is Linux, or that the kernel version is somewhere between 2.6.13 – 2.6.24. However, that type of investigation will be left to you – the reader – to perform. Hopefully, I have instilled at least a modicum of curiosity regarding our target system.

The Next Step

In the next article in this series, we will move onto the Vulnerability Identification phase of our penetration test. For those readers who might jump ahead,

Listing 2. Available Web Files on Target System

```
-rw-r--r-- 1 1000 513 24627 Jan 14 2008 200611_001.pdf
-rw-r--r-- 1 1000 513 13781 Jan 14 2008 200711_002.pdf
-rw-r--r-- 1 1000 513 79062 Jan 14 2008 200711_004.pdf
-rw-r--r-- 1 1000 513 117946 Jan 14 2008 Acceptable_Use_Policy.pdf
-rw-r--r-- 1 1000 513 255077 Jan 14 2008 Audit_Policy.pdf
-rw-r--r-- 1 1000 513 242960 Jan 14 2008 Email_Policy.pdf
```

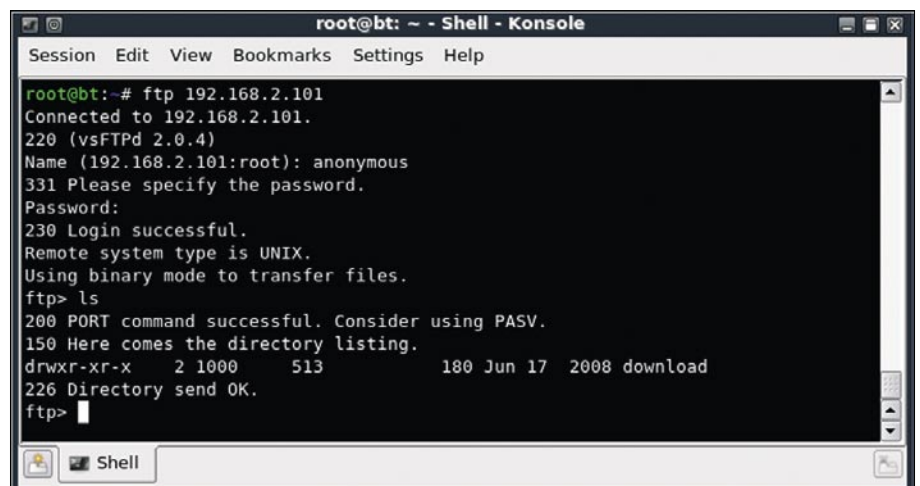


Figure 4. FTP Login Success

remember that what we are attempting to perform in this series of articles is attacks against trust relationships. I encourage you to conduct additional Information Gathering against our target system, so that the task of Vulnerability identification is easier and quicker.

To identify trust relationships, we must always be cynical about the administrator and our results. The key to finding the additional port was to not trust the Nmap scan results. We will find that cynicism will force us to do things during a penetration test that others would never do – think outside the box and try different approaches against a single target. Too often in a penetration test, engineers simply look at the scan results and make a conscious decision to attack or ignore a target on the scan results alone. As we saw in this article, something strange is happening on the target system, and we would have been oblivious to it had we simply accepted our initial Information Gathering results.

If you read my bio, you will see that I am an Associate Professor, which means I cannot conclude this article without assigning homework. As mentioned, in the next article we will be performing Vulnerability Identification against our target; however, the article concluded

without truly understanding the services running on our target. Before the next article, we need to:

- Identify all available services running on the target system. This includes whatever is on port 13782 and port 6666
- Verify version information of all services running on the target system
- Verify the Operating System and kernel version of the target system

The next article will discuss these mystery ports in greater detail, but in the mean time, feel free to conduct your own investigation by downloading the target from <http://heorot.net/hakin9> and attacking it with Nmap, netcat, telnet, and any other tool you want to try out. The web page above will also have a link to the Heorot.net forums; feel free to join the conversation as we move through each article and discuss how to exploit the Trust Relationships of our target system.

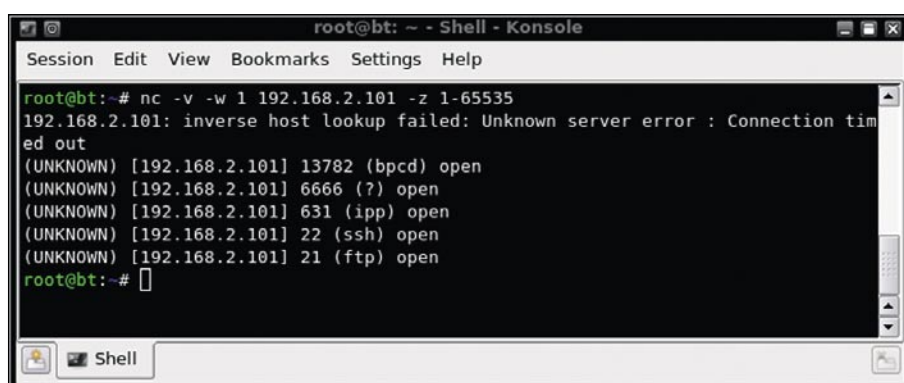
Conclusion

Trust relationships are the most difficult of attack vectors to exploit, but they yield the greatest advantage. When attacking trust relationships, the penetration test engineer has to assume the guise of

someone (or something) else, and impersonate their actions exactly to successfully exploit a system. Other exploits – especially buffer overflows – take advantage of flaws in coding; hacking trust relationships require penetration test engineers to out-think the target, which is truthfully a lot more entertaining when successful.

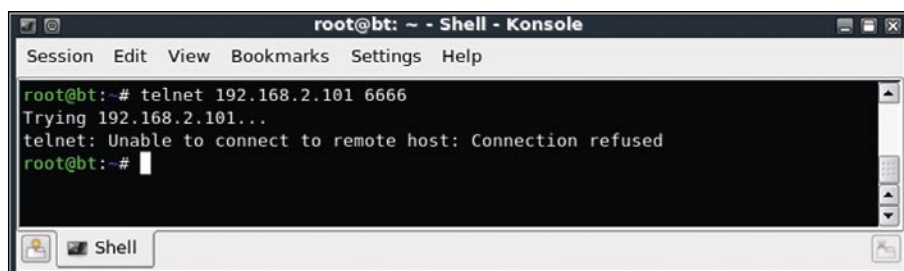
The need to out-think the target obviously includes system and network detection systems, not just the system operators, which adds to the difficulty. Ideally, everything we do when conducting an attack against the target's trust relationships should be seen as legitimate, resulting in our ability to intrude undetected. However, if we are not careful and mimic a valid user or process exactly, we will certainly be detected.

Therefore, to find and exploit trust relationships requires a different mindset than looking for coding flaws. Hacking trust is almost like a chess game when two highly-intelligent foes face each other, attempting to outmaneuver and out-think their opponent. To win the game requires a high level of cynicism and assumption that your opponent is incapable of properly securing their server. With that mindset, new attack vectors present themselves. In the next article, we will use that cynical mindset to identify vulnerabilities within the trust relationships of our target system.



```
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# nc -v -w 1 192.168.2.101 -z 1-65535
192.168.2.101: inverse host lookup failed: Unknown server error : Connection timed out
(UNKNOWN) [192.168.2.101] 13782 (bpcd) open
(UNKNOWN) [192.168.2.101] 6666 (?) open
(UNKNOWN) [192.168.2.101] 631 (ipp) open
(UNKNOWN) [192.168.2.101] 22 (ssh) open
(UNKNOWN) [192.168.2.101] 21 (ftp) open
root@bt:~#
```

Figure 5. Netcat Scan of Target System



```
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
root@bt:~# telnet 192.168.2.101 6666
Trying 192.168.2.101...
telnet: Unable to connect to remote host: Connection refused
root@bt:~#
```

Figure 6. Failed Port 6666 Connection Attempt

Thomas Wilhelm

Thomas Wilhelm has been involved in Information Security since 1990, where he served in the Army for eight years as a Signals Intelligence Analyst / Russian Linguist / Cryptanalyst. A speaker at security conferences across the U.S., including DefCon, HOPE, and CSI, he has been employed by Fortune 100 companies to conduct Risk Assessments, participate and lead in external and internal Penetration Testing efforts, and manage Information Systems Security projects. He currently designs and conducts Hacker training courses and certification boot camps through Heorot.net.

Thomas is also a Doctoral student who holds Masters degrees in both Computer Science and Management. Additionally, he also dedicates some of his time as an Associate Professor at Colorado Technical University, and has contributed to multiple publications, including both magazines and books. His latest contribution was multiple chapters in the Syngress publication titled "Professional Penetration Testing," released in August, 2009, which was his fourth book contribution to Syngress.



```
"><img src=a onerror=alert('yawn')>
```

Let's face it. Checking every web page for cross-site scripting is fun for about four minutes. Then it gets really dull.

Outsource the boredom with Burp Scanner.

<http://portswigger.net>



ABHIJEET HATEKAR

Videojacking: Hijacking IP Video Calls

Difficulty



Have you ever wondered about the hacking technologies used in Hollywood movies like Ocean's twelve or The Thomas Crown Affair to steal valuables like diamonds or sculptures?

How heroes hack into video surveillance camera, fooling security officer by showing fake video from the classified location and escape with valuables? Well if you have thought all these stunts can happen only in movies and only James bond can perform them then, obviously you have mistaken.

Researchers from VIPER Lab have released Videojak: A free security assessment tool which can be used to hijack ongoing video conversation from IP video phones as well as live feed from video surveillance camera making these hacking adventures possible.

This article will turn you into James Bond for a while by allowing you to hack your own IP Video phones or Cisco Surveillance camera streams. Readers are expected to have working knowledge of VoIP network and their configurations.

Statutory warning: Information from this article is strictly for education purpose only. Intercepting and eavesdropping on people's audio/video calls are not legal.

Author cannot be hold responsible for any such actions.

Pre-requisites

- IP Video Phones or Surveillance Camera (CIVS-IPC-4300, AXIS-211a)
- IP PBX
- Linux Box running Videojak
- A small AVI clip

IP Video Phones or Surveillance Cameras

You can build test lab environment using following equipments:

- Cisco 7985G IP Video Phones (Use Cisco proprietary SCCP Protocol and costs approx \$2200)
- Grandstream GX3000 series IP video phones (Use SIP and costs only \$230)
- Polycom Video phones
- Cisco CIVS-IPC-4300 video surveillance cameras (Uses RTSP and costs approx \$1600)
- AXIS Video surveillance cameras (Use RTSP and costs approx \$1000)
- Grandstream Video Surveillance Camera (Use SIP and costs approx \$370)

Videojak can launch attacks against all the above mentioned devices and protocols.

IP PBX

SIPXecs is the open source IP PBX for Unified Communication and is very easy to install. You can download SIPXecs from its official website: <http://www.sipfoundry.org/downloads.html>.

Linux Box running VideoJak

You can use any Linux box to download and compile Videojak on it. However I will recommend

WHAT YOU WILL LEARN

New class of attacks called VideoJacking

At the end of this article users will be able to successfully assess the security posture of their VoIP video solution.

WHAT YOU SHOULD KNOW

Basic networking

familiar with ARP poisoning attacks

VoIP Basics

using a Linux live distribution designed only for VoIP Security Assessments: VAST. It stands for VIPER Assessment Security Tools.

VAST has all the necessary VoIP Security Assessment tools pre-installed along with its dependencies making it plug and play attack base system.

Download Videojak from <http://videojak.sf.net> and VAST from <http://vipervast.sf.net>.

If you choose to install Videojak from source then follow the instruction from the <http://videojak.sf.net/install.html> or else you can install Videojak from a Debian package.

Adjacent network diagram shows replica of my simple Unified Communication Lab used for this attack simulation.

I have configured both Grandstream GXV3000 Phones with IP address 172.16.20.2 and 172.16.20.3 and registered them with SipXecs IP PBX with extension numbers 200 and 201 respectively.

We assume that attacker is also on the same network with IP address 172.16.20.10.

Now that we have our lab ready, we can start our hacks or hacking. There are following three different kinds of attacks supported by Videojak tool.

- Severe Media Denial of Service Attack
- Playing Fake Media Contents Attack
- Media Blackhole

Videojak does *Man-In-The-Middle* (MITM) attack against the whole network to carry out all the above attacks against target devices. Now the question might arise, what if the target network has separate VLANs for voice and data? Videojak is designed in consideration with today's UC infrastructure implementations in which QoS requirements dictate the separation of data and VoIP/Video into discrete networks or VLANs. Videojak has inbuilt VLAN hopping features using which an attacker can – accomplish VLAN hop from data VLAN – to voice VLAN and start the attack.

Media Denial of Service Attack

As the name suggests, this is Denial of Service attack against media. Videojak can launch DoS attack using H.264, H.263 video codecs and G711 alaw, G711µlaw audio codecs.

If target network has separate VLANs – for voice and data then Videojak can be asked to do VLAN hop into voice VLAN before performing *Man-In-The-Middle* attack.

Videojak can auto-discover Voice VLAN either by CDP Sniffing or by CDP Spoofing. If attacker is already aware

of Voice VLAN ID then s/he can directly VLAN Hop into specific Voice VLAN.

Adjacent and screenshots shows the usage of Videojak in CDP Sniff and CDP Spoof mode respectively. Videojak offers extensive help in case you need to explore more command line options.

As our test network does not have VLANs, let's launch Videojak from attacker machine and arp-poison whole 172.16.20.x network using following command.

Once Videojak has arp-poisoned whole network; it starts sniffing for VoIP Signaling and media traffic. Videojak has



Figure 1. Pre-requisites

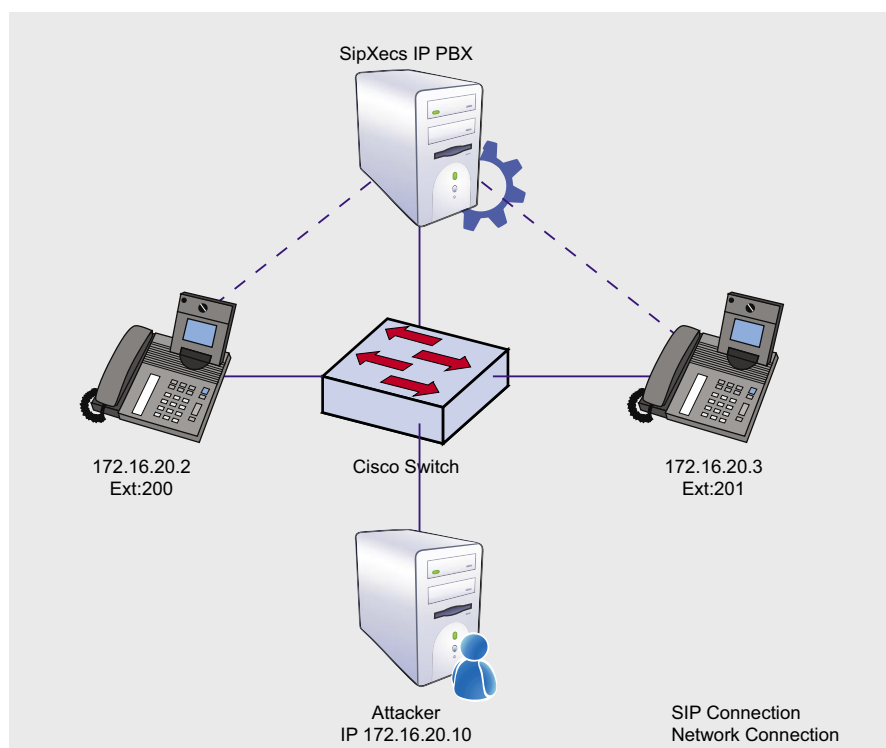


Figure 2. Network Dgm

ATTACK

```
root@chackraview:~/abhijeet# videojak -c 0 -i eth0 // //
videojak 1.06 starting
Parsed 2 entries in Targets file, targets.txt
UCSniff running in CDP Sniff Mode
Capturing CDP Packets on eth0
```

Figure 3. Videojak-cdpsniff

```
root@chackraview:~/abhijeet# videojak -c 1 -i eth0 // //
videojak 1.06 starting
Parsed 2 entries in Targets file, targets.txt
UCSniff running in CDP Spoof Mode
Capturing CDP Packets on eth0
```

Figure 4. Videojak-cdpspoof

```
Inline help:
[VV] - change the visualization mode
[LL] - print the hosts list
[OO] - print the profiles list
[CC] - print the connections list
[AA] - print the Directory Users List
[DD] - print the hosts that have GARP feature disabled
[FF] - plays the video file on the active video conference
[aA] - attack an active video call by sending garbage video payload
[pp] - attack an active video call by dropping video packets
[ss] - stop the video attack, started by p or f switch
[ll] - arp poison the hosts using unicast ARP requests
[rR] - re-arp the hosts using unicast ARP requests
[tt] - print the Targets List
[yY] - print the Active Calls in progress
[<space>] - stop/cont printing packets
[qq] - quit
```

Figure 5. Videojak-help

```
root@chackraview:~/abhijeet# videojak -i eth0 // //
```

Figure 6. Videojak-arp

```
Starting Unified sniffing...
Warning: Please ensure that you hit 'q' when you are finished with this program.
Warning: 'q' re-ARPs the victims. Failure to do so before program exit will result in a DoS.
Mapped new target entry: (IP: 172.16.20.3) --> extension 201:
Mapped new target entry: (IP: 172.16.20.2) --> extension 200:
SIP Video Call in progress. (extension 201, ip 172.16.20.3) calling (extension 200, ip 172.16.20.2)
[+] Press either 'a', 'f' or 'p' from help menu to begin attack against active Call Session.
```

Figure 7. VideoJak-callDetect



Figure 8. Video call in progress

necessary signaling and media dissectors which can detect new SIP and SCCP call setup, call termination, media flow etc.

Let's make a call from Grandstream phone registered with Ext: 201 to Grandstream phone registered with Ext: 200

As soon as we make a call, Videojak detects the ongoing call session between corresponding video phones saying *SIP/SKINNY Video Call in progress* depending on the used signaling protocol and display it on screen. Once the call is detected, you can press *a*, *f*, *p* keys to launch different attacks against ongoing call session.

More information about these attacks can be viewed by pressing *h*. Pressing *h* will launch Inline Help menu of Videojak. Following screenshot shows the inline help menu offered by Videojak.

Reading the menu, we get to know that pressing *a* will attack active video call session by sending garbage video payload. This means Videojak will send malicious audio/video media traffic to one of the video phones involved in an ongoing video call session. This results in the severe DoS attack against the attacked phone and sometimes causing video phones to restart its power cycle.

This attack forces Cisco 7985G IP Video phones to reboot. Adjacent screenshots show the media degradation attack against one of the target video phone from the ongoing video call session.

Playing Fake Media Attack

While pressing *f* will play random video file on the attacked video phone, this is an awesome attack as attacked video phone caller will not get to see the original video from called phone. This attack can be used to play fake contents on the video phone including pornography. Currently Videojak has support for playing only .264 and .avi files.

Attacker can use this play file attack to display old video footage AVI file on video surveillance camera monitor and steal valuables in the background. Let me take you through a Hollywood ride.



Figure 9. Gibberish video screen after media DoS attack



Figure 10. Still from Movie Speed

Remember a scene from the movie *The Speed* in which Jack Traven (Keanu Reeves) with the help of news team loops the old video footage on the hidden video camera while the passengers are safely evacuated.

When we select specific attack to launch, Videojak asks us a bunch of questions related to target video call sessions and target phones. These questions help Videojak to narrow down the target phone as Videojak can detect multiple video call sessions.

As all phones or cameras need different attack settings; the attacker needs to choose target phone/camera vendor

from Cisco, AXIS or other vendors. Other vendors could be Grandstream, Polycom, Cisco CVIC-IPC or any other soft phone with video capabilities E.g. Eyebeam etc.

Let's make another call between Ext: 200 and Ext: 201 and play AVI file on the target phone. Once video call session is detected by Videojak; press *s f* key to enter in play video file attack mode.

Now select the target call session and target phone for the attack. Videojak will display all the available files from the current directory.

Videojak can play video files placed only in current directory. Choose the video file to be played and attack mode. Videojak has two attack modes – One time attack and Infinite attack.

As the name suggests, *One time attack* will play video file on target phone only once, while *Infinite attack* will play it till the user stops attack by pressing [Ctrl]+[C].

Here are the screen shots of play file attack where I have played a clip from the movie *The Matrix* on the target phone. Left hand side image shows short video clip from the movie *The Matrix*, thereby, replacing the live feed from surveillance camera while the right hand side image shows laptop screen respectively on target phone.

Media Blackhole Attack

In Media Blackhole attack, Videojak drops the media packets destined to target phone freezing the video call session.

It can be launched by pressing *p* once Videojak detects the ongoing video call session and can be stopped by pressing *s* when attack is in progress.

Play file and Media Blackhole attacks are also valid and in fact more effective on video surveillance cameras.

Videojak shows how easy it is for an attacker to hijack video phone calls and video surveillance camera which are getting popular these days.

```
[+] Option 1 selected.  Attacking Active SCCP Call Session against
Please select the victim phone/camera model
[1] Cisco 7985
[2] Axis Q1755
[3] Other
Please select attack type
[1] Indefinite looping
[2] One time attack
[+] Please select one the 5 listed file to be played
1.blah.pcap
2.sipdump.pcap
3.H264-media-1.264
4.italianjob.avi
5.replay.avi
Please enter the file index:
```

Figure 11. 3options

On the 'Net

- Download Videojak: <http://videojak.sourceforge.net>
- Download ViperVast: <http://vipervast.sourceforge.net>
- Dynamic ARP Inspection: <http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/19ew/configuration/guide/dynarp.html>
- Download ARPStar: <http://arpstar.sourceforge.net>



Figure 12. Matrix movie clip being played on the target video phone as a part of media play file attack

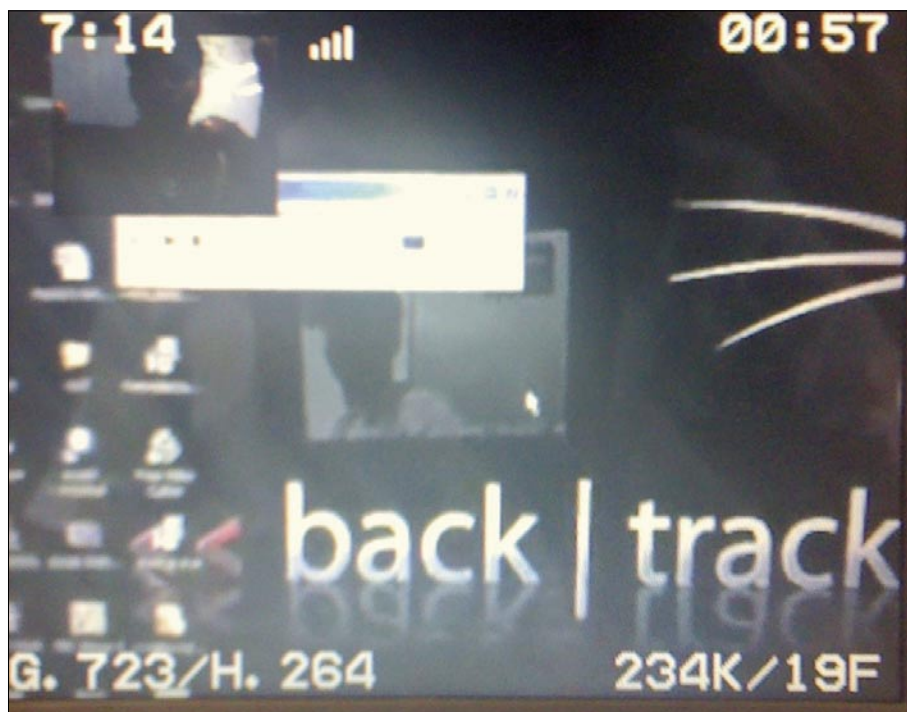


Figure 13. Desktop clip is being played on target phone as part of Videojak file play attack

Mitigation Techniques

You have seen the implications of Videojak over video appliances. Now let's talk about the mediums of mitigations.

VLAN separation will not protect against Videojak. As all the attacks launched by Videojak involve Arp poisoning and *Man-In-The-Middle* (MITM) techniques; having strong layer 2 security will defeat them. Companies can also use encryption technologies like TLS and SRTP which are not enabled by default on the phones and servers.

Arp poisoning can be avoided by enabling Dynamic Arp Inspection on Cisco catalyst switches. Once the Dynamic Arp Inspection is enabled on the switch, interface sending spoofed ARP packets will automatically get disabled, shutting down the attack.

For those who are using Linux box as a router/gateway can use Arp* (ARP star) Linux kernel module which detects and defeats ARP Poisoning attacks.

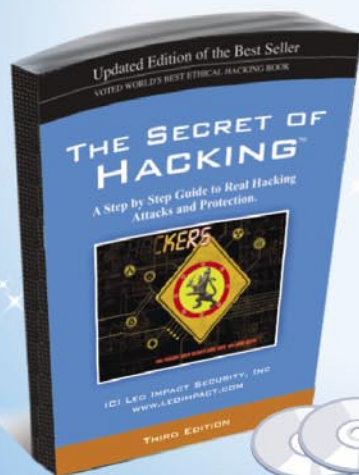
Note: Cisco 7985 IP Video phones cannot be poisoned using regular ARP poisoning techniques. Videojak comes with a feature which makes use of spoofed unicast ARP request to poison them. You can use this feature by pressing *I* from the Videojak inline help menu.

Abhijeet Hatekar

Abhijeet Hatekar works as a Security Researcher in Spera VIPER (Voice over IP Exploit Research) Lab. He is a graduate from University of Pune, India and Author of Videojak (<http://videojak.sf.net>), OAT (<http://voat.sf.net>) and XTest (<http://xtest.sf.net>) VoIP assessment tools. He has published an article on VoIP Security Assessment in January 2010 issue of Hackin9 magazine. Abhijeet has also spoken in Information Security conferences like ClubHack, FRHack 01, SingCERT and can be reachable on abhijeet@viperlab.net, abhijeet@chackreview.net



Want to Be the Best Ethical Hacker & Security Expert?



Latest 2010 Year 3rd Edition Printed Book



1st Edition PDF Version (Free)

New * Third Edition - 10 Times More Powerful

Even the most secure computers are Hackable...

- All E-mail address are Hackable (gmail, rediff, yahoo etc)
- Easily pass CEH (ver6), CHFI, CISSP, CISA Certification.
- Learn How to secure your system and network from Hackers.
- Learn Advanced Hacking (Metasploit, Remote Hacking, NetBanking, Credit Cards, Phone Hacking, Reverse Engineering, VIRUS R&D)



Advantages

- First Edition PDF Version FREE
- Each Topic Cover by Videos
- Easy Language with Email Technical Support
- Access 4000+ Videos Anywhere, anytime
- 2 DVD's FREE
- 30 day money back guarantee
- Secure Payments Via Credit Cards
- No shipping and Hidden cost



Order Now

& **Save 35% + FREE Gifts**

- SMS LEOIMPACT to 54242
- or call 9829944518, 9953244518, 0141-4043404

(between 10:30 to 5:30 pm)



Free Video

Membership & 2 DVD

(including 20,000 Full version
Software related to Hacking, Security and Forensic)

www.theseretofhacking.com



**LEO IMPACT
SECURITY**

LEO IMPACT SECURITY SERVICES PVT LTD
T8, Malyia Apartment, Near BJP Office
C-Schme, Jaipur (Rajasthan) 302001
E-mail: contact@leoimpact.com

Leo Impact Security, INC:
616, Corporate Way, Suite 2
#4000, Valley Cottage, NY 10989
Phone: +1 818 252 9090 (USA)

Educational Institutes can contact for free Ethical Hacking workshop



FLORIAN EICHELBERGER

AutoAnalysis Problems

Difficulty



With thousands of malware samples appearing every day[1], even Anti-Virus companies no longer have the resources to figure out what each new family is doing, let alone every sample.

Although auto-update functionality is widespread, having the behavioral pattern of a malware sample aids in its removal. In this article I want to cover commonly used Auto-Analysis Systems and how they can be fooled to raise awareness on this topic and how to treat their output.

Can you trust the output of those online systems or your Sandboxie[2] software ?

During my current and past jobs I encountered and analyzed an uncountable number of malware samples and lately found some *in the wild* that actively alter their behavior depending if they're detecting the presence of one of those analysis systems.

The two samples I want to explain in more detail show different approaches to produce innocent looking output concealing their malicious nature. The systems those samples have been tested on are:

- CWSandbox[3]
- JoeBox[4]
- Anubis[5]
- Vmware[6]
- Sandboxie

Detections explained

The first sample I want to dissect here was recently detected and handed over to me for

a short analysis. In most of the cases, only the malware itself was captured, missing any installers or exploits that deployed them. The sample showed a very effective way of producing a total innocent-looking output when it was run *as-is* thus defeating the purpose of all tested and later explained systems.

I tried to highlight and comment the Figure directly, so they should speak somehow for themselves. Figure 1 shows the beginning of the trojan horse malware (see Table 1 for Filename/ MD5 hash).

On address 0040996A you see the trojan is checking the length of a string (in our case it turns out to be the command line arguments) and if this length is > 0 it executes the highlighted conditional jump at 00409980. We will come back to this jump later on.

Following the code we end up at the the address shown on Figure 2 and starting at 00890A09 we see a primitive string obfuscation used so the command line arguments could not be detected by the means of any kind of string extraction utility like the unix *strings*[7] and it's derivatives. If you look at the memory (near the bottom of the screenshot) you see the constructed string *-update*.

The same obfuscation is used several more times for more different arguments, but it is clear that without the proper command line argument, the trojan will not inflict it's damages. Coming back to the jump in the beginning, following the other

WHAT YOU WILL LEARN

How recent malware samples try to avoid analysis systems.

WHAT YOU SHOULD KNOW

Some basic assembly, windows basics, debugger basics.

execution branch (no/wrong command line argument) results in a jump at 00409A6D jumping near the end of the function and exiting the trojan (This part is not shown for the sake of length and because it's pretty much standard code).

Earlier versions of Zibku and similar malware employed a similar way of silently exiting the trojan if decryption keys were not present in the registry, although this method was rather easy to spot by looking at the keys checked in those automated systems.

While the first sample used a generic approach, the second sample shows detection and evasion tactics specific for the tested and very commonly used analysis systems.

Starting with the code shown on Figure 3, we see a call to `GetUserNameA` and a loop checking for various usernames used in analysis systems. This seems to be a somewhat generic evasion tactic though.

Specific detections start at code 004052FE as seen on Figure 4. This detection checks if the `dll SbieDll.dll` from Sandboxie was loaded into the address space of the running process. If this dll is found `eax` is set 1 and the code returns to the calling function, a classic. The same method is used in detecting `dbghelp.dll`, the Microsoft debugging dll.

The next code snippet as shown on Figure 5 tries to generically detect all Analysis systems based on VMWare using the widely known VMWare-specific parameters followed by an `in` opcode. Additional information on this detection is available on [CodeProject\[8\]](#).

The last detection implemented into this malware is directed against Anubis Malware Analysis system by checking if the filename that is currently executed is called `c:\sample.exe` as every `.exe` that is sent to Anubis is copied there and renamed to `sample.exe`. This is easily determined by the trojan and also spotted on Figure 6.

Specific Details

CWSandbox

The first sample did not produce any report that might indicate some

malicious activities, the second sample did not implement specific detections against CWSandBox, so you can see the results on Figure 9. CWSandbox is also

detecting the creation of copies of the virus into the directories of commonly used file-sharing applications that should trick people downloading them.



Figure 1. Beginning of malware and command line arg check

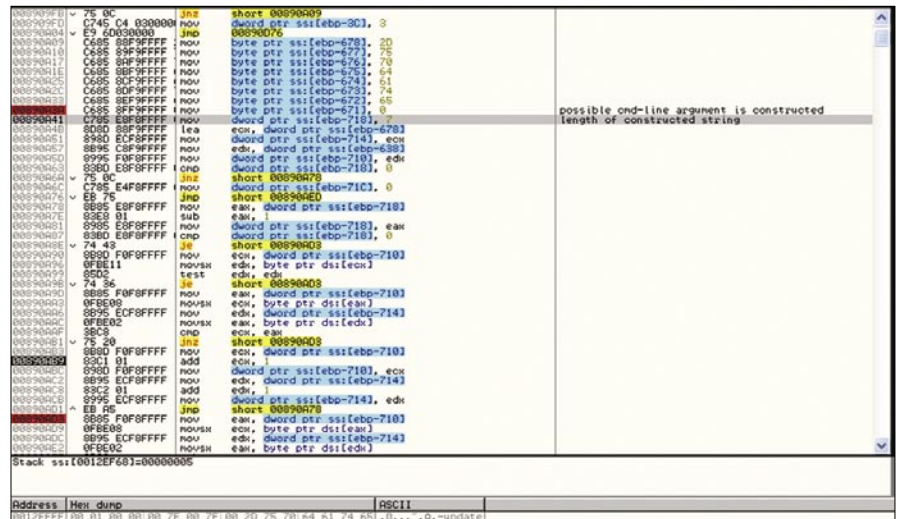


Figure 2. Command line argument string obfuscation



Figure 3. Different usernames of analysis systems are checked



Figure 4. Sandboxie detection

ATTACK

Excerpt:

C:\Programme\kazaa\my shared folder\
Limewire PRO Final Edition.exe
C:\Programme\kazaa\my shared folder\
Steam Crack.exe
C:\Programme\kazaa\my shared folder\
Counter Strike Source Crack.exe
C:\Programme\kazaa\my shared folder\
Windows XP Validator Crack.exe
C:\Programme\kazaa\my shared folder\
Spore Full Patcher.exe

C:\Programme\kazaa\my shared folder\
Spore Crack.exe
C:\Programme\kazaa\my shared folder\
Hotmail Hacker.exe
C:\Programme\kazaa\my shared folder\
Hotmail Cracker.exe
C:\Programme\kazaa\my shared folder\
Norton AntiVirus ALL VERSIONS
Crack.exe

This system produces generally not much noise.



Figure 5. Vmware detection



Figure 6. Anubis Detection



Figure 7. Anubis result, notepad.exe false positive

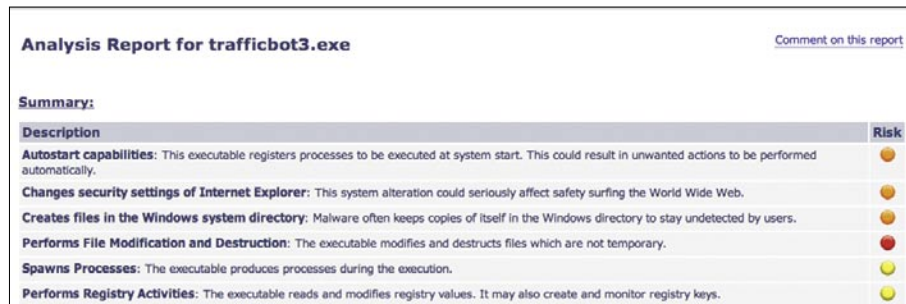


Figure 8. Anubis sample2 result

JoeBox

The first sample did not produce any results showing something malicious, the second sample did show some suspicious behavior, including the copied malware as mentioned with CWSandbox.

If network traffic (PCAP data) is requested, some noise that is generated by the analysis system is downloaded as well. The downloadable network traffic for the second sample did not reflect what was shown on the page.

Anubis

The first sample did not produce any result as Anubis is avoided generically. The second sample interestingly did provide useful information as with the latest version of Anubis the detection of the malware obviously was circumvented. Anubis missed the created copies of the malware as mentioned with CWSandbox. See Figure 8 for the Anubis results.

Anubis has a relatively high noise ratio, classifying even an original notepad.exe as risky. See a comparison on Figure 7, comparing the first sample and notepad.exe

VMWARE

As the first samples avoids analysis systems at all and the second one detects VMWare, the underlying system of the analysis systems itself, it will most likely be detected. There are numerous ways of detecting VMWare. VMWare itself is not an analysis system but one of the most widely used underlying fundamentals of analysis systems.

Sandboxie

As with version 3.42 both samples don't produce any suspicious results as Sandboxie is detected. Although SandBoxie is not directly an Analysis System, it's most widely used among end-users and might therefore offer a false sense of security for users if this is specifically detected and avoided and users trust the results and run the file outside the Sandbox.

Conclusion and Suggestions

In general, due to the amount of malware released every day and the public nature

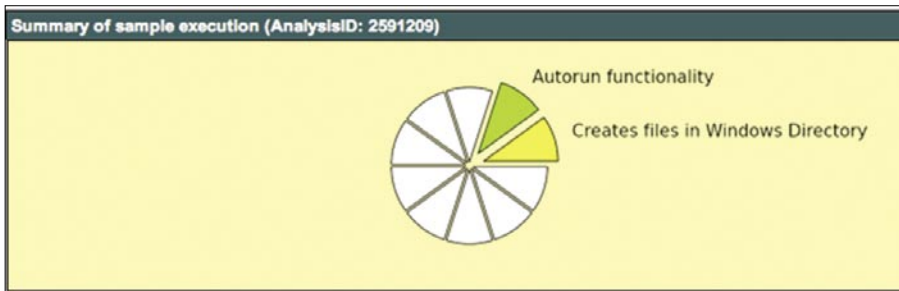


Figure 9. CWSandbox sample2 result

First Sample

File: A0016078.exe

Size: 224768

MD5: 9651C3834CBB0E34DB705CBF5FCA9B87

Second Sample:

File: trafficbot3.exe

Size: 53248

MD5: 74AB05D1EBDBA509FD68711B360C1235

On the 'Net

- David Marcus, McAfee Labs – <http://www.avertlabs.com/research/blog/index.php/2009/07/22/malware-is-their-businessand-business-is-good/> [1]
- Ronen Tzur, Sandboxie – <http://www.sandboxie.com> [2]
- University of Mannheim – CWSandbox – <http://www.cwsandbox.org> [3]
- Stefan Buehlmann, Christopher Liebchen, JoeBox – <http://www.joebox.org> [4]
- Ulrich Bayer, Anubis: Analyzing Unknown Binaries – <http://anubis.iseclab.org/> [5]
- VMware – <http://www.vmware.com> [6]
- strings – <http://linux.die.net/man/3/string> [7]
- <http://www.codeproject.com/KB/system/VmDetect.aspx> [8]

of those tested systems, it is very unlikely they go undetected for long, even after they get updated to thwart detection efforts. As most of the malware still is not using full evasion tactics and those systems get updated to prevent detection, they normally provide helpful information in a quickly and structured way a human analyst could not achieve. End-user might want to combined these results though with results from e.g. www.virustotal.com for plausibility checking.

All the tested systems did no provide any means of specifying command line parameters for example to get the first sample to behave in the way it was found on the infected machine prepared by the malware installer. A human malware researcher or analyst is necessary to figure out what command line parameters are needed to be provided to get useful results on those systems.

No means are provided in specifying function names in case the malware is

available only as a .dll file and is originally called by *rundll32.exe*.

When it comes to semi/professional malware analysis, it might be better to away from commercial virtual environments as VMWare or Virtual PC as there are too many detection methods available and widely implemented unless you just have to manually few samples and avoid those detections during debugging/analysis. In-house developed emulators and analysis systems should be used especially if mass-analysis needs to be undertaken and greater control is needed. This is standard for most Anti-Virus companies by now. Noise, in this case system activity reported but not maliciously related to the sample, should be minimized as not to make the users unsecure.

Florian Eichelberger

fio@dynamix.at



[GEEKED AT BIRTH]



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

[IT'S IN YOUR DNA]

LEARN:

Advancing Computer Science
Artificial Life Programming
Digital Media
Digital Video
Enterprise Software Development
Game Art And Animation
Game Design
Game Programming

Network Engineering
Network Security
Open Source Technologies
Robotics And Embedded Systems
Serious Game And Simulation
Technology Forensics
Virtual Modeling And Design
Web And Social Media Technologies

www.uat.edu > 877.UAT.GEEK



JEREMIAH BROTT

Analyzing Malware & Malicious Content

Difficulty



One of the biggest challenges storage managers face is to try and calculate just how much storage is needed by that ultra important, mission critical application in the datacenter.

Malware, short for malicious software, is a piece of software that's sole purpose and design is to infiltrate or cause damage to a computer system without the owner's well informed consent. In the information security world we hear this term or expression all the time used by professionals to describe a variety of hostile, intrusive or other wise annoying code running on a system.

It is not uncommon to hear people still using the term *computer virus* as a catch-all sort of phrase to include all types of malware, without neglecting a real computer virus that would fall within. Having said that, Malware can be used to classify computer viruses, worms, Trojan horses, majority of root kits, spyware, suspicious adware, along with any other un-wanted software

Just because a piece of software may be buggy or defective will not be enough to be classified as malware. [1] Some findings by Symantec published in 2008 suggest that *the release rate of malicious code and other unwanted programs may be exceeding that of*



legitimate software applications. If you think this is something to worry about, according to F-Secure, [2] *As much malware [was] produced in 2007 alone, than has been created in the previous 20 years all together.*

It is a safe bet to say, if you have ever browsed the internet, sent & received some emails or browsed to your favorite website. You have ultimately encountered a piece of malware once or twice.

WHAT YOU WILL LEARN

How to analyze and organize Malware from the ground up.

Tools, tips and techniques used for analyzing and organizing your malware collection.

WHAT YOU SHOULD KNOW

Should have a basic understand of malware infections, ASM knowledge would help with the reverse engineering process. But bare bones just an understanding of malware and high level of how it works.

Summary

Analyzing malware & malicious content aim's to be a paper covering the topics of analyzing malicious content and malware infested executables. This paper will take the reader through the very beginning steps of safely acquiring some malware specimen, and analyzing & disassembling the malware in a safe, controlled environment. Some basic knowledge of ASM, C will be assumed, as well as familiarity with tools such as OllyDBG, IDA Pro, Wire Shark and so on...

Why Analyze Malware anyways....?

In short, there is a multitude of different reasons we would analyze malware, see below to name just a few:

- Discovering signs of an intrusion or attempt there of.
- Assess the damages after an intrusion has been detected.
- Identifying vulnerabilities or ways to detect and identify new malware for developing techniques at preventing future infections.
- Answering questions...?

Creating Safe Malware Analysis Environments

If you can recall in the story *The Three Little Pigs*, they all had their own ideas of building a safe & secure shelter. When the first two failed, luckily the 3rd one had built an armored fortress house made of brick. When building our malware analysis environment, we want to make sure their will be no mistakes and we have a proper armored fortress the first time.

At the very least when building a malware analysis environment, we want to be 100% for sure that these system(s) do not have any access to any live production systems or the Internet. It is a good idea to always start out with a clean slate using a fresh install of the OS of your choice for analysis. I prefer to keep around several pristine installs of various operating systems and versions for analysis on a live system. When you start from

a pristine state it becomes much easier to track and monitor changes done to the system after running a piece of malware for analyzing.

Physical VS Virtual Environments

You will have several options when creating a malware analysis environment. If you have the hardware lying around you can always build your lab using bare metal machines. I prefer to use virtualized Operating systems for several reasons that will be touched on below:

- Faster Restore times to a pristine state using non-persistent images
Non-persistent images support: VMware, Xen, Parallels
- No need for extra hardware lying around
- Switching between Operating Systems much faster
- The list could go on... which is going out of scope for this paper ;)

With anything good, there is always some down falls or faults that lie hidden within the details. I have always said if you wish to see the most creativity in computer security/programming, check and see what the malware authors are up too. At the end of the day their creations sole purpose is to run undetected and do its dirty deeds in the background. These guy's are savvy to debugging and disassembling as well, I have seen more than one piece of malware completely disable it's self

when it is detected to be running within a virtual environment. This leads too a manual intervention using our good friends OllyDBG, IDA Pro along with others we will dive into shortly.

Building the Environment

If you are using real machines or virtual machines, the setup logically is pretty much the same. We will need at least one machine to serve the purpose of the victim. On this machine we will run the operating system of choice on which we wish to infect and perform a live analysis on a piece of malware.

The second machine will be the monitoring machine. Here we will only ever store and hash malware specimen. Be sure to NEVER run, load and or debug any malware on this machine. Everything must be READ-ONLY!

The only thing to really change is in the networking if you are using real machines or virtual machines. If you are using real machines and not familiar with sniffing on a switched network OR the setting up of SPAN ports on a switch. I suggest you use a simple hub, as all packets will be replicated to each port allowing the easy capture of traffic from the victim machine. Be sure to make sure your monitoring machine is fully patched, some malware is weaponized.

Meaning they have the ability to scan for and infect other host on the network. I have seen malware in the wild that would run their own DHCP server handing out valid addresses but a poisoned DNS server under an attacker's control.

Table 1. Analyzing Malware on Physical Machines

Pros	Cons
Essentially every piece of malware/virus will operate* Tighter control to ensure the malware does not escape the environment into the host.	Can be costly on hardware. Requires more physical space for hardware. Takes longer to restore pristine image

*always add some extra character on the end of malware extension to prevent accidental execution. Yes it does happen... Yes, I have done it before... =>

Table 2. Analyzing Malware on Virtual Machines

Pros	Cons
Faster setup times Non-persistent image support	Virtualization detection from malware

Listing 1. Example Directory structure for organizing malware

```
Naming Scheme:
<malware_type>.<platform>.<family_name>/<malware_type>.<platform>.<family_
name>.<variant>.<file_type>
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/notes.txt
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/file_exe
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/unpacked/file_exe
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/unpacked/file_exe.md5
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/unpacked/file_exe.shal
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/unpacked/file_exe.strings
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/file_exe.md5
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/file_exe.shal
Backdoor.Win32.Kbot/Backdoor.Win32.Kbot.al/file_exe.strings
```

There is a catch 22 involved here... It is a good idea to allow the malware to interact with its server out in the wild & scary internet for complete inspection, though doing so could very well could put you in the middle of a WW3 battle with the malware bot herders. Ever heard that saying about not meddling in the affairs of a system admin? Trying angering a bot herder....;D

Physical Machines

- Minimal 2 machines. (1 machine for running malware) (1 machine for running network analysis)
- Norton Ghost or some other imaging software
- Hub – Switch (networking) hub is probably the best choice, makes the sniffing of traffic much easier (Table 1)

Virtual Machines

- Minimal 1 Real system
- Virtualization software:
 - VMware [Workstation – Player, Server (free)]
 - XEN (free)
 - QEMU (free)
 - Microsoft Virtual PC (free) (Table 2)

Organizing the Malware Collection – Keeping things tidy

once your collection starts to grow, if it has not already... Things will tend to get pretty messy. It is a good idea to keep your collection organized from the beginning, as this will greatly pay off in the end.

I have found a scheme online for managing viruses and modified it of sorts for keeping my collections clean and organized. Reverse engineering of

malware can be a very time consuming process and require lots of notes and logging of details. Using the methods and naming convention below, this will allow for an easier to manage collection for management of the binaries as well as data and notes pertaining to each file in the collection.

I like to keep my collection organized by starting out with the family name of the malware. Under each family name I will then create sub directories for each variant that is classified to be within a specific family of malware. Example: see Listing 1.

Analyzing Malware – The first steps...

Now that we have a safe environment built for analyzing some malware, and assuming that you have some malware collected already.

Let's begin the steps of analyzing to see what we can learn about its mysterious ways. We first choose our initial method of attack at analyzing the malware.

Static VS Dynamic Analysis

When analyzing malware there is generally 2 approaches you can take. In most cases both may be required, where as in others one may be enough.

Static Analysis

Static analysis will always be your safest bet when analyzing malware. This is



mainly due to the reason that ALL of the analysis is done as read-only per se. No actual executing of the malware is ever performed in a live environment. Performing a static analysis you will not need to go to such great lengths as compared to a dynamic analysis of malware on a live system.

Dynamic Analysis

Dynamic analysis can get a bit more interesting for you as a researcher as well as exposing yourself or others to further risk of attack. If you must allow your victim machine access to the net during analysis for capturing traffic to the mother ship, I would suggest making use of the TOR network. The bot herder may notice you tampering around and turn the BOT net around on you. This never equals good times...

Fingerprinting Malware Binaries

The very first step before analyzing any piece of malware is the renaming & hashing of the original binary. The process of renaming the malware is a good habit to get into this will prevent the accidental execution of the malware. Assuming we have a file file.exe we are attempting to analyze.

Original: file.exe New: file_exe

For the purpose of hashing we will use tools like MD5, SHA1. Hashing a file using the utilities is fairly simple to do. Hashing of the file is pretty essential when you are analyzing Malware. This will ensure the malware file its self does not change during analysis.

To keep things organized, I store the file hash in a file with the following naming convention: *File_ext.hashType* Example: *file_exe.md5*

We first hash the file with the following process.

```
Magikh0e@ihtb.org:~$ md5sum file.exe >
file_exe.md5 d41d8cd98f00b204e980099
8ecf8427e file.exe Magikh0e@ihtb.org:
~$ sha1sum file.exe > file_exe.sha1
da39a3ee5e6b4b0d3255bfe95601890af
d80709 file.exe
```

Gathering Strings from binaries

After hashing the binary, I always tend to search for some low hanging fruit.

Using the strings command is a perfect way to grab all of the human readable/printable characters from the file and store them away in a separate file for later analysis.

Gathering strings from a binary can be performed in several different ways. Linux systems with GNU tools come with a utility built in which can be accessed using the strings command.

Note: In some cases no strings will be returned, this usually means the file has been packed.

```
Magikh0e@ihtb.org:~$ strings file_exe
> file_exe.strings /lib/ld-linux.so.2
__gmon_start__
libc.so.6
_IO_stdin_used....
```

Automating the process

I have hacked up a quick Perl script to auto-mated the process of collecting

the MD5 and SHA1 hash of the binary, then gather printable characters found within the binary and log them to a file. The purpose of the script is to give you an idea at creating some of your own automated way's of managing/analyzing your collection. If you come up with anything you think would be useful, I would love to check it out... (see Listing 2).

Identifying Known Malware

I've always hated to re-invent the wheel or see someone else suffer trying. Having said that, now we have renamed and hashed our malware specimen. It is time to pull out your favorite anti-virus scanner.

Start scanning the binary you are analyzing using different AV scanners. We do this for several purposes. One the file may have already been identified and documented, voiding the need for manual analysis.

There is a fair amount of good online resources for this purpose and help considerably when speeding things up.

Listing 2. A perl script to make gathering some information from malware files a bit easier/automated. It will collect md5, sha1 sums and printable strings from the binary given

```
#!/usr/bin/perl # Script for collecting hashes and strings from a malware binary.
use strict; use warnings;
my $MD5 = '/usr/bin/md5sum'; my $SHA1 = '/usr/bin/sha1sum'; my $STRINGS = '/usr/bin/
strings'; my $mw_file; $mw_file = $ARGV[0];
if (!$mw_file) { print "Malware File: "; chomp($mw_file = <STDIN>); }
if (-e $mw_file) { system("$MD5 $mw_file > $mw_file.md5"); system("$SHA1 $mw_file >
$mw_file.sha1"); system("$STRINGS $mw_file > $mw_file.strings");
} else { die "File: $mw_file - does not exist: $0"; }
```

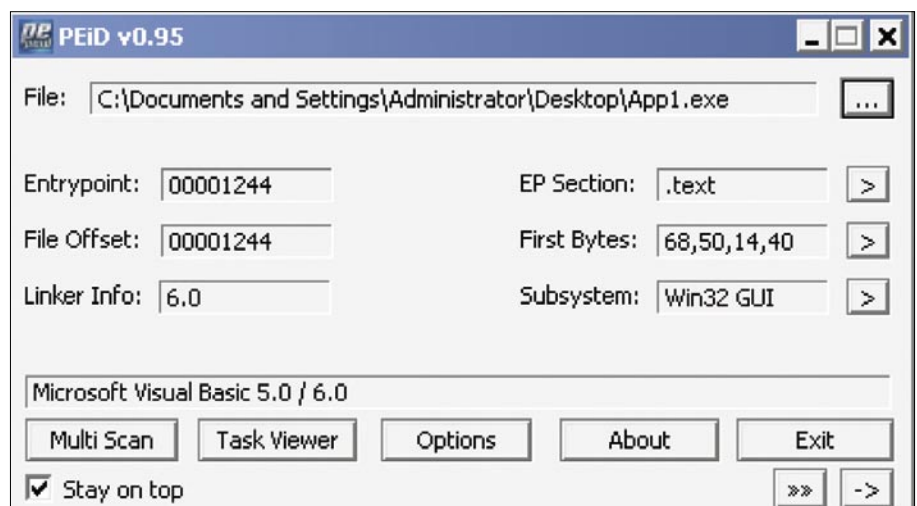


Figure 1. Example of using PEiD to detect the packer used on a binary file

One of the best sites I know of for this purpose is <http://virusscan.jotti.org/>. Using this site you will have the ability to submit a file and then have it scanned using 21 of the leading anti virus vendors. I tend to live by the motto of security being a process, not a product. Make sure to use more than one scanner.

Online Virus scanning:

- <http://virusscan.jotti.org>
- <http://www.virustotal.com>

Known Malware

If you are lucky, the AV will be able to detect the malware you are analyzing. Upon detection the chances are very high of someone on the internet documenting every detail about the malware will already be posted. Google is your friend.

Unknown Malware

Now you must have a binary that you have suspicions of, but none of the current anti-virus scanner definitions are picking up any malware infection within the binary it's self. Now is where things will get tricky, interesting, frustrating and mentally anguishing all at the same time. Have no fear; there is a vast variety of tools to help you accomplish the task of debugging, disassembling and analyzing binaries of unknown origins and or purposes.

Unknown Malware, Packers and Cryptors... oh my!

The first steps towards identifying an unknown piece of malware is too first analyze the binary with a fine tooth comb. Chances are the binary has been packed with some sort of packer. If luck

is on your side the packer being used will already have been documented with details on the unpacking of the executable.

Now you are most likely wondering just how to identify what packer is being used. For this purpose we will be using a utility called PEiD by: Jibz, Qwerton, snaker, xineohP. At the time of this writing the latest version available is 0.95.

Packed Malware

Malware authors often use packer software in order to evade detection and to make sure the malware has a smaller footprint so it can squeeze in places and remain hidden for as long as possible. Packers can be quite tricky in identifying and more importantly, un-packing. A tool as the one below is used in order to attempt identifying which packer a binary is currently using. Once identified hopefully finding the packer or resources for manual unpacking will be easier to find.

PEiD

PEiD detects most common packers, cryptors and compilers for PE files. It can currently detect more than 600 different signatures in PE files (see Figure 1).

If you are lucky your job will be made tremendously easier and there will not be a packer being used on the executable; however that would be in the perfect world. Most malware authors do not like people poking around in and with their code. So they will go to great lengths with packers, cryptors and other obfuscation techniques they manage to conjure up in order to keep the prying eyes away from the prize.

So, you ran into a packed file huh? Once you have identified that you are dealing with a packed piece of malware, hopefully PEiD will be able to identify the packer currently being used. If so, this will tremendously speed up the efforts at reverse engineering the piece of malware successfully. Once you have determined the name of the packer, we rush away to several places on the web with a vast resource of information on specific packers and how to unpack them. In some cases you will be unable to successfully

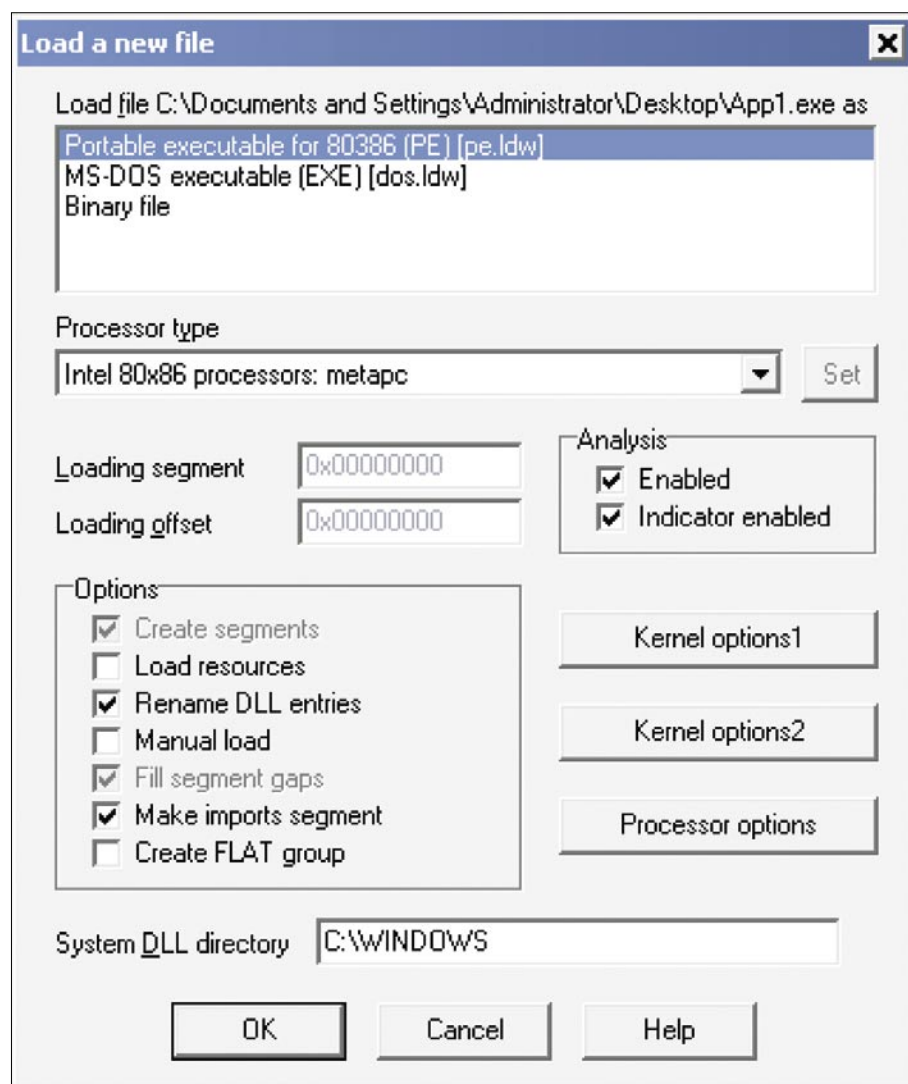


Figure 2. Loading a new file into IDA Pro

ANALYZING MALWARE & MALICIOUS CONTENT

unpack a binary. In these cases your only option is to perform dynamic analysis while the malware is running. More on this later...

Manual Un-Packing Malware

When attempting to unpack binaries by hand manually we will be on the hunt for the OEP entry. Sounds easy enough, yeah? ;) Olly and IDA both have some plugins to help in the finding of the OEP, other times this is a pain staking manual process. When attempting to locate the OEP, keep an eye out for the following:

- Jumps or CALLs to EAX. This may point you to the OEP, this JMP or CALL could possibly be preceded by POPA or POPAD.
- Tricky jumps via SEH, CALL, RET etc...
- If the packer you are dealing with uses SEH, You can anticipate OEP by tracking stack areas used to store the packers' handlers.
- Breaking on the un-packers calls to LoadLibraryA or GetProcAddress may help in getting closer to OEP.
- When unpacking in OllyDbg, try SFX (bitwise) and OllyDump's Find OEP by Section Hop.

Unpacking a binary quick and dirty

Infect the victim system and dump the memory using LordPE or OllyDump.

Unpacking a binary butcher style

Find the OEP *original entry point* after the un-packer has been executed.

Packer Analysis Database

- http://www.openrce.org/reference_library/packer_database
- MW-Blog: <http://www.teamfurry.com/wordpress/>

The above sites are some pretty good resources that I have found on the net with information related to packers.

The MW-Blog seems to have not been updated for awhile, at the time of

this writing the last update was on June 15th 2009. None the less still contains some good information on un-packing packers. Most of the blog postings seem to use IDA Pro for the actual unpacking process which leads us into the next section...

Disassembling – (static analysis)

The art of disassembly is another paper on it's own to cover in detail and give it due justice. However that is a bit out of scope for this paper, so we will briefly cover it and some tools you can use along with tips to help you out along the way.

Almost all of the software you see available today is written using a high

level language like C, C++, and Delphi etc... Same goes for the malware...

These languages generally speaking, are what are known as a compiled language that will turn the high-level code into low-level code that a computer can understand.

This is what we know as Assembly. Not to be confused with de-compilers, Disassemblers will take a binary file that could have been written in any available compiled language, and disassemble the binary into assembly source code.

While a de-compiler takes a binary file generated by a specific compiler and attempts to reverse the binary back into the high-level code it was created from originally.

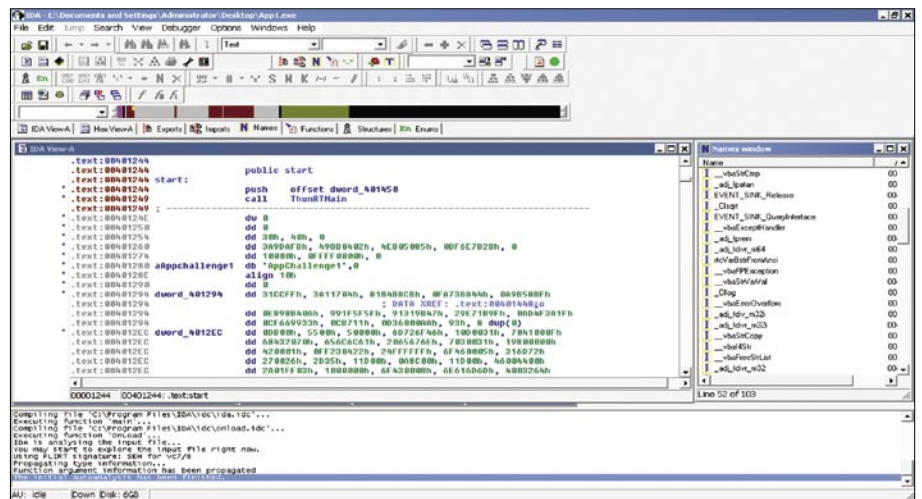


Figure 3. The main window of IDA Pro

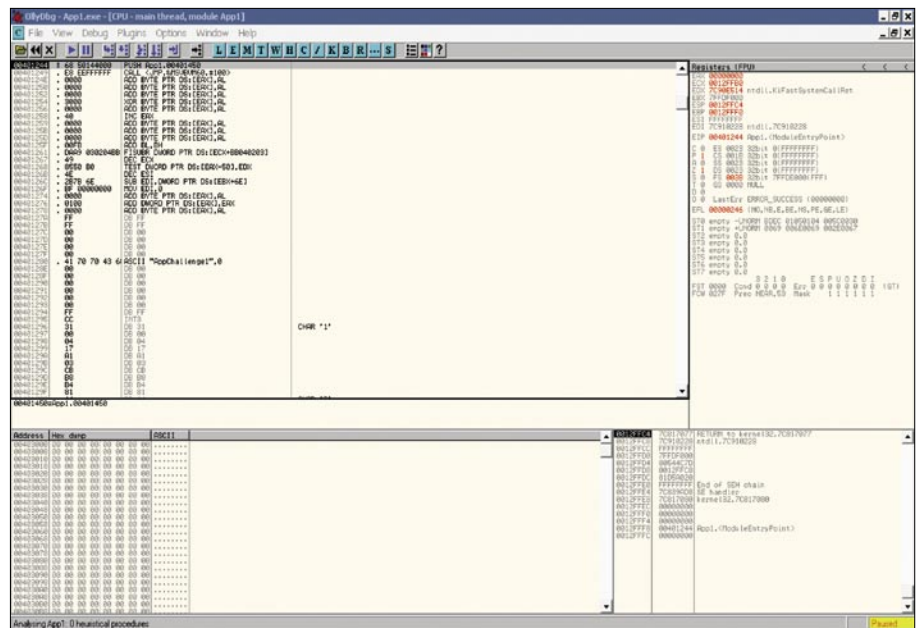


Figure 4. The main window of OllyDBG

Common x86 Registers and Uses

- EAX – Addition, multiplication, function results
- ECX – Counter
- EBP – Base for referencing function arguments (EBP+value) and local variables (EBP-value)
- ESP – Points to the current top of the stack; changes via PUSH, POP, and others
- EIP – Points to the next instruction
- EFLAGS – Contains flags that store outcomes of computations (e.g., Zero and Carry flags)

Picking a good disassembler

A good disassembler needs to have the ability to accurately distinguish between data and code of a binary. While a good

de-compiler will need to do this and have the ability to understand what code construct in the original high-level language generated the code to begin with.

IDA Pro

For static analysis my personal choice in preference is IDA Pro, why IDA Pro you may ask?

Well IDA Pro is not a de-compiler or a disassembler... Think of IDA as the love child if W32dasm and OllyDBG had a baby... IDA stands for the interactive disassemble, and it can withstand living up to its name. IDA also has loads of features that you could classify it as a de-compiler due to a feature known as FLIRT.

If you have ever dabbled in C you will know that every program ends up using

some functions that are supplied by the compiler itself, or as a part of the Win32 API as an example.

Take a look at a `printf()` statement, any C program that makes a call to `printf` will have the same piece of code inside them.

During compile time the process of the compiler will in-turn link the code for the function from its included library. If it is not so obvious yet, this will allow for the disassemble to recognize the code patterns of specific functions then add the ability of mapping a name to it that is somewhat meaningful.

If you ever dabbled in reverse engineering you can see how this will become useful, ever spent an hour tracing through a function only to find out later that was only that specific compilers variant of `fseek()`. To me this is what sets IDA aside from others with the FLIRT abilities. The FLIRT libraries come with a huge set of signatures for specific functions from the various available compilers on the internet. Best of all they do not stop at C alone, there is also signatures for compilers like Pascal, Delphi, VB and more... To learn more about FLIRT, check out the IDA page in resources. FLIRT is just one of the many great things about IDA, there is a vast resource of plugins you can find online or learn how to create your own. Loading of a file in IDA Pro... (see Figure 2).

IDA Pro main window after a binary has been loaded, the green circle also indicates that the IDA analysis has been completed... (see Figure 3).

IDA Pro Interface

- IDA View A - Action name:
WindowOpen - Disassembly window
- Exports - Action name:
OpenExports - Exports Window
- Imports - Action name:
OpenImports - Imports Window
- Names - Action name: OpenNames
- Names Window

The GUI version displays a small icon for each name:

- L (dark blue) – library function
- F (dark blue) – regular function
- C (light blue) – instruction

Listing 3. Detecting VMware

```
procedure TForm1.btnJerryClick(Sender: TObject);
var a, b:cardinal;
begin
a:=0;
try
asm
push eax
push ebx
push ecx
push edx
mov eax, 'VMXh'
mov ecx, 0Ah
mov dx, 'VX'
in eax, dx
mov a, ebx
mov b, ecx
pop edx
pop ecx
pop ebx
pop eax
end;
except on E:Exception do ShowMessage(E.Message);
end;
if a=$564D5868 then
begin
ShowMessage('In VMware');
case b of
1 : ShowMessage('Express');
2 : ShowMessage('ESX');
3 : ShowMessage('GSX');
4 : ShowMessage('Workstation');
else ShowMessage('Unknown version')
end;
end
else
ShowMessage('Native system');
end;
```


- A (dark green) – ascii string
- D (light green) – data
- I (purple) – imported name

Functions – Action name:

OpenFunctions – Functions Window

Listed for each function are:

- function name
- segment that contains the function
- offset of the function within the segment
- function length in bytes

The last column of this window has the following format:

- R – function returns to the caller
- F – far function
- L – library function
- S – static function
- B – BP based frame. IDA will automatically convert all frame pointer [BP+xxx]operands to stack variables.
- T – function has type information
- = – Frame pointer is equal to the initial stack pointer. In this case the frame pointer points to the bottom of the frame
- M – reserved
- S – reserved
- I – reserved
- C – reserved
- D – reserved
- V – reserved
- Structures – Action name:
- OpenStructures – Structures Window

You can modify structure definitions here: add/rename/delete structures, add/delete/define structure members.

Enums – Action name: OpenEnums
Enums Window

You can modify enum definitions here: add/edit/delete enums, add/edit/delete enum members (i.e. user-defined symbolic constants)

IDA Pro Tutorials

- <http://www.hex-rays.com/idapro/idasupport.htm>

- <http://ebook-net.blogspot.com/2008/05/reverse-engineering-with-ida-pro.html>
- <http://www.woodmann.com/crackz/Tutorials/IdaTut.zip>
- http://www.hex-rays.com/idapro/debugger/gdb_qemu.pdf
- http://www.hex-rays.com/idapro/debugger/gdb_vmware_linux.pdf
- http://www.hex-rays.com/idapro/debugger/gdb_vmware_winkernel.pdf

IDA Pro Shortcuts

Here we will list some handy shortcuts...

- Text search [Alt+T]
- Show strings window [Shift+F12]
- Show operand as hex value [O]

Insert comment:

- Follow jump or call in view [Enter]
- Return to previous view [Esc]
- Go to next view [Ctrl+Enter]
- Show names window [Shift+F4]
- Display function's flow chart [F12]
- Display graph of function calls [Ctrl+F12]
- Go to program's entry point [Ctrl+E]
- Go to specific address [G]
- Rename a variable or function [N]
- Show listing of names [Ctrl+L]
- Display listing of segments [Ctrl+S]
- Show cross-references to selected function Select function name>[Ctrl+X]
- Show stack of current function [Ctrl+K]

Listing 4. Detecting Wine

```
function TForm1.IsRunningWine: boolean;
var hnd:THandle;
    wine_get_version: function : pchar; {$IFDEF Win32} stdcall; {$ENDIF}
    wine_unix2fn: procedure (p1:pointer; p2:pointer); {$IFDEF Win32}
stdcall; {$ENDIF}
begin
result:=false;
hnd:=LoadLibrary('ntdll.dll');
if hnd>32 then
begin
wine_get_version:= GetProcAddress(hnd, 'wine_get_version');
wine_unix2fn:= GetProcAddress(hnd, 'wine_nt_to_unix_file_name');
if assigned(wine_get_version) or assigned(wine_unix2fn) then result:=true;
FreeLibrary(hnd);
end;
end;
```

Listing 5. Detecting VirtualBox

This was written by the VirtualBox author, just detects the existence of VBoxService.exe.

```
function InVirtualBox:boolean;
var handle:THandle;
procinfo:ProcessEntry32;
begin
result:=false;
handle := CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);
procinfo := sizeof(PROCESSENTRY32);

while(Process32Next(handle, procinfo)) do
begin
if POS("VBoxService.exe",procinfo.szExeFile)>0 then
begin
CloseHandle(handle);
result:=true;
exit;
end;
end;
CloseHandle(handle);
end;
```

Debugging and Debuggers – (dynamic analysis)

Debugging and debuggers themselves play another important part in analyzing malware. These are typically used once you get to the dynamic analysis portion when analyzing a new piece of malware in your collection.

When it comes to debuggers I am torn between two like Michael Jordan and baseball?? Good you are still paying attention.

Anyways, the two debuggers of my preference for analyzing malware is Immunity DBG and Olly DBG. They both are very good and what they do with some advantages and disadvantages over one another. Which we will not go

into here... For sole purpose of simplicity and more information online, we will mainly be focusing on Olly DBG for this paper.

OllyDBG

OllyDBG main window... (see Figure 4).

Note: I did not forget about gdb, we will be using some linux tools soon for performing some analysis.

OllyDBG

Shortcuts Here we will list some handy shortcuts...

- Step into instruction [F7]
- Step over instruction [F8]
- Execute till next breakpoint [F9]

- Execute till next return [Ctrl+F9]
- Show previous executed instruction [-]
- Show next executed instruction [+]
- Return to previous view [*]
- Show memory map [Alt+M]
- Follow expression in view [Ctrl+G]
- Insert comment [;]
- Follow jump or call in view [Enter]
- Show listing of names [Ctrl+N]
- New binary search [Ctrl+B]
- Next binary search result [Ctrl+L]
- Show listing of software breakpoints [Alt+B]
- Assemble instruction in place of selected one *Select instruction*>Spacebar
- Edit data in memory or instruction opcode *Select data or instruction*>[Ctrl+E]
- Show SEH chain *View*>*SEH chain*
- Show patches [Ctrl+P]

Listing 6. Detecting Virtual PC

```
function TForm1.IsRunningVirtualPC: boolean;
asm
    push ebp;
    mov ebp, esp;

    mov ecx, offset @exception_handler;

    push ebx;
    push ecx;

    push dword ptr fs:[0];
    mov dword ptr fs:[0], esp;

    mov ebx, 0; // Flag
    mov eax, 1; // VPC function number

    // call VPC
    db $0F, $3F, $07, $0B

    mov eax, dword ptr ss:[esp];
    mov dword ptr fs:[0], eax;

    add esp, 8;

    test ebx, ebx;

    setz al;

    lea esp, dword ptr ss:[ebp-4];
    mov ebx, dword ptr ss:[esp];
    mov ebp, dword ptr ss:[esp+4];

    add esp, 8;

    jmp @ret1;

@exception_handler:
    mov ecx, [esp+0Ch];
    mov dword ptr [ecx+0A4h], -1; // EBX = -1 ->; not running, ebx = 0 -> running
    add dword ptr [ecx+0B8h], 4; // ->; skip past the call to VPC xor eax, eax; //
    exception is handled @ret1: end;
```

Defeating Malware Defenses

Most malware nowadays implement some sort of defense aimed at defeating analysis of the binary or any sort of reverse engineering in order to evade detection. Along with the evasion of detection, packing techniques are also used to make the malware smaller and easier to cram in places. As we have mentioned earlier on that some types of malware have the ability to detect if they are running within an emulated environment or not. Malware gets trickier and more obfuscated by the day; the creators are even more insane with new creations...

They also have the ability to detect well known debuggers running on the host system and/or implement obfuscation techniques within the code its self. I will include a tutorial in the references for a great read on anti-debugging tips, tricks and other techniques used.

OllyDBG undercover

Conceal OllyDbg using HideOD and OllyAdvanced.

- http://www.openrce.org/downloads/details/238/Hide_Debugger
- http://www.openrce.org/downloads/details/241/Olly_Advanced

IDA Stealth

IDA Stealth is a plugin which aims to hide the IDA debugger from most common anti-debugging techniques. The plugin is composed of two files, the plugin itself and a dll which is injected into the debuggee as soon as the debugger attaches to the process. The injected dll actually implements most of the stealth techniques either by hooking system calls or by patching some flags in the remote process. <http://newgre.net/idastealth>.

Detecting Virtualized Environments

As we have discussed earlier on, a lot of malware that you see nowadays have the ability to detect that they are currently running within a virtual environment. Below we will look into the way's a virtual environment could be detected.

VMWare

VMWare is a bit easier to detect depending on your configuration. For the main part vmware-tools is usually always installed on the OS anyways. There is also a small C routine by the name of Jerry.c that works well for the most part, though does have some quirks. Jerry.c uses a simple communication channel that VMware has left open. As mentioned depending on the configuration, the administrator can change this channel. Chances are high of the OS being designed to act as a honey pot if this channel has been changed anyways...

You can think of Jerry.c is a first generation of it's kind. As with anything good, usually better improvements follow. This is where Scoopy comes into play, written by the same author as Jerry.c.

Scoopy is far more complex in the way it works as compared to Jerry.c. Scoopy inspects how the system is actually virtualized using the SIDT CPU instruction instead of a communication channel, as Jerry.c does. Another interesting write up on detecting virtualized operating systems if the red pill paper.

- Jerry.c – <http://www.trapkit.de/tools/index.html> – Outdated now, see ScoopyNG.
- Scoopy – <http://www.trapkit.de/tools/index.html>
- Red Pill Paper: <http://invisiblethings.org/papers/redpill.html>

VMWare can also be detected via the MAC address being used. VMware by default will use a MAC address starting with 00-05-69. Though keep in mind, a MAC address can be changed fairly easy.

Wine

For wine we also have several options in order to detect being run within the environment. When you are using wine they have created a special dll entry to `NTDLL.DLL` called `wine_nt_to_unix_file_name`.

When you are using LoadLibrary and GetProcAddress, you can determine if you

are under a wine environment or not by checking for the presence of the new entry into `NTDLL.DLL`. Jerry.c also works under wine, just will state that vmware has been detected.

Registry Keys

Wine can also be detected by opening the following registry keys: `HKLM\Software\Wine` or `HKCU\Software\Wine`.

Last but not least. Open up any critical Windows file and locate the OEP, when running under a wine environment the function will disassemble to the following instructions:

- `.text:10001000 public start`
- `.text:10001000 start proc near`
- `.text:10001000 mov eax, 1`
- `.text:10001005 retn 4`
- `.text:10001005 start endp`

For the quick and dirty, Just search for the following binary string `B8 01 00 00 00 C2 04 00` at `.text:10001000 ;D`

XEN

Xen can be detected fairly reliable using the WMI interface and querying the BIOS manufacturer information. Xen can also be detected via the MAC address being used. Xen host will use a MAC address starting with 00-16-3E

VirtualBox

Virtual Box MACs all start with 08:00:27

Code Examples

The following is some code examples that can be used in detection of each specified virtual environment.

Special Thanks too everyone I bugged in the process of writing this! Gr@ve_r0se, Neize aka Oryl, Skiffy, Mannibal, Redsand... darpa hax0rs

Jeremiah Brott

Jeremiah currently holds a lead role with Access2Networks Toronto as an Information Security Consultant. In addition to holding numerous certifications, Jeremiah is also the professor for Malicious Code – Design & Defense along with Ethical Hacking at Sheridan Institute for the Applied Information Sciences System Security degree program.

Hacker's do it with all sorts of characters... www.IHackedThisBox.com

Listing 7. Detect Virtualized guest based on MAC address

```
#!/usr/bin/perl
# Detect virtualized guest based on MAC address.
# Do not expect this method to be very reliable at all. MAC addresses
# can easily be changed. ;D
my $ifconfig = '/sbin/ifconfig -a eth0 2>&1 | /bin/grep HWaddr';
my @info = split(/\s+/, $ifconfig); print $info[4] . "\n";
if ($info[4] =~ /08:00:27/) { print "\nVirtualBox MAC Detected\n"; }
if ($info[4] =~ /00:16:3E/) { print "\nXen MAC Detected\n"; }
if ($info[4] =~ /00:05:69/) { print "\nVmware MAC Detected\n"; }
if ($info[4] =~ /00:0C:29/) { print "\nVmware ESX MAC Detected\n"; }
if ($info[4] =~ /00:50:56/) { print "\nVmware VC ESX MAC Detected\n"; }
```

On the 'Net

- http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xiii_04-2008.en-us.pdf [1]
- http://www.f-secure.com/f-secure/pressroom/news/fs_news_20071204_1_eng.html [2]



MAYANK AGGARWAL,
SMOBIILE GLOBAL THREAT
CENTER RESEARCH
ENGINEER

Study of MITM Attacks Against Smartphone Devices

Difficulty



Let us consider a scenario, where a smartphone user connects to the Internet to determine if the bank transaction is successful or not.

The user is having a cup of coffee in a cafe, which happens to provide a free Wi-Fi hotspot. The user decides to use the free Wi-Fi hotspot to connect to online bank website and enters account credentials to log in to the site, directly from the browser on his smartphone handset. The user successfully enters the online bank account and verifies the transaction history. After finishing the work, the user log out and closes the web page as requested by the online bank website. At this point, the user has no reason to suspect that anything malicious occurred. The user finishes the coffee and returns to the office.

As is the current state of the technology surrounding smartphone devices, there are very limited amounts of reputable applications that even address traditional information security concerns. The reasons why the technology is lacking are wide ranging, but lead to one simple conclusion...basic information security malware and anomaly detection capabilities are limited to signature based malware detection.

Let us reconsider the same scenario from an attacker's perspective. The attacker visits the same cafe that offers a free WiFi hotspot and decides to employ basic host, network identification and enumeration tools from the laptop to enumerate all the active devices connected to the Wi-Fi hotspot. From the results, the attacker notices a MAC

address referring to a Nokia smartphone. The attacker know that there is little to no detection capabilities present on an overwhelming majority of smartphone's in use today, so the owner would likely never find out about a successful *man-in-the-middle* attack (MITM). The well-informed attacker creates a successful MITM attack. In the meantime, the smartphone owner accesses the online bank website and enters the login

credentials required to gain access to the banking information. In this scenario, all of the communication between the smartphone and the online bank site is routed through the attacker's machine and the attacker can see the login details in plain text, as well as can capture all the sites accessed by the victim.

Man in the Middle Attack (MITM)

A man-in-the-middle attack intercepts communication between two systems by relaying messages between them. In this attack, the attacker makes an independent connection with both of the victim's machines. The attacker machine forces the traffic between the victim's machines to route through it by sending a false ARP reply to both machines. The attacker can then create new connections and kill existing connections, as well as view and replay anything that is private between the targets machines.

WHAT YOU WILL LEARN

Awareness about the existing security flaw in using Wi-Fi connection

Techniques used by an attacker to perform man-in-the-middle attack on smartphone's

WHAT YOU SHOULD KNOW

It requires medium level knowledge of Wi-Fi security.

Basic level experience in using smartphone for internet access.

STUDY OF MITM ATTACKS AGAINST SMARTPHONE DEVICES

Table 1. List of test devices used in this study

Name	OS
Nokia N 95	S60 3 edition
Windows HTC tilt	Windows Mobile 6.1 Professional CE OS 5.2.19212
Tmobile G1	Android
Apple iPhone 3G S	iPhone OS 3.1

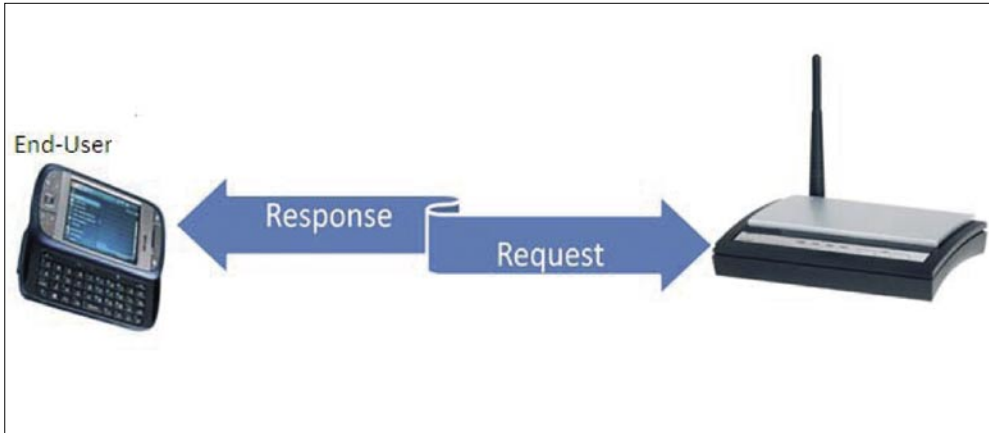


Figure 1. A schematic of wireless connectivity between handset and router

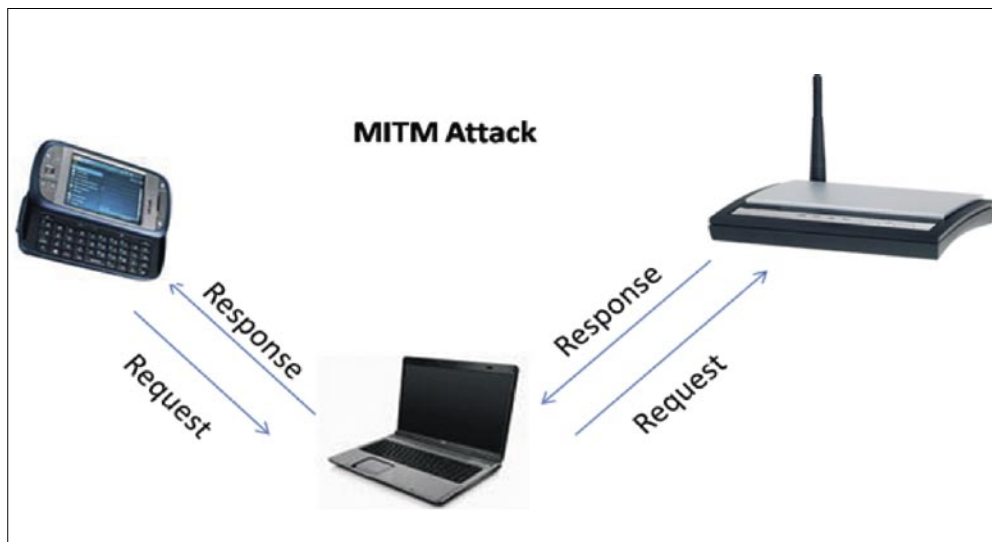


Figure 2. A schematic of successful MITM attack

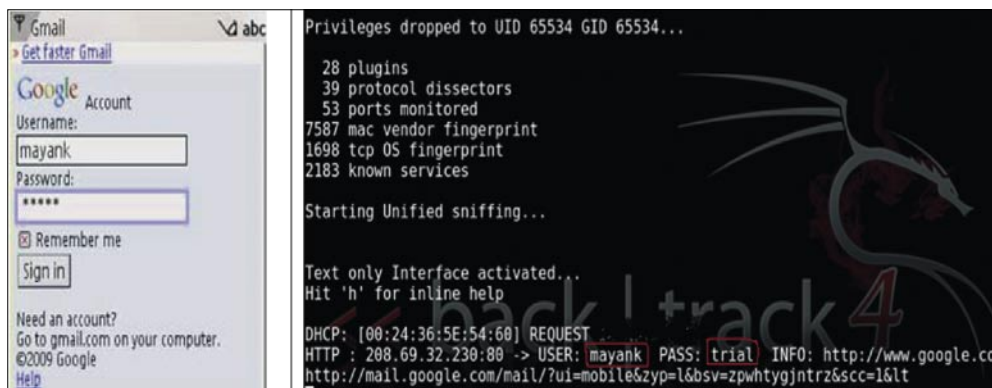


Figure 3. The result from accessing email through browser

In the following section we will discuss the tools used in implementing a MITM based SSL bypass attack.

Tools

The tools mentioned below are just a few of the possible tools that an attacker could use to perform a successful MITM attack and break the security provided by SSL.

Arpspoof

It redirects packets from a target host on the LAN to the intended host on the same LAN. It does so by forging the ARP replies to target host.

SSLStrip

Allows for the transparent hijacking of HTTP traffic on a network, watches for HTTPS links and redirects, and then maps those links into either look-alike HTTP links or homograph-similar HTTPS links. It also supports modes for supplying a favicon that looks like a lock icon, selective logging, and session denial [7].

Etercap or Wireshark

A multipurpose sniffer/interceptor/logging utility for switched LAN's. It is also used to implement MITM attacks in the networked environment [8]. Whereas, Wireshark is a network protocol analyzer and is often used as packet sniffer [9].

MITM Attack Implementation

The graphic displayed in Figure 1 shows the network connection between an end user and the router providing the connection to the Internet before a successful MITM attack has been implemented. Modern day web browsers regularly rely upon SSL certificates to ensure the security of the data included

ATTACK

in the encrypted communication between the browser and a secure website.

As was proven in this year's BlackHat conference presentation *More tricks for defeating SSL* by Moxie Marlinspike who authored the SSLStrip tool [10], it is entirely possible to defeat these protections. The question remains whether this same vulnerability exists in the smartphone, mobile computing world.

Utilizing the tools that were previously mentioned in this study, let us discuss how an attacker performs a MITM based SSL bypass attack. Below, you will find the steps performed by an attacker on the machine to perform a MITM attack

designed to bypass SSL security. It is worth mentioning that these particular tests were performed from Backtrack 4, pre-release, but should be the same when running under Ubuntu or other Linux distribution.

```
echo 1 > /proc/sys/net/ipv4/ip_
forward
```

In this step, the attacker enables IP forwarding on his machine. By default, the value stored in `ip_forward` is 0, which implies that IP forwarding is disabled. Changing this default setting to 1 enables IP forwarding

```
arp spoof -i wlan0 -t #victims ip
address #router ip address
```

ArpSpoof forges ARP replies to the victim's machine. This step assumes that the attacker has already determined the victim's IP address, utilizing some sort of network/port scanning tool

```
iptables -t nat -A PREROUTING -p
tcp --dport 80 -j REDIRECT --to-
ports 10000
```

The above command selects the Network Address Translation (NAT) table and appends the rule to the PREROUTING chain. The rule is set for protocol `tcp` with destination port of 80 and it sets the firewall to redirect all of the traffic coming in on port 80 to port 10000.

```
sslstrip -a -k -f
```

The above command executes the SSLStrip tool and tells it to log all SSL and HTTP traffic to and from the attacker machine.

```
ettercap -T -q -i wlan0 or Wireshark
```

Finally, the attacker starts a sniffer to view the traffic going through his machine. The attacker can use ettercap in text mode (`-T`) to sniff only user name & passwords using (`-q`) flag or can use a tool such as Wireshark.

At this point, the attacker has successfully setup the machine to act as a man-in-the-middle between the unsuspecting smartphone and the Wi-Fi hotspot. Utilizing this method, the attacker has effectively told the victim device to route all traffic through the attacker's machine, and the attacker machine then forwards the requests on to the Wi-Fi hotspot. Since the attacker machine now captures all the traffic from the victim smartphone, the attacker can kill or modify the active connections. The attacker has already run the ssl bypass tool on the machine, thus as soon as the victim accesses any email or online bank website, the login credentials will appear in plain text on the attacker machine.

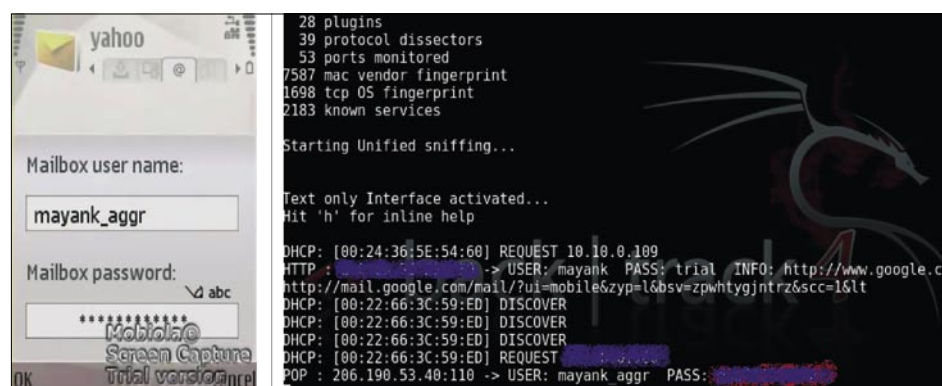


Figure 4. The result from accessing email through mailbox



Figure 5. The result from accessing email through Xpress Mail



Figure 6. The result from accessing email through IE browser

STUDY OF MITM ATTACKS AGAINST SMARTPHONE DEVICES

The Figure 2 shows successful implementation of MITM attack.

Results

The following section discusses the results obtained on the test devices mentioned in Table 1. In all of the test cases, the testing team set out to gain access to email credentials (user name and password) as a means to prove a successful MITM attack against a smartphone device and web server offering SSL security. For each platform, we tested these MITM attacks against different email clients.

Nokia N95

For the N95 we test MITM based SSL bypass attack against three different email clients.

Web Browser

We began by using the Nokia N95 web browser to access email. Fig. 3 consists of a screen shot of the end user device and a screen shot from the attacker's machine. We can see from the screen shot of the attacker's machine that the end user's login credentials are visible. The end user continues to access the Internet unaware of the loss of privacy. The attacker can also use a tool such as "webspY" that will open all of the visited websites by the victim on to the attacker's machine.

Mailbox

The second option to access email is by configuring the mailbox service found within the messages folder on the device. As in Figure 3, Figure 4 consists of screen shots from both the N95 and the attacker's machine. The end user's email credentials are visible on the attacker's machine while the end user is setting up an email account for the first time. The user's login information becomes available on the attacker's machine whenever the user refreshes the mailbox.

Xpress Mail

The final application that is tested on the Nokia N95 is the Xpress Mail application. This application encrypts the data between the device and

the server in a manner that does not rely on SSL encryption between the client and the server. In this scenario, sniffing or SSL bypass methods do not compromise the encryption provided by Xpress Mail. Figure 5 shows the screen shots from both the N95 and the attacker's machine. The screen shot from the N95 shows the account setup step, whereas the screen shot from the attacker's machine shows the sniffed packet that consists of encrypted login credentials.

Windows HTC tilt

For the Tilt, we tested MITM based SSL bypass attacks against three different email clients

Web browser: Internet Explorer

With the Tilt device, we initially accessed email through the web browser. Figure 6 consists of screen shots from the Tilt device and the attacker's machine. The screen shot from the attacker's machine shows the sniffed user name and password from the Tilt device. As expected, the end user is unaware of the fact that the login credentials are intercepted by the attacker's machine. The attacker can also run "webspY" tool to sniff and open all the web pages accessed by Tilt user.

Setup Email option in Messaging

Tilt provides a native option for accessing email on the device. It is similar to mailbox

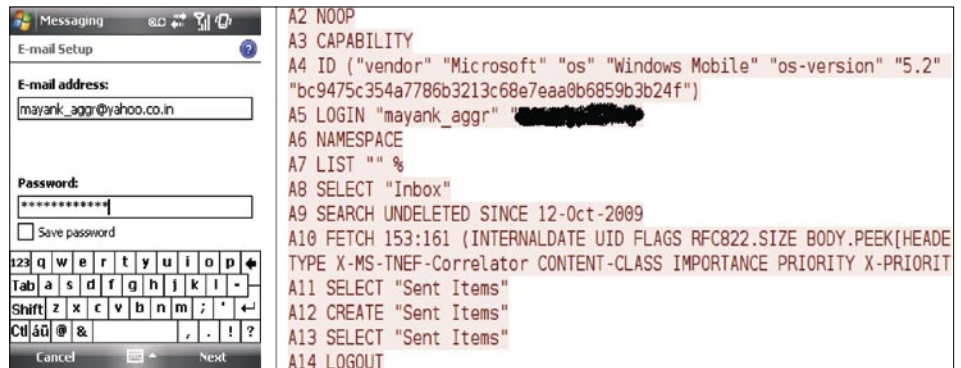


Figure 7. The result from accessing email through Tilt email client

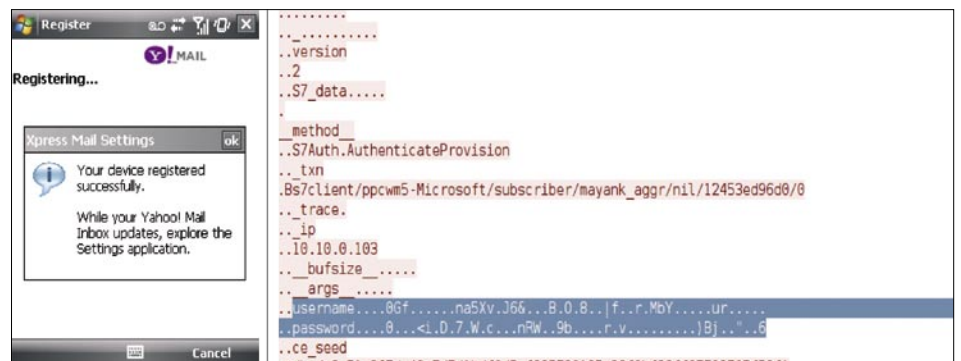


Figure 8. The result from accessing email through Xpress Mail

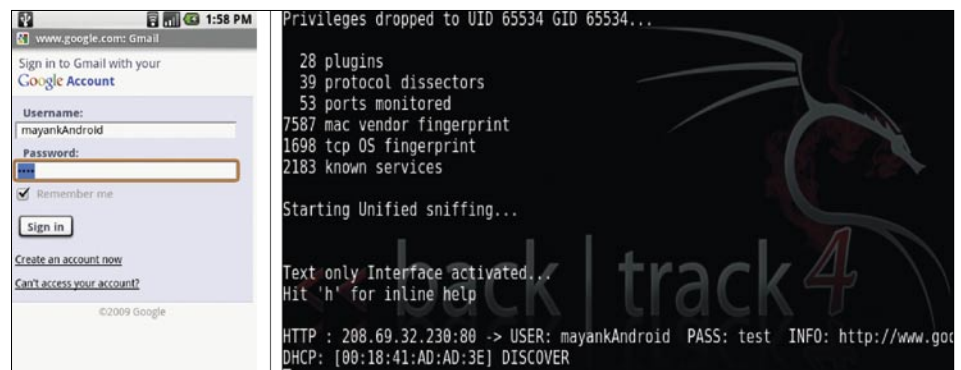


Figure 9. The result from accessing email through Web Browser

ATTACK

option on Nokia N95. Figure 7 shows the results and consists of screen shots from both the Tilt and the attacker's machine. The attacker is using Wireshark to sniff packets and we can see that the user name and password is visible in the captured packet. The Wireshark captures the same information each time the end user refreshes the email client for getting new emails.

Xpress Mail

The third option that we test for accessing email is the Xpress Mail client. The

Xpress Mail application provides end-to-end encryption between the device and the server. Fig. 8 consists of the screen shot from the device and the attacker's machine. As expected, the user name and password are encrypted. However, it cannot prevent interception of end user private information for other websites, i.e. an online banking website from MITM attack.

Android G1

At the time of testing, the Xpress Mail client is not available for Android. Thus, we

conducted our MITM attack tests on two possible options.

Web browser:

Default Android web browser

Figure 9 shows the results obtained by implementing the MITM based SSL bypass attack on an Android device. The result consists of screen shots from both the Android device and the attacker's machine. We can see that the attacker was successful in sniffing the user credentials and the end user remains oblivious of this sniffing attack on an Android smartphone.

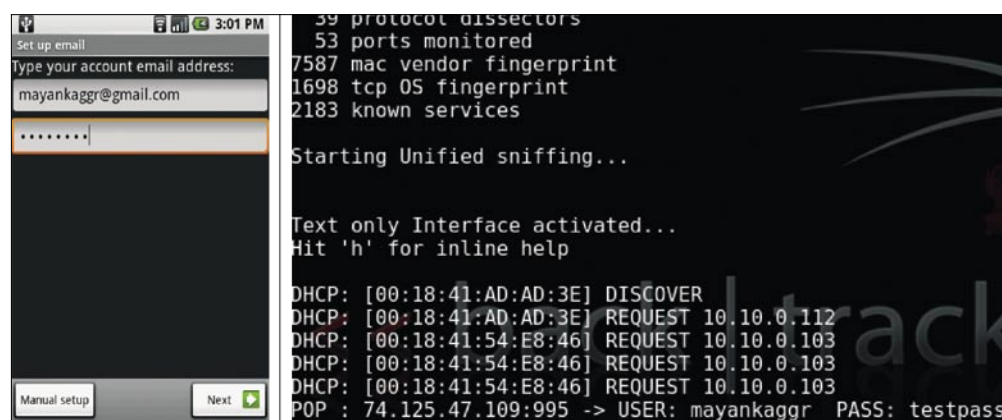


Figure 10. The result from accessing email through Android email client



Figure 11. The result from accessing email through Safari Web browser

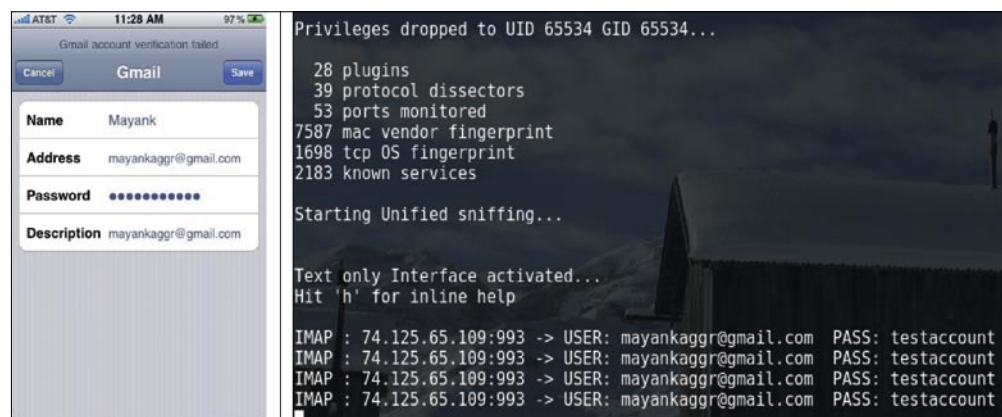


Figure 12. The result from accessing email through iPhone email client

Email Application

The Android smartphone provides an email client similar to Windows Tilt and Nokia N 95 email clients. The Figure 10 above shows email setup screen along with the sniffed credentials at the attacker's machine. The attacker is able to sniff the login credentials, whenever the user refreshes the email client application to update email messages on the smartphone.

Apple iPhone 3G S

The iPhone does not support Xpress Mail client at the time of testing. Therefore, we tested only two ways of accessing email on the iPhone.

Web browser: Safari

The result in Figure 11 shows the successful SSL bypass attack on the iPhone using the Safari web browser to access email.

Email Application

The native email client is the second option that we tested on the iPhone. It is similar to the email client on the Tilt (Windows) or G1 (Android). Figure 12 shows the results of the test, we can see that the username and password is visible on the attacker's machine.

Conclusion

MITM attacks are considered to be a legitimate threat to confidential or private data in the PC side of

On the 'Net

- <http://metrics.admob.com/2009/02/january-metrics-wifi-by-geo-in-the-us-ipod-touch-still-climbing/> [1]
- <http://www.informationweek.com/news/mobility/business/showArticle.jhtml?articleID=218900308> [2]
- <http://www.yankeegroup.com/pressReleaseDetail.do?actionType=getDetailPressRelease&ID=2468> [3]
- <http://www.abiresearch.com/home.jsp> [4]
- <http://www.internetnews.com/mobility/article.php/3835381> [5]
- <http://www.ethicalhacker.net/content/view/31/24/> [6]
- <http://www.thoughtcrime.org/software/sslstrip/> [7]
- <http://ettercap.sourceforge.net/> [8]
- <http://www.wireshark.org/> [9]
- <http://www.blackhat.com/html/bh-usa-09/bh-usa-09-archives.html> [10]
- Download link of Xpressmail for Windows smartphone. http://xpressmail.cingular.com/downloads/sd_ppcwm5/XpressMailPPCWM5Setup.cab [11]
- Download link of Xpressmail for Windows & Symbian smartphone. <http://www.wireless.att.com/businesscenter/solutions/email-messaging/xpress-mail-personal-howto.jsp#step3> [12]

information security. The testing team has adequately shown that with a mobile laptop in a Wi-Fi network, it is possible to intercept communications between the smartphone and the Wi-Fi hotspot.

The testing team was able to perform successful MITM attacks against four different smartphone devices, illustrating that protections provided by SSL can be bypassed and login credentials can be intercepted. This study underscores the fact that the use of publicly available Wi-Fi hotspots should be approached with caution and care should be taken to ensure that confidential or private data is

adequately encrypted, when it becomes necessary to access such data. Where possible, smartphone users should seek out and identify applications that provide adequate encryption technologies to protect confidential or private information. At this point, such applications do exist, but are scarce. When selecting applications to handle sensitive communications, users should search for applications that provide end-to-end encryption between the client application and the end server. Additionally, when dealing with applications that provide access to financial institutions or other sensitive information, the same

precautions should be taken to ensure those communications are encrypted end-to-end. When such applications are not readily available, users must ensure they take necessary precautions to ensure they are only accessing sensitive information over, either, the service provider's internet connection provided from their data plan or from a trusted, secure Wi-Fi network, where available.

Additionally, personal smartphone users and enterprises providing (or allowing) smartphone access into their environments for productivity, should ensure that security software is installed that provides firewall and anti-virus capabilities, at the least. Users and enterprises must begin to treat their smartphone devices with the same care that they do when using their PC's or laptops. The threats, while not as extensive at this point, are quite similar and costly when successful attacks occur. Moreover, as always, as vulnerability/exploit research continues to occur against smartphone devices, so will the number of exploits that translate into successful attacks against smartphone users.

Mayank Aggarwal

is a security researcher at Global Threat Center, SMOBILE Systems where his research focuses on exploiting security loopholes in smartphone's, malware analysis and reverse engineering. He received his MS in network security from The Ohio State University, USA and is a certified ethical hacker (CEH). The author can be contacted at magggarwal@smobilesystems.com

A D V E R T I S E M E N T

WWW.CYBER-RECON.COM

Exceptional Computer Security Training
CompTIA Security+ Training in Stafford, Virginia, USA
Stafford, Virginia Class May 30 - June 4, 2010

Class Limited to 12 students

All Inclusive Training — Materials Yours To Keep After Training

- ASUS Netbook
 - Test Voucher
 - Books and Training Material
 - Catered Meals
- Online Mentored Training Starting April 2, 2010

information@cyber-recon.com

(571) 255-2771



TAM HANNA

Symmetric Secrets

Difficulty



An article in the last issue of this magazine introduced you to the various forms and applications of cryptography and cryptology.

The first topic which usually gets covered is symmetric cryptography – if maths mavens like the JKU Linz’s Wolfgang Windsteiger follow this structure, why not adapt it for our personal usage?

Why symmetry?

When talking about symmetry, users usually start to think about geometry: symmetric lines, bodies, boobs or other entities. In cryptography, symmetry means something completely different: it means that the key used for encrypting and decrypting the data is the same. Figure 1 illustrates the process.

Substitution ciphers

Before looking at real encryption algorithms, let’s start off with a simple example which can be used to clarify some concepts. ROT13 is a so-called substitution cipher, which, to some extent is similar to all other encryption algorithms.

It works by replacing each letter with its *equivalent*, as shown in Figure 2.

A sample code processing a single character is embedded in sample implementation. It is written in C and works on all systems which use ASCII code for representing characters: see Listing 1.

Should you now want to encrypt more than byte (aka eight bits), the algorithm is applied again and again unmodified. Each bit *block* is processed the same way, which would make this a so-called block cipher.

Unfortunately, all substitution ciphers are vulnerable to statistical attacks (except for Verdun algorithms using so-called one time pads). Statistical attacks make use of known properties of human languages in regards to character distribution: for example, the letter e is used more commonly, than, say z.

As each and every instance of e gets transformed to the same target letter, analyzing the amount of occurrences for each *code letter* allows you to gain valuable insights into how the code is structured.

Verdun’s one time pads

Conspiracy heads have probably heard of one time pads in relationship to numbers stations and espionage – they are used in these scenarios as they are impossible to break if implemented correctly. The idea behind one time pads is that they provide a stream of random bytes, which are known to both sender and receiver (see Figure 3).

These bits are used to encrypt and decrypt the message, and the sequence is discarded afterwards. This means that each message bit has its own, unique *key bit* – assuming that the stream is perfectly random, there is no block-wise reuse.

As every one time pad stream is used but once, the only way to really find out more about the message is a large-scale brute-force attack (which tends to be impractical).

Usually, the bits are simply XORred together – the secrecy of the transmission is not guaranteed

WHAT YOU WILL LEARN

What symmetric encryption algorithms are and some practical examples

WHAT YOU SHOULD KNOW

A C-like programming language

Basic understanding of cryptographya

by a complex algorithm, but rather by the unique (and most likely very long) key used.

Verdun's algorithm is both the most simple and most complex stream cipher: while rarely used in IT due to the difficulty in generating, distributing and managing OTP data, it clearly shows the difference between a stream cipher and the aforementioned block cipher. In a block cipher, the same key *sequence* is used for multiple bits – in a stream cipher, each message bit gets its unique key bit.

AES

AES, or *Advanced Encryption Standard*, is considered the classic example of a symmetric block cipher. However, it is recommended by the US NIST – as an individual who always has a healthy distrust of governments which tax their citizens more than 25%, I personally am pretty sure that they have some kind of special attack against it.

Thus, AES will be replaced by BlowFish in this article for both novelty and security value. Individuals looking for further info on AES can do so online with relative ease.

BlowFish

BlowFish is a completely open-source and unpatented algorithm created by a security researcher called Bruce Schneier. It was first published in 1993, and has not been proven insecure so far. This could of course have to do with AES getting more attention by independent researchers – but, as we say in Austria, you always have to take a risk.

BlowFish works with a fixed block size of 64 bits, its key can be 32 bits to 448 bits long (default 128), as long as the length is a multiple of 8.

WARNING
Free, proven implementations are available from Bruce Schneier's web site for almost any language. Don't reinvent the wheel...

Bruce Schneier's default implementation (<http://www.schneier.com/blowfish.html>) looks as follows: see Listing 2.

First of all, the initialize blowfish routine opens an included file called

Blowfish.dat. It contains hexadecimal digits of P_i , which are used to initialize the P and S arrays with initial data. The P array entries are then XORred with the key, and the algorithm is initialized by encrypting an all-zero block.

The result replaces the P1 and P2 slots, which are encrypted again in order to replace P3 and P4, and so on. In the end, the entire P and S entries are replaced – which requires the encryption of about 4KB whenever a new key is loaded into the system.

The physically first function, called F, splits the 32bit input into four 8bit words. These words are then fed into a substitution box, and finally return a result. Substitution boxes are like lookup tables which transform an input into an output – in the case of BlowFish, four substitution boxes are used. They accept an 8bit input, and produce a 32bit

output. These outputs are then XORred to produce the final output.

Encryption and decryption is handled by the two relatively similar functions, who each want a pointer to a left and a right 32bit word of either plain or ciphertext. The main difference is the sequence in which the P array slots are used.

Essentially, BlowFish runs the encryption process 16 times. Each time, a different P slot is XORred onto the left and right parts (using the F substitution function with the already-encrypted left part rather than another P array entry), which are then exchanged with one another. When the process is done, the last two P slots are simply XORred into the result halves.

As XOR can be *undone* with relative ease as long as the sequence and data used are known, the reversing essentially takes the same lines.

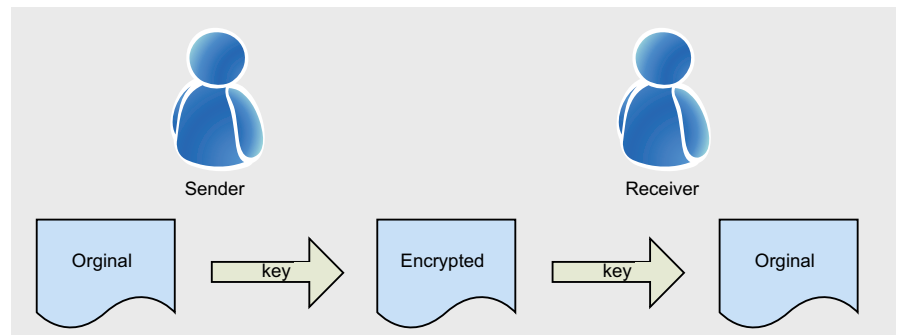


Figure 1. In a symmetric algorithm, the same key is used for encryption and decryption

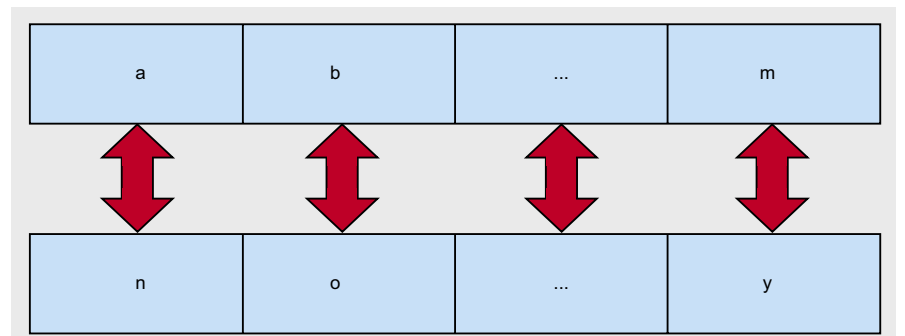


Figure 2. ROT13 made easy. A becomes N, and N becomes A...

ZDXWWW	EJKAWO	FECIFE	WSNZIP	PXPKIY	URMZHI	JZTLBC	YLGDYJ
HTSVTV	RYYEG	EXNCGA	GGQVRF	FHZCIB	EWLGGR	BZXQDQ	DGGIAK
YHJYEQ	TDLQQT	HZBSIZ	IRZDYS	RBYJFZ	AIRCWI	UCVXTW	YKPQMK
CKHVEX	VXYVCS	WOGAAZ	OUVVON	GCNEVR	LMBLYB	SBDCDC	PCGVJX
QXAUIP	PXZQIJ	JIUWYH	COVWMJ	UZOJHL	DWHPER	UBSRUJ	HGAAPR
CRWVHI	FRNTQW	AJVWRT	ACAARD	OZKIIB	VIQGBK	IJCWHF	GTTSS
EXFIPJ	KICASQ	IOUQTP	ZSGXGH	YTYCTI	BAZSTN	JKMFXI	RERYWE

Figure 3. Sample one-time pad data (from http://en.wikipedia.org/wiki/File:One-time_pad.svg)

Listing 1. Rot13, single character processing

```
char rot13(char victim)
{
    if((victim>='A'&&victim<='M') || (victim>='a'&&victim<='m'))
        return victim+13;
    else if((victim>='N'&&victim<='Z') || (victim>='n'&&victim<='
        z'))
        return victim-13;
    else //If victim cannot be rotted
        return victim;
}
```

Listing 2a. Blowfish reference implementation

```
#ifndef little_endian /* Eg: Intel */
#include <dos.h>
#include <graphics.h>
#include <io.h>
#endif

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#ifndef little_endian /* Eg: Intel */
#include <alloc.h>
#endif

#include <ctype.h>

#ifndef little_endian /* Eg: Intel */
#include <dir.h>
#include <bios.h>
#endif

#ifndef big_endian
#include <Types.h>
#endif

#include "Blowfish.h"

#define N 16
#define noErr 0
#define DATAERROR -1
#define KEYBYTES 8
#define subkeyfilename "Blowfish.dat"

unsigned long P[N + 2];
unsigned long S[4][256];
FILE* SubkeyFile;

short opensubkeyfile(void) /* read only */
{
    short error;

    error = noErr;

    if((SubkeyFile = fopen(subkeyfilename,"rb")) == NULL) {
        error = DATAERROR;
    }

    return error;
}

unsigned long F(unsigned long x)
```

```
{
    unsigned short a;
    unsigned short b;
    unsigned short c;
    unsigned short d;
    unsigned long y;

    d = x & 0x00FF;
    x >>= 8;
    c = x & 0x00FF;
    x >>= 8;
    b = x & 0x00FF;
    x >>= 8;
    a = x & 0x00FF;
    //y = ((S[0][a] + S[1][b]) ^ S[2][c]) + S[3][d];
    y = S[0][a] + S[1][b];
    y = y ^ S[2][c];
    y = y + S[3][d];

    return y;
}

void Blowfish_encipher(unsigned long *xl, unsigned long *xr)
{
    unsigned long Xl;
    unsigned long Xr;
    unsigned long temp;
    short i;

    Xl = *xl;
    Xr = *xr;

    for (i = 0; i < N; ++i) {
        Xl = Xl ^ P[i];
        Xr = F(Xl) ^ Xr;

        temp = Xl;
        Xl = Xr;
        Xr = temp;
    }

    temp = Xl;
    Xl = Xr;
    Xr = temp;

    Xr = Xr ^ P[N];
    Xl = Xl ^ P[N + 1];

    *xl = Xl;
    *xr = Xr;
}

void Blowfish_decipher(unsigned long *xl, unsigned long *xr)
{
    unsigned long Xl;
    unsigned long Xr;
    unsigned long temp;
    short i;

    Xl = *xl;
    Xr = *xr;

    for (i = N + 1; i > 1; --i) {
        Xl = Xl ^ P[i];
        Xr = F(Xl) ^ Xr;
    }
}
```

Looking at the example above, one can quickly see the main advantage of BlowFish: it needs less than 5KB of RAM, and doesn't use any kind of complex arithmetical operations. This makes it ideally suited for tasks where resources are limited.

Unfortunately, BlowFish also has its weaknesses. For example, encryption accelerators (and out-of-order execution hardware) found in modern processors tend to accelerate AES, but not BlowFish – which means that the current speed

advantage of the algorithm will likely get eroded over time.

RC4

RC4 was developed by RSA Security in 1987, and was considered a trade secret.

Listing 2b. Blowfish reference implementation

```

/* Exchange Xl and Xr */
temp = Xl;
Xl = Xr;
Xr = temp;
}

/* Exchange Xl and Xr */
temp = Xl;
Xl = Xr;
Xr = temp;

Xr = Xr ^ P[1];
Xl = Xl ^ P[0];

*xl = Xl;
*xr = Xr;
}

short InitializeBlowfish(char key[], short keybytes)
{
    short    i;
    short    j;
    short    k;
    short    error;
    short    numread;
    unsigned long data;
    unsigned long datal;
    unsigned long datar;

    /* First, open the file containing the array initialization data */
    error = opensubkeyfile();
    if (error == noErr) {
        for (i = 0; i < N + 2; ++i) {
            numread = fread(&data, 4, 1, SubkeyFile);
#ifdef little_endian    /* Eg: Intel We want to
                        process things in byte */
                /* order, not as rearranged in a longword */
                data = ((data & 0xFF000000) >> 24) |
                    ((data & 0x00FF0000) >> 8) |
                    ((data & 0x0000FF00) << 8) |
                    ((data & 0x000000FF) << 24);
#endif

            if (numread != 1) {
                return DATAERROR;
            } else {
                P[i] = data;
            }
        }

        for (i = 0; i < 4; ++i) {
            for (j = 0; j < 256; ++j) {
                numread = fread(&data, 4, 1, SubkeyFile);

#ifdef little_endian    /* Eg: Intel We want to
                        process things in byte */
                    /* order, not as rearranged in a longword */
                    data = ((data & 0xFF000000) >> 24) |
                        ((data & 0x00FF0000) >> 8) |
                        ((data & 0x0000FF00) << 8) |
                        ((data & 0x000000FF) << 24);
                    #endif

                    if (numread != 1) {
                        return DATAERROR;
                    } else {
                        S[i][j] = data;
                    }
                }
            }

            fclose(SubkeyFile);

            j = 0;
            for (i = 0; i < N + 2; ++i) {
                data = 0x00000000;
                for (k = 0; k < 4; ++k) {
                    data = (data << 8) | key[j];
                    j = j + 1;
                    if (j >= keybytes) {
                        j = 0;
                    }
                }
                P[i] = P[i] ^ data;
            }

            datal = 0x00000000;
            datar = 0x00000000;

            for (i = 0; i < N + 2; i += 2) {
                Blowfish_encipher(&datal, &datar);

                P[i] = datal;
                P[i + 1] = datar;
            }

            for (i = 0; i < 4; ++i) {
                for (j = 0; j < 256; j += 2) {

                    Blowfish_encipher(&datal, &datar);

                    S[i][j] = datal;
                    S[i][j + 1] = datar;
                }
            }
        } else {
            printf("Unable to open subkey initialization file : %d\
                n", error);
        }

        return error;
    }
}

```

However, a leaked description of the code was posted to a mailing list in 1994, which was proven to be authentic by comparing the output to the one generated by RSA-licensed software products. As RC4 is still a trademark, the algorithm is often referred as ARCFOUR or ARC4, which

stands for Alleged RC4 – the folks at RSA still haven't released their implementation.

Nevertheless, RSA can be considered THE stream cipher: even though attack vectors are known, it is used in loads of open standards like WEP, WPA, BitTorrent and TLS.

Listing 3. An open-source implementation of RC4 (via <http://en.wikipedia.org/wiki/RC4>)

```
unsigned char S[256];
unsigned int i, j;

void swap(unsigned char *s, unsigned int i, unsigned int j) {
    unsigned char temp = s[i];
    s[i] = s[j];
    s[j] = temp;
}

/* KSA */
void rc4_init(unsigned char *key, unsigned int key_length) {
    for (i = 0; i < 256; i++)
        S[i] = i;

    for (i = j = 0; i < 256; i++) {
        j = (j + key[i % key_length] + S[i]) & 255;
        swap(S, i, j);
    }

    i = j = 0;
}

/* PRGA */
unsigned char rc4_output() {
    i = (i + 1) & 255;
    j = (j + S[i]) & 255;

    swap(S, i, j);

    return S[(S[i] + S[j]) & 255];
}

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define ARRAY_SIZE(a) (sizeof(a)/sizeof(a[0]))

int main() {
    unsigned char *test_vectors[][2] =
    {
        {"Key", "Plaintext"},
        {"Wiki", "pedia"},
        {"Secret", "Attack at dawn"}
    };

    int x;
    for (x = 0; x < ARRAY_SIZE(test_vectors); x++) {
        int y;
        rc4_init(test_vectors[x][0], strlen((char*)test_vectors[x][0]));

        for (y = 0; y < strlen((char*)test_vectors[x][1]); y++)
            printf("%02X", test_vectors[x][1][y] ^ rc4_output());
        printf("\n");
    }
    return 0;
}
```

WARNING

RC4 is EXTREMELY SENSITIVE to implementation errors! DO NOT implement it yourself under ANY CIRCUMSTANCES unless you are a cryptography expert!!

RC4 is another algorithm which makes extensive use of the symmetric properties of the XOR function (anyone see a pattern here?). It generates a pseudo-random stream of bits out of the key – essentially, we are looking at a function dependant on nothing but the key and the position of the bit *i*.

These bits are then XORred together in a fashion similar to Verdun's algorithm outlined above – the only difference is that the bit sequence does not come from a one time pad, but rather from the key.

The KSA function (for key-scheduling algorithm) initializes the S array, while mixing in key bits. The PRGA algorithm then handles the actual generation of a random sequence.

Essentially, the security of the process depends on the unpredictability of the key sequence: if the attacker knows the plain-text equivalent of a few bytes at the beginning of the encrypted message, he can try to guess the key sequence from them.

For example, protocols like SMTP always transfer a fixed header – this is unhealthy to say the least...

Conclusion

Symmetric algorithms are used extremely often – they provide a good tradeoff between speed and security. However, the issue of the key transfer remains: if an attacker managed to get hold of the key, he can encrypt and decrypt messages at will.

Asymmetric algorithms solve this problem by having two separate keys: a private and a public one. More on that in the next issue...

Tam Hanna

Tam Hanna has been in the mobile computing industry since the days of the Palm Ilc. He develops applications for handhelds/smartphones and runs for news sites about mobile computing:
<http://tamspalm.tamoggemon.com>
<http://tamspc.tamoggemon.com>
<http://tamss60.tamoggemon.com>
<http://tamswms.tamoggemon.com>
If you have any questions regarding the article, email author at: tamhan@tamoggemon.com

Passware Password Recovery Kit Forensic 9.7

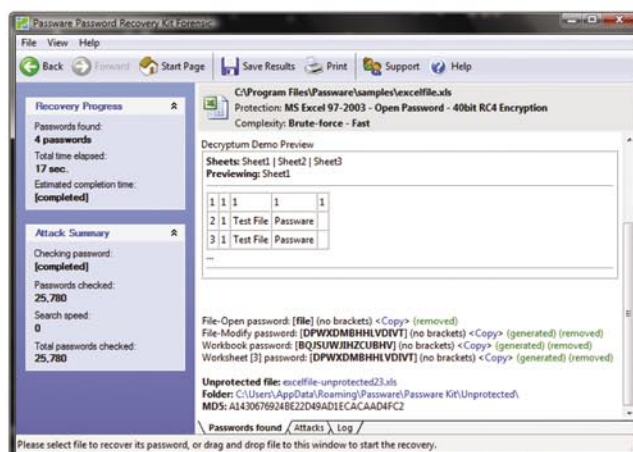
A Complete Password Recovery and E-Discovery Solution for Computer Forensics

Passware Inc., has combined all its proven password recovery tools and encryption detection technology, and released a complete evidence discovery solution for computer forensics.

All password recovery and decryption algorithms that Passware has developed and improved for more than 12 years are now available in the all-in-one **Passware Password Recovery Kit Forensic**.

Key Features

- Scans computers and network for password-protected files
- Recovers passwords for **180+ file types**
- Unlocks hard drives protected with **BitLocker** and **TrueCrypt**
- Retrieves electronic evidence in a matter of minutes from a Windows Desktop Search Database
- Includes **Portable Version** that runs from a USB thumb drive and recovers passwords without installation on a target PC



Let me just say "well done". Excellent software, excellent support. I have watched this software evolve over the past six year. World class stuff.
Craig Vogel, myComputerGuy, inc.



Advanced Features

- Recovers many password types **instantly**
- Accelerates password recovery with distributed computing (**Distributed Password Recovery**)
- Uses **multiple-core CPUs** and **nVidia GPUs** efficiently to speed up the password recovery process by 3,500%
- Uses **Tableau TACC hardware accelerators** to speed up the password recovery process by up to 25 times
- Provides 8 different password recovery attacks (and any combination of them) with an easy-to-use setup wizard and drag & drop attacks editor
- Provides detailed reports with **MD5 hash values**



For additional information, please visit:
www.lostpassword.com/kit-forensic.htm

Passware Inc.
 800 West El Camino Real, Suite 180
 Mountain View CA 94040

Contacts
 Nataly Koukoushkina
media@lostpassword.com
 Phone: +1 (650) 450-4607
 (Sales calls only)



FLORIAN EICHELBERGER

ICQ Forensics

Difficulty



Being around for 14 years, the Mirabilis (now AOL) ICQ Client is probably one of the more widely used Instant Messenger Clients.

Having been used by that many people also includes malevolent people and often forensic examiners face the challenge of having to reconstruct the activity's of a user.

The forensic evaluation this part of the IM forensic evaluation series covers, is the original ICQ Client offered by AOL on www.icq.com in version 6.5 (6.5.0.1042). Please see table_1 for important Directories and filenames.

Please note: In January the version 7.0 of ICQ was released but although the files are still in the same directories as for 6.5 the format was changed and some details added and modified.

The last paragraph of this article deals with those modifications. If you're just interested in ICQ 7, you should nonetheless read the whole article as most of it applies to ICQ 7 as well.

Please note (2): As neither the Microsoft .MDB Format nor the format of the binary ICQ data has been made available, the information in the article was gathered through reverse engineering and testing. This was collected carefully but because of the nature of the work, errors cannot 100% ruled out.

ICQ 6.x stores Application properties, the users contact list and the messages (of all kinds like IM Text messages, SMS, etc.) in Access 2000/2003 format. This format can easily be read by either *Microsoft Access* or freeware Alternatives like *MD Viewer Plus*[1]. In fact those files also store certain forensically relevant

timestamps and artifacts in binary form which will be covered later on in this article.

Application Settings and Password

If you can find a file called `<icqnr.fb>`, the existence of this file means the ICQ connected to the server at least once and although the format is undocumented, it seems to contain the contact list downloaded at the last successful connect to the ICQ servers.

The first file, of the 3 existing important database ones, I will discuss in here, is the application.mdb file containing not only the full application settings but a forensically relevant timestamp about the last usage of the ICQ Client.

In general, to extract the binary data of the found .mdb files, they were opened by *MDB Viewer Plus* and the table(s) exported to xml format. Although there are many ways of extraction information from an Access .mdb file, this way was chosen as this can be followed rather easily. For regular examinations or mass-examinations of files one would most probably develop some software to extract the information and to automate certain steps.

The binary information will be output in BASE64 encoded[2] form and needs to be converted back to its original form. You can do this by using one of the various decoding utilities for example the `uudecode -m` command

WHAT YOU WILL LEARN

Where to find the users relevant ICQ files, what can be found, how it looks like and what it shows you.

WHAT YOU SHOULD KNOW

Basic Windows Knowledge,
Basic DB Knowledge,

Basic computer forensic knowledge.

on linux and MacOSX or download a version of uudecode for windows[3] The forensically important part of the application.mdb is the records table and within there, the most important section of the exported .xml file is called RecentOwners.

The RecentOwners section (see Figure 1, RecentOwners with highlighted LastLoginTimestamp) covers most of the main ICQ users settings like the ICQ Nr, the displayed name, the email address used for this account, the screenname, the name that is shown to other users and the LastLoginTime (highlighted on the screenshot).

This timestamp is saved as a Microsoft Variant time format[4] and you need to do use some software to convert it into a normal human readable timestamp.

A Proof of Concept Code you can use to convert the Variant Timestamps found in the ICQ files to human readable dates can be downloaded[10], the tool is self-explanatory. Credits go to Ludwig Ertl for quickly coming up with it.

The next important file is called owner.mdb and contains two tables records and Users.

The records table of the the owner.mdb file contains various timestamps, user details once more and if the user selected to save it, the MD5 hash[5] of the password.

Saving the password locally is enabled by default.

The important parts of this file (in the exported xml) are the following sections:

- "\"
- "GlobalStatsReporter"
- "Details"
- "NatDetect"

The first section "1" (this is the name in the DB) contains the users connection settings, like the status to be set online, if the password was saved, and the base64 encoded hash of the MD5 hash of the users password. You can see the decoded hash on Figure 5 ("1" section with highlighted encoded Password Hash).

Reverse Engineering of the enclosed coolcore49.dll revealed that 2 hashes are saved into the config file (in our example starting with "CC03") both being unsalted MD5 hashes, one of the password entered and the second one of the password entered after it was converted to lowercase.

As an unsalted MD5 hash of a password converted to lowercase should be considered insecure, the

password, up to about 8-9 digits, might be revealed within seconds using RainbowTables[6].

This facts may add additional benefits to the investigation as people tend to use the same password more than once and this, if revealed, will provide full access to the account.

As for the GlobalStatsReporter the relevant information is the

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789AB	CDEF
0x00	1000	0000	0A00	0000	4C00	6900	7300	7400	L.i.s.t.
0x10	0000	1200	0000	0100	0000	0800	1400	0000
0x20	3500	3600	3000	3500	3000	3600	3800	3600	5.6.0.5.	.
0x30	3300	0000	1000	0000	1800	0000	4400	6900	3.....	D.i.
0x40	7300	7000	6C00	6100	7900	4E00	6100	6D00	s.p.l.a.y.N.a.m.	
0x50	6500	0000	0000	0000	0800	1C00	0000	6600	e.....	f.
0x60	6F00	7200	6500	6E00	7300	6900	6300	5F00	o.r.e.n.s.i.c..	.
0x70	7400	6500	7300	7400	0000	1600	0000	4C00	L.
0x80	6F00	6700	6900	6E00	4500	6D00	6100	6900	o.g.i.n.E.m.a.i.	
0x90	6C00	0000	0000	0000	0800	2000	0000	6100	l.....	a.
0xA0	7200	7400	6900	6300	6C00	6500	4000	7200	r.t.i.c.l.e.@.	.
0xB0	7500	6C00	7400	2E00	6100	7400	0000	1C00
0xC0	0000	4C00	6100	7300	7400	4C00	6F00	6700	..L.a.s.t.L.o.g.	
0xD0	6900	6E00	5400	6900	6D00	6500	0000	0000	i.n.T.i.m.e.....	
0xE0	0000	0700	63D4	P0B7	9D9E	E340	1600	0000cÔö·□žã@.
0xF0	5300	6300	7200	6500	6500	6E00	4E00	6100	S.c.r.e.e.n.N.a.	
0x0100	6D00	6500	0000	0000	0000	0800	1400	0000	m.e.....
0x0110	3500	3600	3000	3500	3000	3600	3800	3600	5.6.0.5.0.	.
0x0120	3300	0000	0000	0000	1400	0000	4C00	6100	L.a.
0x0130	7300	7400	4F00	7700	6E00	6500	7200	0000	s.t.O.w.n.e.r...	
0x0140	0000	0000	0800	1400	0000	3500	3600	3000	5.6.0.
0x0150	3500	3000	3600	3800	3600	3300	0000	0000	5.0.
0x0160	0000								..	

Figure 1. „RecentOwners” with highlighted LastLoginTimestamp

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789AB	CDEF
0x00	1000	0000	2A00	0000	4C00	6F00	6700	6F00	L.o.g.o.
0x10	6E00	4C00	6100	7300	7400	5300	6500	7300	n.L.a.s.t.S.e.s.	
0x20	7300	6900	6F00	6E00	5400	6900	6D00	6500	s.i.o.n.T.i.m.e.	
0x30	0000	0000	0000	0300	F1C5	3F4E	0000	0000	NA?E
0x40										

Figure 2. "LogonLastSessionTime" with highlighted Timestamp

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	1011	1213	1415	1617	1819	1A1B	1C1D	1E1F	0123456789ABCDEF0123456789ABCDEF	
0x00	1000	0000	0A00	0000	4A00	6F00	6200	7300	0000	1100	0000	0100	0000	1000	0000	0000	U.o.b.s.....
0x10	0000	1900	0000	4C00	6100	7300	7400	5500	7000	6400	6100	7400	6500	6400	0000	0000	L.a.s.t.U.p.d.a.t.e.d.....
0x20	0000	0700	00E5	3EDE	785C	E340	1800	0000	4100	6300	6300	6F00	7500	6800	7400	5400	öb-ešö...A.c.c.o.u.n.t.
0x30	7900	7000	6500	0000	0000	0000	0300	0000	0000	1800	0000	5600	6100	6C00	6900	6400	y.p.e.....V.a.l.i.d.
0x40	6100	7400	6500	6400	4500	6000	6100	6900	6000	0000	0000	0000	0000	0000	0000	6100	a.t.e.d.E.m.a.i.l.....e.
0x50	7200	7400	6900	6300	6C00	6500	4000	7200	7500	6C00	7400	2B00	6100	7400	0000	1600	r.t.i.c.l.e.@
0x60	0000	5000	7200	6900	7400	6100	7400	6500	4800	6500	7900	0000	0000	0000	0800	1200P.r.i.v.a.t.e.K.e.y.....
0x70	0000	8BC8	4A24	7AA3	453A	9435	F9D8	D722	8E49	0000	1400	0000	0000	6800	6600	6F00KJ\$st "5b@".....K.n.f.o.
0x80	4C00	6500	7600	6500	6C00	0000	0000	0000	0300	0300	0000	1A00	0000	5000	7200	6900	L.e.v.e.l.....P.r.i.
0x90	7600	6100	6300	7900	4C00	6500	7600	6500	6C00	0000	0000	0000	0300	9100	0000	1200	v.a.s.y.L.e.v.e.l.....
0xA0	0000	4800	6900	6300	6800	4800	6100	6F00	4500	0000	0000	0000	0800	1C00	0000	4400N.i.c.k.N.a.m.e.....f.
0xB0	6F00	7200	6500	6800	7300	6900	6200	5F00	7400	6500	7200	7400	0000	0B00	0000	4500	o.r.e.n.s.i.c.....E.
0xC0	6E00	6100	6900	6C00	7300	0000	1100	0000	0100	0000	1000	0000	1400	0000	4900	7300	m.a.i.l.s.....K.s.
0xD0	5000	7200	6900	6D00	6100	7200	7900	0000	0000	0000	0800	FFFF	1000	0000	4100	6400	P.r.i.m.a.r.y.....y.p.....A.d.
0xE0	6400	7200	6500	7300	7300	0000	0000	0000	0800	2000	0000	6100	7200	7400	6900	6300	d.r.e.s.s.....A.F.t.i.c.
0xF0	6000	6500	4000	7200	7500	6C00	7400	2B00	6100	7400	0000	0000	0000	1800	0000	5200	l.e.@
0x0100	6500	6700	6900	7300	7400	6500	7200	6500	6400	4800	6100	6000	6500	0000	0000	0000	e.g.i.s.t.e.r.e.d.N.a.m.e.....
0x0110	0800	1400	0000	3500	3600	3000	3500	3000	3600	3800	3600	3300	0000	0000	0000	00005.6.0.5.0.

Figure 3. "Details section" with full example data

LogonLastSessionTime as shown on the Figure 2 (LogonLastSessionTime with highlighted Timestamp), which is followed by a UnixTimestamp[7] ("3F4B C5F1" in our example) that can easily be translated into a human readable date by various offline or online tools[8].

Those unix timestamps will be used throughout the ICQ database files.

Going on with the Details section, this contains the users details (including a yet unknown private key) as well as the email address used to register the ICQ number.

(see Figure 3 (Details section) for the full example data).

The next being NatDetect, this section is somehow important as this unix timestamp (see Figure 4, (NetDetect section with highlighted timestamp) shows the last time ICQ tried to connect to the server and figure out the type of NAT, Network Address Translation connecting the user to the internet, there might be.

The last section in the xml file being OwnerStatistics contains another unix

Timestamp that shows the last time the ICQ client tried to collect Owner Statistics.

Reverse Engineering of the ICQ Client did not fully reveal the reason and functionality behind this, but it seems to only happen when the ICQ client is active and thus gives another hint on when this ICQ client was last used.

User Messages and Contact List

The Users table of the the owner.mdb file contains the users that have been added to the contact list with their user details. What is interesting from a forensic perspective is the fact that every user that has been added to the list will remain on the list, even if the user was deleted from the contact list within the ICQ Application.

This deleted users messages will also remain in the later discussed messages.mdb.

The message and chat history of the users from the contact list are stored in the messages.mdb which is mostly human readable text and consists of 4 tables, from which only 3 are of forensic use. As per default, full logging of all messages is enabled in this ICQ client and even if it was turned off, some messages still are saved to the file although not visible by a Access or MDB Viewer Plus. This will be covered in the last part of the article.

Those 3 tables are:

- "ChatHistory"
- "Messages"
- "Users"

Starting with chatHistory, as seen on Figure 6 (Content of a Chat-History table) stores the ICQ Nr.(as UID) of the users that where in contact with the local ICQ user and an unique id in the to column.

This is undocumented but might be part of some kind of anonymity schema.

Those to ids are used in the Messages table.

The Users table contains again the ICQ Nr. and the name of the users receiving and sending messages to the local ICQ user.

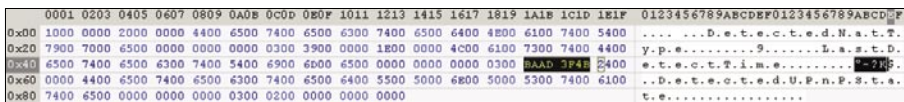


Figure 4. "NetDetect section" with highlighted timestamp

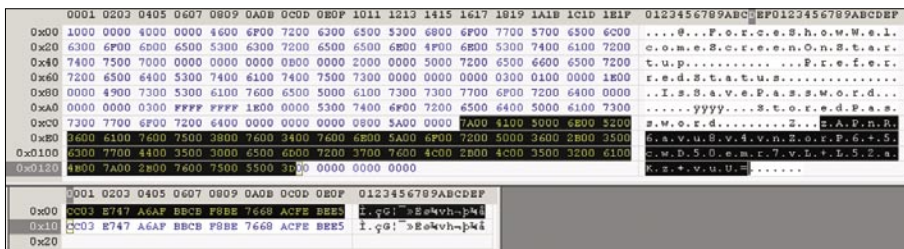


Figure 5. "\t" section with highlighted encoded Password Hash

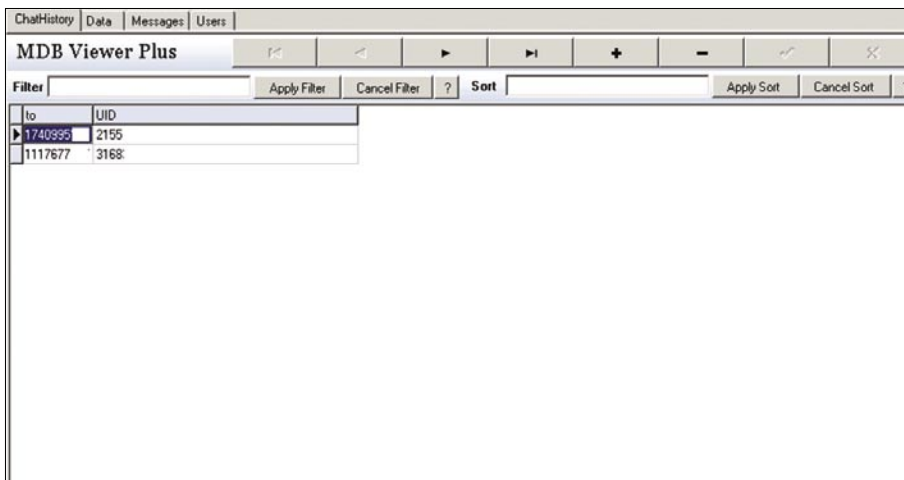


Figure 6. Content of a Chat-History table

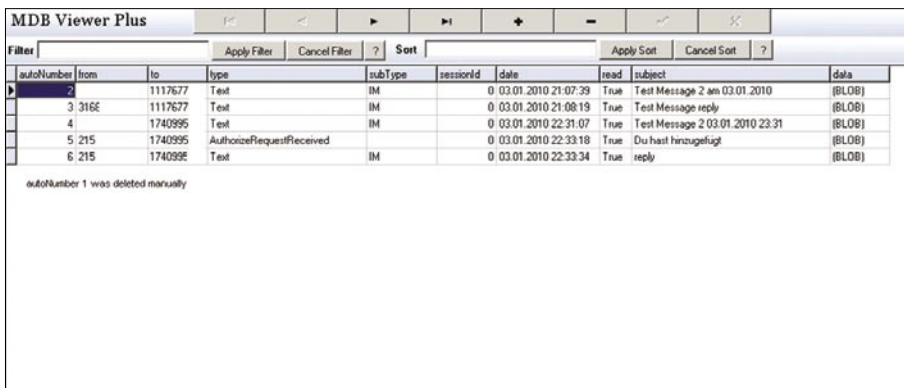
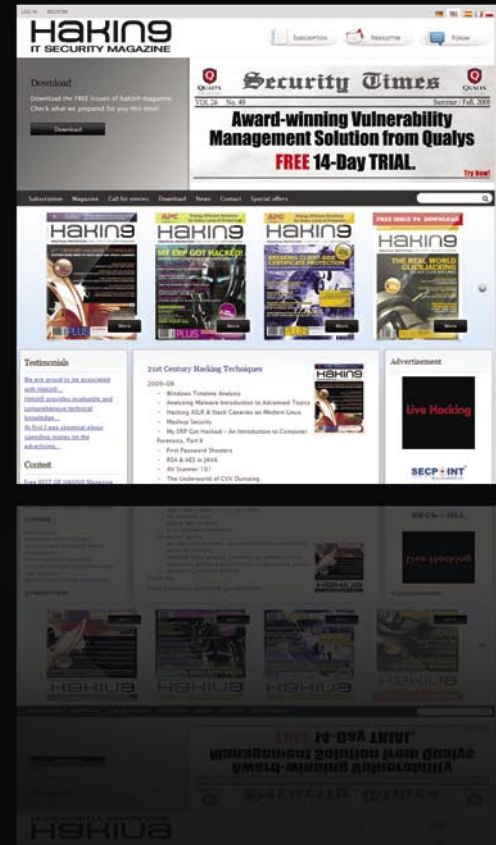


Figure 7. Content of a Messages table

**VISIT OUR
WEBSITE**



What you will find?

**materials for articles:
listings, additional
documentation, tools**

**the most interesting
articles to download**

**free issues to
download**

**information
on the upcoming
issue**

WWW.HAKING9.ORG/EN

Table of directories and files (table 1)

Default installation path:(replace 6.5 with 7 for ICQ7)

- XP: %ProgramFiles%\ICQ6.5
- Vista: %ProgramFiles%\ICQ6.5

Default path of the history and application settings .mdb files:

Note

There is a separate directory for every ICQ Nr. used. containing these files.

XP:

%APPDATA% normally maps to: C:\Document and Settings\
<windows username>\Application Data
%APPDATA%\ICQ\Application.mdb

Application Settings

%APPDATA%\ICQ<icq nr>\messages.mdb

Message / Chat / SMS History

%APPDATA%\ICQ<icq nr>\owner.mdb

Contact List / Password Hashes

%APPDATA%\ICQ<icq nr>\<icqnr>.fb>

Cached contact list saved after connection to ICQ servers.

Vista:

%APPDATA% normally maps to: C:\Users\<<windows usernam>\AppData\Roaming
%APPDATA%\ICQ\Application.mdb

Application Settings

%APPDATA%\ICQ<icq nr>\messages.mdb

Message / Chat / SMS History

%APPDATA%\ICQ<icq nr>\owner.mdb

Contact List / Password Hashes

%APPDATA%\ICQ<icq nr>\<icqnr>.fb>

Cached contact list saved after connection to ICQ servers.

Received files through ICQ

XP:

C:\Documents and Settings\<<Windows username>\My Documents\ICQ<icqnr>\

ReceivedFiles\icqnr icqname

Vista:

C:\Users\<<Windows username>\Documents\ICQ<icqnr>\

ReceivedFiles\icqnr icqname

Note

The existence of this directory DOES NOT necessarily mean the user got files from that contact.

If the *View files* option from the ICQ context menu on a contact is used, this folder gets created.

The Messages table contains the complete message history of the user, including authorization requests and responses. What might be interesting from a forensic perspective is the fact, that every message that is sent or received is stored together with an auto-incrementing number in the autoNumber column, so missing numbers might be a hint for some kind of manipulation.

On Figure 7 (Content of a Messages table) you see a short message history.

The subject column saves the first 255 characters of the message, whereas the full length message (if its length exceeds 255 bytes, otherwise the same data as in the subject column is saved), including HTML formatting is stored again as binary data in the data column.

To be able to reconstruct a flow of communication it is important to know the direction of the messages. If the from column is empty, this means the message was sent to the user whose to id can be found in the to column. If a message was received the ICQ Nr. directly in the from field denotes the senders ICQ.

Deleted Messages

Within the ICQ Client itself, it is not possible to delete the message history of a user.

Messages that have been deleted from the messages.mdb manually

might still be there and can be found by opening the messages.mdb in some hex-editor and checking for messages around the ones visible in the MDB Viewer Plus.

On Figure 8 (Messages.mdb opened in hex editor with deleted message highlighted) you see such a deleted message, that is visible right after an untouched one. On the screenshot, the red circles as well as the first marked byte (01) show the original autoNumber the messages used to have.

As the .mdb format is not fully documented, an easier way to recover deleted entries is most probably possible at the time the documentation becomes available.

ICQ 7

As for ICQ 7 the filenames and paths did not change but the extensions where changes to .qdb and they are now SQLITE Ver 3 files.

You can easily read and extract data from them using for example SQLite3 Explorer[9].

To make it short, the password is still stored the same way as in ICQ 6.5.

There are two noteworthy changes in messages.qdb:

In table messages, Autonumber was changed to messageId but with the same function as before.

“to” column was changed to participantsHash but it’s functionality is the same as with ICQ 6.5.

“type” is now represented numerical “100” meaning “ Instant Message ” “1100” meaning “ Games ” “1400” meaning “ Chats ”

“subtype” was removed.

The table chatHistory was renamed to Participants and still contains only the ICQ Nr. of the user (as userId) and the participantsHash.

The Users table contains the same information as in 6.5.

The date column is no longer using ascii text but is now using variant time as the timestamp for the message, see [4] for conversion.

The changes in “Owner.qdb”

In table records there have been a few additions but all the information in there is stored as binary blobs, so you should export them to some text file, Base64Decode them and look at them in a Hex Editor. The forensic interesting addition is in the Key column, value 1 and Section value “\”, a timestamp that was added:

ImportContactsTime that is a variant timestamp.

Conclusion

A default installation that an investigator might encounter reveals a lot of information that can be forensically evaluated and integrated into case work. Even if some settings are modified, mainly the password or the history not saved, a lot of information can be found using basic forensic techniques like string searches or similar carving techniques used to recover other filesystem or file-format artifacts.

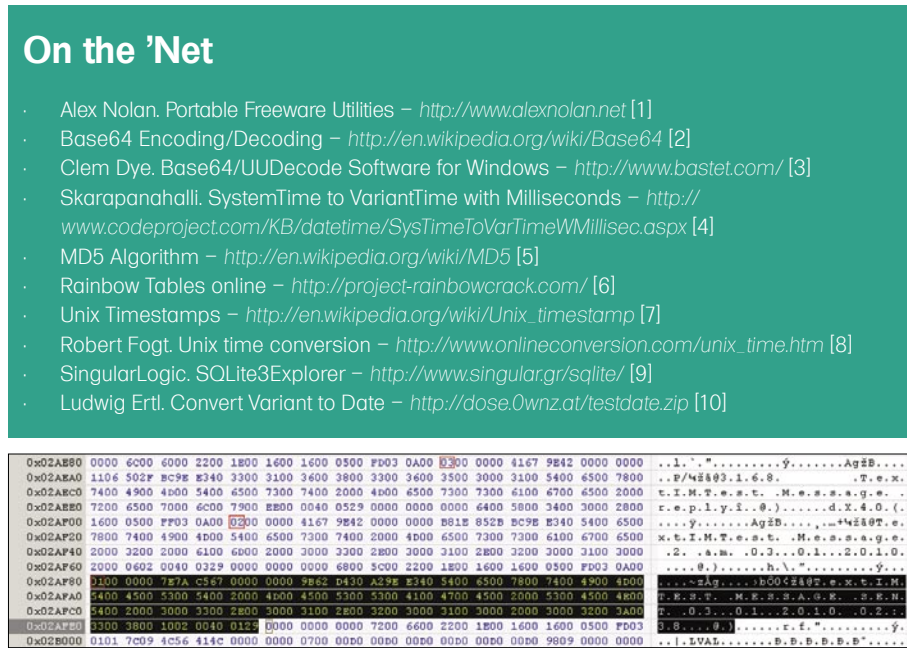


Figure 8. Messages.mdb opened in hex editor with deleted message highlighted

Florian Eichelberger
flo@dynamix.at



SPIN LEGENDS

www.tony-deslandes.mobi



RIC MESSIER

Cracking SIP

Difficulty



The Session Initiation Protocol has been gaining considerably more use in recent years.

Voice Over IP has demonstrated both cost-effectiveness and quality such that both consumers and businesses are using it to connect with friends and colleagues worldwide. SIP is a text-based protocol which makes it far easier to troubleshoot and understand than a similar protocol like H.323.

In part because of its ease of use, however, SIP has some vulnerabilities which may not be widely understood. One of these vulnerabilities is the ease of cracking passwords. There are several freely available tools that make this process trivial. While there are ways of mitigating these risks, even the mitigations have workarounds, exposing the core vulnerability again.

Setting Up SIP Infrastructure

While businesses are likely to use some of the large names in the telephony industry like Avaya or Cisco, home users can use many commercial offerings for VoIP phone service. Alternately, more advanced home users could set up their own PBX at home using something like Asterisk. Asterisk can be used to create a PBX which can make use of either SIP or traditional TDM/PSTN services.

Asterisk was used for the purposes of writing this article, in part because support and documentation for Asterisk is more widely available than other SIP proxies. The objective was to have a SIP proxy which could handle

registrations from an endpoint. Starting with a Ubuntu 9.10 installation, Asterisk is very easy to install.

```
apt-get install asterisk
```

Ubuntu will take care of installing Asterisk, complete configuration files and startup scripts. Asterisk has a large number of configuration options making it a completely customizable PBX but for our purposes here, there isn't much that needs to be done. By default, many SIP proxies (*User Agent Server or UAS*) and clients (*User Agent Client or UAC*) use UDP for their default protocol. This particular configuration goes back to the original definition of SIP where TCP wasn't a required transport protocol. As a result, UDP became the least common denominator.

UDP is a less-secure protocol than TCP in that there is less of a guarantee that the messages are coming from the location they claim to be. UDP is far easier to spoof than TCP is. In the file `/etc/asterisk/sip.conf`, we turn on TCP so that it can be used, making communications with the Asterisk server a little more secure than if UDP was used. TCP is turned on with the following line:

```
tcpenable=yes ; Enable
server for incoming TCP connections (default
is no)
```

WHAT YOU WILL LEARN

How the SIP REGISTER process works for authentication

How to crack SIP authentication

How to better protect your SIP infrastructure

WHAT YOU SHOULD KNOW

What is the Session Initiation Protocol

How to capture packets

Once TCP is turned on, we can see that Asterisk is listening on the SIP port, 5060, on both TCP and UDP protocols.

```
kilroy@milo:/etc/asterisk$ sudo
netstat -atunp | grep 5060
tcp        0      0 0.0.0.0:
5060      0.0.0.0:
*          LISTEN   1409/
          asterisk
udp        0      0 0.0.0.0:5060
0.0.0.0:*
          1409/asterisk
```

We need to add a user so that we can register using a client. This is done in two places. First, we add a block into sip.conf creating the user.

```
[kilroy]
type=friend
username=kilroy
secret=12345
host=dynamic
context=tutorial
```

Additionally, we map the user that we created to an extension. This is done primarily in the case of phones, which could also be used for this extension but soft clients are not only easier and far cheaper to come by but easier to demonstrated. The following block was put into extensions.conf:

```
[tutorial]
exten => 6001,1,Dial(SIP/kilroy)
```

Once the Asterisk server has been restarted to take the configuration changes, we are ready to register a client to the server and start cracking some passwords.

Configuring the Client

There are a large number of SIP clients available for common operating systems like Windows, Linux and Mac OS. Blink is a simple client that runs under Mac OS X. The following screen capture demonstrates how to configure the client to use the new server (see Figure 1 and Figure 2). Blink needs to be configured to REGISTER to the SIP proxy. Timers can also be set to increase the registration interval. In cases where a SIP client is

registering through a network address translation (NAT) device, the registration interval needs to be set lower so that the pinhole in the NAT/firewall can be kept open to allow incoming phone calls to find the SIP client. The following preferences screen shows where to turn on registering and setting the registration interval.

Capturing and Cracking SIP Traffic

The SIP REGISTER scenario uses digest authentication in the same way the Web

traffic does. In fact, SIP was designed with HTTP in mind and behavior, status codes and messages are similar. A SIP REGISTER message will come in to the proxy which responds with an appropriate 4xx message indicating that the user is not authenticated. Embedded in the response message from the proxy is a nonce which will be used to create a hash of the username, realm and password. The client responds with another REGISTER message, this time including the



Figure 1. Logging into your SIP account

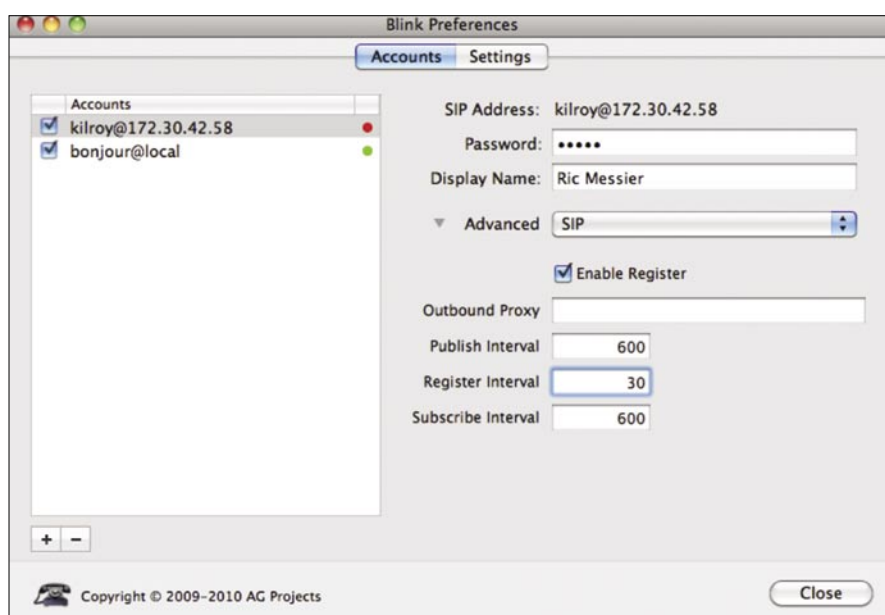


Figure 2. SIP Account settings for REGISTER

Listing 1. wordlist.cpp to create a wordlist

```
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char **argv)
{
    for (int i=0; i<99999999; i++)
    {
        cout << setfill('0') << setw(8) << i << endl;
    }

    return 0;
}
```

calculated authentication information. With the nonce provided, a brute force attack on the password is possible (see Figure 3).

This process is made significantly easier because often phones are used as SIP clients, particularly in business environments. Phones are limited in their abilities to generate passwords, being restricted to numbers. The sipcrack utility uses a wordlist to do the brute force by calculating the hash for every password in the wordlist and comparing it against the hash that was provided by the client to the server. The following code will generate all possible 8 digit passwords using only numbers (see Listing 1).

The sipcrack package comes with a sipdump utility which will take either a previously captured PCAP file or will capture traffic in real time by putting the

network interface into promiscuous mode and looking for SIP messages (see Listing 2).

Once we have a dump file, we can feed it into sipcrack to get the password. Even with very large dictionary files nearing or larger than one gigabyte of data, cracking the password is a very fast process. A password can be calculated in a matter of seconds. The wordlist in use here is slightly smaller than 1G.

```
kilroy@milo:~$ ls -la wordlist.txt
-rw-r--r-- 1 kilroy kilroy 899999997
    2010-02-14 13:09 wordlist.txt
```

The cracking process, however, is very fast. In this case, the password used was at the very bottom of the file. Even so, it took less than a minute to work through 10,000,000

possible passwords and calculate the hash, comparing it to the provided hash value.

While we captured the data on the system where the server is located, it doesn't have to be done there. Man in the middle utilities can certainly be used to redirect traffic to a third party system where the SIP REGISTER messages can be collected in order to do the password cracking. Utilities like ettercap or arpspoof can be used to collect the packets at a third party system.

Mitigation and Conclusions

Commercial offerings, particularly those that use phones, often use DHCP to redirect the client onto a separate voice VLAN. This is done for security purposes as well as performance. From a design perspective, it makes sense to have voice traffic separate from other types of data traffic. This allows for a better quality of service. While this makes it harder to get the REGISTER messages, it doesn't make it impossible. There are utilities which decipher the DHCP or Cisco Discovery Protocol (CDP) messages and automatically join the system to the voice VLAN. Once on the voice VLAN, everything here works the same as described.

Requiring complex passwords, which is a typical mitigation against password cracking, doesn't work when you are limited to a telephone keypad to put in passwords. Certainly, the keys on the

No..	Time	Source	Destination	Protocol	Info
1	0.000000	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
2	0.000173	172.30.42.57	172.30.42.56	SIP	Status: 401 Unauthorized (0 bindings)
3	0.000225	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
4	0.006988	172.30.42.57	172.30.42.56	SIP	Status: 200 OK (0 bindings)
5	14.555997	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
6	14.557049	172.30.42.57	172.30.42.56	SIP	Status: 401 Unauthorized (0 bindings)
7	14.557091	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
8	14.560051	172.30.42.57	172.30.42.56	SIP	Status: 200 OK (1 bindings)
9	24.156988	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
10	24.157124	172.30.42.57	172.30.42.56	SIP	Status: 401 Unauthorized (0 bindings)
11	24.157165	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57
12	24.236134	172.30.42.57	172.30.42.56	SIP	Status: 200 OK (1 bindings)
13	54.338991	172.30.42.56	172.30.42.57	SIP	Request: REGISTER sip:172.30.42.57


```
To: "Ric Messier" <sip:kilroy@172.30.42.57>\r\n
Contact: <sip:gzvchwaq@172.30.42.56:54969>\r\n
Call-ID: yb0bukbP5EvrV2JX4XThcMB-vtKR5-SN\r\n
CSeq: 4 REGISTER\r\n
Route: <sip:172.30.42.57;lr>\r\n
Expires: 0\r\n
User-Agent: blink-0.15.0\r\n
Authorization: Digest username="kilroy", realm="washere.com", nonce="22cc6d57", uri="sip:172.30.42.57", response="358dfbf4df53c99a9611004a-
Content-Length: 0\r\n
```

Figure 3. Wireshark capture of REGISTER packet with hashed response

Listing 2. Using sipdump to capture and decode SIP REGISTERs

```
kilroy@milo:~$ sudo sipdump -i eth0 sip.dump

SIPdump 0.2 ( MaJoMu | www.codito.de )
-----

* Using dev 'eth0' for sniffing
* Starting to sniff with packet filter 'tcp or udp'

^C

kilroy@milo:~$ sudo sipdump -p sip.pcap sip.dump

SIPdump 0.2 ( MaJoMu | www.codito.de )
-----

* Using pcap file 'sip.pcap' for sniffing
* Starting to sniff with packet filter 'tcp or udp'

* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
* Dumped login from 172.30.42.57 -> 172.30.42.56 (User: 'kilroy')
```

telephone can be used to enter letters but it's not common to do that. Not all phones have displays that are capable of indicating which selection (which letter or number, whether uppercase or lower case) has been made. Because of that, most systems that interface with phones as the user interface device make use of solely numbers to take authentication information. As a result, the best that can be done when restricted to numerical input is requiring longer password lengths but there is a limit to that as well without running into the problem of it being too long for users to remember and resorting to writing it down to know what it is.

As noted above, using switches doesn't help protect against capturing traffic not destined for the place where it is captured. Switches can be attacked as well using either a flood or by simply using gratuitous ARPs to redirect the messages at layer 2. ARP spoofing can

be a very effective attack. These attacks require physical access to the network, however. Other mechanisms can be used (DNS spoofing or spoiling) to get traffic redirected at layer 3.

The one mechanism that has so far resisted attack is requiring that TLS be used to encrypt the SIP messages. If TLS is used, the TLS messages need to be cracked before the REGISTER messages can be attacked in order to locate the password that was used. The best way to easily secure VoIP infrastructure is to ensure that SIPS and SRTP are used to communicate between endpoints. This will at a minimum make it very difficult to attack the messages to acquire critical data.

Ric Messier

Ric began in the IT industry and developed a lifelong interest in security in the early 80s. He worked for a global Tier 1 ISP for a half dozen years and spent several years doing security assessments for a global leader in telecommunications solutions.

Join

Hakin9 team!



If you would like to help our team in creating Hakin9 magazine you can join our authors or betatesters today!

All you need to do, is to send an email to:

editors@hakin9.org

and give us a brief description of your field of interest.

We look forward to hearing from you!



MATT DAVIS

Faith in the Format: Unintentional Data Hiding in PDFs

Difficulty



Adobe's Portable Document Format (PDF) has gained a prominent foothold as a method of distributing text-based and graphic-based information. Its use has become ubiquitous across academic, technical, and governmental institutions and has become one of the forerunners of information dissemination. The success of this format can be thanked, in part, to the designers and their creation of a platform-independent method for rendering text and graphics.

Such independence allows users to feel assured that their data is viewed similarly to the original document, despite the platform the reader is consuming the information upon. However, that trust is derived from the faith in the format through the tool used to read the data, even if there is more attributes hiding under the covers. When a document is created, modified, and re-saved, Adobe's standard allows for previous versions to be retained, in essence creating a running history of the document. Such revisions, while not seen in the distributed copy, still reside in that document. The basis of this article is focused on the concept that document maintainers might not be aware of this property, and consequently distribute modified versions containing non-public information. With such a high usage of this format, it would not be a surprise that there are numerous public documents containing non-public data that is merely non-visible but still accessible. As will be explained, the retaining of a document's history in the distributed copy is not unique to PDF, but is common across other document formats as well.

encapsulation, to prevent other coders from using objects improperly and/or compromising the overall integrity of the system in development. Likewise, strategies used behind the covers of a file format do not all have to be understood to be effective. For instance, understanding the Huffman-Coding or frame structure of an MP3 file seems rather irreverent for a casual listener. What does matter is the effectiveness of the format. Does the format do what is expected by the end-user? It is at this point one might question their trust in the format.

In the case of PDF files (*Portable Document Format*), it seems a rational judgement to assume that a document encoded to PDF specifications will render nearly identical on all intended recipient's machines. After all, *P* in this case stands for *Portable*. The rendering of a PDF is truly the responsibility of both the application that encoded the document and the application reading the document. That trust in creating versus reading is why there is a standard surrounding PDFs [1]. It is this trust, or faith of document formats, that concerns this article.

Introduction

The success of a file format can be attributed to the frequency of use. Such frequency is typically a function of convenient features that makeup the format. However, some of these features are not always apparent to the end-user, and some might argue that a bit of sheltered ignorance is not a bad thing. APIs use techniques of information hiding,

Portability Features Underlying PDF

It might be arguably so that the success of the PDF format is solely based on its portability. No matter what platform a user might be running, chances are that there is a PDF document reading utility available. So what makes this portability a reality? This section discusses some of the features that enable the PDF format to become both an efficient and portable way to distribute information.

WHAT YOU WILL LEARN

File formats are not always what they seem

Some document formats retain previous version history

WHAT YOU SHOULD KNOW

Documents contain metadata

Published documents might have more information behind the covers

Embedded Fonts

The innumerable amounts of fonts available might first make one queasy. After all, how is the reader supposed to view a document when they can only, at best, have a subset of all available fonts the world over? The PDF format addresses this concern by allowing for the original document's font to be embedded into the resulting PDF document, being that the font is one of the numerous supported formats (e.g. TrueType, Type 1, OpenType). This portable-friendly feature allows a PDF creator to use nearly any font of their choice, no matter how obscure, and removes worry regarding how the resulting document will look on the recipients' machines. This feature requires no intervention by either the developer or reader while also removing any kind of dependence on the machine that is responsible for reading the document. Such functionality does not come without drawbacks, mainly size. Adding fonts to a PDF increases the overall size of the document. However, Adobe allows for data compression of various objects inside a document as a *stream*. Therefore, space can be reduced by having the font embedded in a compressed fashion. Consider a document composed of numerous fonts, and the amount of processing that a typical PDF save might require. Assume that each fresh save of the document will have to bundle in these fonts so that the resulting PDF is portable. The aforementioned concern for efficiency, during file-saves, plays an important role for the *revision history* discussed later [1].

Images and Other Media

The ability to incorporate graphics and images alongside text, is a testament to the universality of this format. By graphics, I elude to the primitive constructs the standard defines, which allows for the drawing of lines and text. Images, on the other hand, can be *content streams* similar to font data. It is important to note that by viewing the images of a PDF document, one places a certain amount of faith that the PDF viewing application renders the data to the likes of the original.

The PDF standard also increases the portability of the format by allowing other types of media to be embedded inside the

document. Media along the likes of movies, slideshows, interactive forms, and even 3D-objects are allowed. Details of such are in the standard, however it should be noted that all media is the responsibility of the viewing application, whether that viewer leverages other libraries or tools to render the document is at the rendering application's discretion. In effect, there is a lineage of trust that exists between the viewing application and utilities harnessed in rendering any part of the PDF. This lineage further abstracts the end-user from what is really going on, and even increases the potential for security exploits. Sharing processing duties amongst other libraries and tools is nothing unique to PDF, and has become ubiquitous in the information age, as such compatibility and reuse aids in arriving at the end result in the most efficient and convenient manner. After all, there is no need to re-invent the wheel.

Compressed Data

As one can imagine, a document containing multiple fonts, images, pages, and media can grow quite unwieldy. To ameliorate what might otherwise be a large document in size, objects composing the PDF can be compressed. Even a series of objects, also known as an *object stream*, can be compressed. With the ability to compress data, document size was probably one of the original concerns when the format was originally considered.

Revision History

While maintaining a document that might be rather large in size, one might also question the efficiency of saving data. Especially if certain portions of the document might undergo an encoding process. When a document undergoes a modification, chances are it is not the entire PDF that needs reworking. For instance, maybe a short and simple text edit to correct a misspelling was all that was needed. In such a case, and especially for large documents, re-encoding the entire document, images and all, could produce a lot of redundancy and wasted cycles. To reduce such redundancy, the PDF specifications allow for the document to be appended to. In other words, only the objects that undergo a change need to be re-created. The object's original and

unmodified brethren still exist, however the PDF viewer need not reference that older object's versions.

Appending changes to the document, instead of recoding the whole work all over again, produces a running history of what was changed. This information is then passed on when the document is distributed. These changes are not limited to modified objects, but also incorporate removed objects. It is my personal feeling that this is a rather *unknown* attribute of the PDF format, and is easily justified by the logic that it creates efficient document re-saves and provides a history of changes for undoing and comparing. The PDF 1.7 specification states:

In addition, because the original contents of the document are still present in the file, it is possible to undo saved changes by deleting one or more addenda. The ability to recover the exact contents of an original document is critical when digital signatures have been applied and subsequently need to be verified.

Appending modified data, instead of overwriting, eases document maintenance. Each object in the document is referenced by an offset of where it appears in the file. All of the object offsets are easily located in a cross-reference table that is stored at the very end of the document. As the document becomes modified, only an updated cross-reference table needs to trail the document. Consider what would happen if an object were to be modified in place, instead of appended. The result would be that the offsets of all subsequent objects would have to be recalculated and the cross-reference table updated. Likewise, more space might be needed, and would result in a new file with appropriate memory allocated to rewrite the changes and all subsequent objects. Appending, versus in-line object updating, also avoids misalignment of the cross-reference table, as no object offsets have to be recalculated. Therefore, appending objects is the easier and more efficient approach to document updating.

There are cases when the unmodified information is not desired. Consider removing information from a document that might have been classified knowledge, and then making the document available in the public domain. It is possible to save a modified document

without producing the extraneous information that might have sensitive data. However, that role lies at the mercy of the PDF encoding application and its user.

PDFs in the Wild

Being that PDFs retain modified and removed data from previous versions, I decided to conduct a very brief study of PDFs that are available in the public domain. I was not terribly sure what I would find, however, I was hoping to find documents that contained multiple revisions.

Tool: pdfresurrect

To aid my efforts in examining the PDFs available in the public domain, I decided to write a small application (This tool can be downloaded from: <http://www.757labs.com>) that produces all versions of the PDF in question. To accomplish this task, the tool merely looks for the PDF end of file tag (`%%EOF`) that gets created after each document revision. The tool then gathers all of the cross-referenced objects, which are later used to produce a summary of the analyzed document. The cross-reference table (xref) contains a list of document objects and their offsets into the document. One can think of this as a table of contents. However, this table is produced just before the EOF marker in the document, as knowing where the offsets are located would be easiest to figure out once all of the objects have been written. Instead of building a fresh PDF for each version of the document that exists in the PDF file, the pdfresurrect tool takes advantage of the file format and appends to the end of the file the cross-reference table that was created for that version. That table, as mentioned earlier, contains the object offsets for the version in question. Therefore, your PDF viewer, if looking at the last cross-reference table, should be able to read the pdfresurrect output documents for the specific versions of the PDF it was to extract the documents from. While the PDF standard contains a nicer explanation, the following is sufficient to grasp the overall concept of how a PDF is laid-out:

```
PDF FILE { DATA OBJECTS - CROSS  
REFERENCE TABLE - EOF }
```

In the case of a basic update to the document, the result would look more like:

```
PDF FILE DATA OBJECTS - CROSS  
REFERENCE TABLE - EOF - UDATED DATA  
OBJECTS - UPDATED CROSS REFERENCE TABLE  
- EOF }
```

Once pdfresurrect has generated the various document versions, a summary file is produced. This summary uses the cross-reference tables for each version, starting with the least recent, and proceeding to the final version, in order to generate a list of objects and provide a object-modification status: (A)dded, (M)odified, or (D)eleted. The object type, if found, is also reported. Since a PDF consists of multiple object types, changes to documents might only be aesthetic (e.g. font, pagination) and not the text changes that some might find more informational.

It must be mentioned that this tool works for simple cases, the object counts are not always accurate, and this tool does not account for all objects or object types in the output summaries. The main purpose of this tool is to extract the versions of the document and not to analyze them. As of the PDF specification version 1.5, the cross-reference table can be encoded into a stream. The pdfresurrect tool is low dependency (aside from some directory creation business). One goal was to make pdfresurrect portable, thus no compression libraries (external libraries) are required. However, by avoiding compression, certain PDFs of version 1.5 or higher might not be able to be processed by pdfresurrect. PDFs of 1.5 or greater have the ability to incorporate compressed cross-reference tables as streams. This is not always true, but if the tool reports '0' versions, then the latter was probably the case.

Searching for the Animal

As mentioned, one of the desires that led to pdfresurrect incarnation was to take a peak at PDFs in the public domain. Reason being, the curiosity to see if any, otherwise hidden versions, contained *interesting* content. Such content that might not be desired public. I decided to comb the web for a few sets of PDF documents, mainly 50 PDF documents per five top level domains (.com, .edu, .gov, .mil, .org). This data gathering resulted in a total of 250 PDF documents collected for the purpose of examining revision history. To remove any personal bias, the documents were obtained as the first 50 results pulled

from a Google query for the said filetype and domain. In the case of a dead link, the next document from the result set was pulled.

To obtain the documents, a simple Google query was conducted using the search string of similar style: `filetype:.pdf domain:.xxx` where domain is one of the five aforementioned. The query results were saved as a .html file from which some quick parsing removed everything but the URL to the animals. A simple script calling `wget` for each entry in the parsed results file then pulled down the documents.

After the sample data was obtained, the pdfresurrect tool was then used to obtain history information from all of the documents. For the purposes of this study, the main goal was to obtain a ratio of non-modified to modified documents (i.e. those with stored previous revisions). Such data should give a rough idea as to how much history is being distributed, possibly unknowingly, in the public domain.

Results and Mitigation

The following list shows the findings resulting from the obtained documents:

- .com: 0 of 50 contained obtainable document revisions (2 unprocessed because of compressed stream xref tables)
- .edu: 2 of 50 contained obtainable document revisions
- .gov: 0 of 50 contained obtainable document revisions
- .mil: 4 of 50 contained obtainable document revisions
- .org: 2 of 50 contained obtainable document revisions (2 others unprocessed because of compressed stream xref tables)

The results produced ratios that were less than expected, and are merely showing that the more popular of results from Google have little retained history information. These findings do not elude to personal documents distributed amongst cohorts, or businesses disseminating documents amongst their employees. Of concern might be classified documents that have varying levels of classification, that were not picked up on the rather small data gathering.

Such a case would be if a document was originally classified as *top secret* and then being modified to produce a *secret* version. Obviously, the institution scrubbing the *top secret* data would not intend to have that information distributed to those authorized at the lesser *secret* level. Some of the information from obtainable histories did result in a few unique findings. None-the-less, this brief exploration has proven that multiple revisions do exist in the public domain, and it is expected that a more thorough investigation should generate greater findings, especially when the search has a targeted focus, such as business intranets. Fixing such a potential problem is rather easy. One simple mitigation that can prevent the distribution of past revisions, is to merely cut and paste the data as a new document and then save that version. The resulting document would be generated as if it were created from scratch, and never have the chance to acquire history information. Another mitigation would be to use the aforementioned *pdfresurrect* tool, which has an option to write a character over data for the original document objects that have been modified.

Other File Formats

While this paper is primarily concerned with information hiding resulting from the maintaining of revision history in a PDF, it only seems appropriate to explore other formats of the like. Two common formats, the open OpenDocument and Microsoft's Word, are used in similar fashion as the PDF and will be briefly explored in this section. These formats do retain document revision information. This paper is not concerned with the utility that can be harnessed from retaining revision data. Of concern is the idea that not all document maintainers are aware that such information is being saved, and thus might reveal information that should otherwise be squelched.

OpenDocument Format (ODF)

The OpenDocument format is an open standard format maintained by the OASIS group. This format encompasses a wide gamut of document types common, but not necessarily limited to, office applications (e.g. word processing, presentations, charts, spreadsheets, drawings) [2]. This format is incredibly flexible, in part to the fact that it is defined as an XML schema. Not only does the format define the various elements viewed when creating the document (e.g. text, paragraphs, images) but there is quite the amount of metadata that can underlie the document. Such information contains creation dates, user modifications, etc. To further flexibility, the metadata can be custom or user created [3].

Upon initial glance at this standard, the document has the ability to manage quite a bit of metadata about itself. It only goes to follow that the method of maintaining revision history is well-defined and incredibly helpful when trying to determine what changes were made. First, the ability to *track* document changes can be toggled on or off. Unlike PDF, the authoring tool can enable or disable this historytracking feature. The functionality of the PDF is based heavily on file offsets of the objects. If a PDF object were to be removed completely, the cross-reference table would not match the objects in the document, and consequently become misaligned. Another rather helpful feature of the ODF's history tracking is that it tells what text data was changed, by whom, and when. It is important to note here that the document changes that have been mentioned regard text documents, and not graphics. The specification states that changes of table data in spreadsheets are tracked but not in common text documents. None-the-less, the potential for encapsulating sensitive information from previous revisions can be retained in such documents.

Microsoft Word '97-07 Binary (.doc)

Microsoft's Office, a suite of office applications, has been an incredibly popular toolset for nearly any domain (home, academia, work, etc). Many institutions require the use of such an application set, even though there are comparable alternatives available that can read and write this format. The format of brief discussion here covers the Word '97 Binary (.doc) format. As expected, features of Word are aided by retaining metadata and past revision information within the document innards. Such features as document merging, annotations, and revision tracking, seek benefit from this retained information. Files of this format contain a flag that can be set allowing recipients of the document to have their changes untracked. Unlike a PDF document, Word's binary format has its main information header at the beginning of the document. The Word Binary specification defines two modes of saving a document. The first being a traditional *full-save* and the other being a *fast-save*. The fast-save, according to the specification, appears to retain previous document information and merely appends the updated or added information to the end of the document. This latter method is faster than the former as the entire document does not have to be re-saved, merely appended to [4].

Conclusion

As one can see, it is easy to place faith in a format and assume it does just what it looks like it should be doing. As we have seen, what you see is not always what you get, and the study shows that a small search of the public domain can turn-over documents that have hidden information. However, quantifying the sensitivity of someone else's revision information is beyond the scope of this study, and making assumptions at that level would only tarnish this paper. What is of concern is not the useful feature of document revision history tracking that Adobe, Microsoft and the OASIS OpenDocument Format leverage, rather the concern is directed towards the document maintainer who might have unintentionally retained sensitive material in a publicly published document.

Matt Davis

Matt Davis aka enferex, has played a software engineer by day, and hacker of software when the lights go down. He has been part of the 757Labs nerd-posse in the Hampton-Roads Virginia area since 2003 when he helped to start a local geek-gathering.

On the 'Net

- Adobe Systems Incorporated. PDF Reference Sixth Edition: Adobe Portable Document Format. Adobe Systems Incorporated. Version 1.7. November 2006. [1]
- opendocument.xml.org ODF Wiki Knowledgebase: OpenDocument Overview. OASIS. <http://opendocument.xml.org/overview> February 2008. [2]
- OASIS. OpenDocument Format for Office Applications (OpenDocument) v1.1 Oasis Standard. OASIS. February 2007. [3]
- Microsoft. Microsoft Office Word 97-2007 Binary File Format (.doc) Specification. Microsoft Corporation. 2007. [4]

APT, Google, China and YOU!

MATTHEW JONKMAN

Advanced Persistent Threats, or APT, are a problem. A big problem, and for everyone. We've all known and assumed that for years the usual juicy targets like defense contractors, government agencies, law enforcement, and similar were being hit and occasionally compromised by foreign governments and vigilante hackers alike. We're in a lot deeper than that now, and it's about time more companies woke up to the new reality. What's the new reality? Here are a few points:

- EVERY company has been compromised at one time or another, and likely has at least one compromise right now.
- EVERYONE is a target. Every Company, Every organization, Every computer. EVERYONE!
- Conventional Defenses (Antivirus, IDS, firewalling) are NOT adequate.

As a guy that's been on the ground cleaning up a number of these types of incidents and deploying IDS systems for large and small organizations let me reinforce number one above. Every company has compromises if they have computers. The NSA has them, the FBI, the State Dept, DoD, and the equivalent organizations in every other country in the world have them. If you think your security stance, technology, and response team are better equipped and funded than the NSA you need a massive clue-bat beating. It doesn't matter how good you are or how locked down your network may be. If you're running the same OS and browser that 90% of the rest of the world runs and you have users you are not safe.

Companies of all sizes and industries are targeted. Every company has bots inside now

and then that just do the usual spread and spam, annoy the security team, maybe get an employee's banking credentials and cost them some heartache. This is a problem of course, but not what we classify as an APT. To make it into this elite category of APT the threats needs to satisfy a few criteria:

- It is targeted at a specific individual, organization, or industry
- It is NOT there to directly make money
- It is run by a human, not one bot in a swarm of thousands
- The malware is generally customized to the attacker/target
- The attacker will sit quietly and monitor for weeks, months, even years

All of these traits make an APT very difficult to detect and counter. They're generally spread by social engineering. A classic will be for an attacker to research the target and find a partner company. They'll identify an accountant or executive in the target company and send them a fake invoice from the known partner company that contains an Adobe PDF exploit. It's very likely the target will open that invoice as they trust the source and are likely expecting an invoice. Or the attacker may send a link to a page hosting IE exploits or just put those in an email assuming the target is likely to at least look. Bang, that fast the bad guy is in, and they haven't even put much thought into it yet.

There are even reports of executives and employees of target companies being physically targeted while abroad. Laptops being imaged or compromised while left in a hotel room, laptops being specifically targeted for mugging style thefts, or communications monitored while overseas and credentials compromised. If you think

this is stuff just reserved for government to government spying, think again. It's a function that's easily outsourced and not that expensive. It can be accomplished by very unsophisticated individuals at the request of an individual that can use the data. Consider the return for a businessman that wants new research and lives in an economy where it's not a problem to commercialize intellectual property that may be stolen. (cough*China*cough). A few thousand dollars invested for a possible disclosure of research from a foreign company is a very worthwhile investment!

Everyone is a target. This is not exaggeration. You are a target. There are many types of APT predators, and not all choose their target up front. Some are looking to learn all they can about an industry and just shop around for information they could sell to others, some are looking at a particular region because they know they have a buyer for any related intelligence, and some are just throwing out a net to see what they come across.

Those just throwing the net are most likely to be looking for information they can either use to blackmail the target company, or that they could sell to that company's rivals. I've seen both happen. Once you find yourself the victim here you're screwed. There's no winning. I've had client companies that have paid out thinking they'd be safe, but the attackers only returned six months later wanting more money. I've seen rivals be offered secret data, and even if they diverted that request to law enforcement and the target paid a ransom the data is still reused, offered to others, or publicly disclosed.

Once data has exfiltrated that an attacker realizes is valuable the target company is going to lose. They can pay and maybe the

disclosure will be less prominent, but the company is at the mercy of the bad guys until the data they possess is no longer valuable. If you really believe an attacker is going to be honorable and not continue to shake you down because you *had a deal* you're living in fantasy land. It's a lot easier to shake down the guy you already have on the hook than to go find new victims.

The most serious threat is the attacker that is not in this for personal gain. They use the same tactics to get into a company. They research their targets well and often use social engineering to gain entry. Their goal is to understand a company or individual and monitor all communication and data to which they are exposed. This attacker will not make a lot of noise on the network looking to spread to other hosts. They'll sit and watch, waiting for an administrator to log into the computer for maintenance and grab those credentials, then use those to push quietly to other target individuals.

But what makes this attacker most dangerous is they're not working for themselves anymore. They're on salary now, and in many cases in a uniform, and they are given very specific targets to compromise, and very specific data to retrieve. Targets are more often commercial and research data than government to government activity.

Lets consider the recent Google vs China debacle. This is a very clear example of government sponsored attacks. There are a number of things to learn from this:

- If Google can be that deeply compromised, so can you
- Among other things, the targets were human rights activists communications
- The compromise was discovered mostly by chance
- this is just one that was made public, many more happen in secret every day

If this were a financial shakedown job the target would likely have been to get data from as many gmail accounts as possible. But this was not the case, the only personal targets Google reported were prominent chinese human-rights activists. Who would go to this length to study their communications besides the chinese intelligence services? The Russian Business Network doesn't care about

Chinese human rights unless they've been paid to care. Most Western powers are friendly to those activists and less likely to target them. And likely there would be a legitimate legal process or intelligence relationship for Western countries to gain access to these communications directly through Google if they so desired.

The malware in use is almost always custom, as it was in the Google hack. Antivirus won't help you. AV companies rely on capturing samples of new strains, finding a common trait and writing signatures to detect those. When the only copy of a piece of malware is running on your there's not much chance your email gateway or workstation antivirus is going to identify it. In the Emerging Threats.net sandnet we analyze tens of thousands of malware samples per day. In general, samples that are not of an established strain of malware constitute at least 30% of new samples each day and are not detected by AV for at least 72 hours after capture. At least 10% of those never are identified specifically. And even of the 70% that are known strains a large number of those aren't even detected until 48 hours or more after collection.

Some statistics on APT malware released by Mandiant

Average File Size: 121.85 KB

- Most Common APT Filenames/Process Names:
 - svchost.exe (most common)
 - iexplore.exe
 - iprinp.dll
 - wiinzf32.dll
- APT Malware avoids anomaly detection through:
 - Outbound HTTP connections
 - Process injection
 - Service persistence
- APT Malware Communication:
 - 100% of APT backdoors made only outbound connections
 - 83% used TCP port 80 or 443
 - 17% used another port

This should scare you. Your IT guy CANNOT sit down to your executive staff computers and look at the list of processes and determine if something looks funny as was possible in the old days. These

backdoors likely won't show at all, but if they do they'll be indistinguishable through normal analysis. Normally we could detect malware through it's command and control connections. But we've seen an increasing trend of malware that works very hard to make all of it's communication look very much like normal http traffic. This makes using Intrusion Detection Systems to find the infection more difficult. Mandiant mentions that 100% of the APTs it's analyzed of late have all used normal http ports and normal http traffic to communicate. And considering that most every one is a custom compile for that target it's safe to assume that the http channel it uses is custom for that target as well.

So how do we defend ourselves? My primary point in this article has been to exemplify that our traditional security stance and tools are woefully inadequate when you are using the majority OS and browser that contain known insecurities. But even if those tools were more secure, when a user is involved social engineering will work in a certain percentage of attacks. It'll never be 0% of your users fell for it. So even without vulnerabilities you are still vulnerable via the humans.

Defense is possible. We need to focus on identifying unusual communications, strengthening our perimeters, and having fast and competent incident response. It certainly won't hurt to consider trying one of the myriad non-Microsoft operating systems. That won't make you 100% secure, but it sure reduces the odds of being a victim.

I am an IDS guy, and I do believe IDS can evolve and help us with this problem. At the OISF (<http://www.openinfosecfoundation.org>) we are building a next generation IDS called Suricata. We are incorporating new ideas to help identify APT and other malware threats. Anomaly detection, Live IP Reputation, and a number of other tools that combined will enhance our awareness of exfiltrating data. We at Emerging Threats.net are working to keep IDS signatures current to detect these command and control channels.

While we'll never be 100% safe, we are in a very bad state now. We are not just a little vulnerable, we're wide open. We must change!!!

As always please send me your thoughts, jonkman@emergingthreats.net.

ID fraud expert says...

The Evil Twins – Identity Fraud and Phishing

JULIAN EVANS

Identity fraud has been around for decades, but only in recent times with the advent of the Internet and media exposure has it started to become a global issue.

Phishing on the other hand has been around for a much shorter time period – less than 15 years. Phishing and identity fraud are *Evil Twins*, working hand in hand and in some instances hand in glove. The 'Evil Twins' is very much here to stay.

Identity Fraud and Identity theft – one half of the Evil Twin

Most people do not understand that there is a difference between identity fraud and identity theft. Identity theft is the misuse of an identity (such as your name, date of birth, current or previous addresses), without your knowledge or consent. These details are used to obtain goods and services in your name.

Identity fraud is the use of a misappropriated identity in criminal activity, to obtain goods or services by deception. This usually involves the use of stolen or forged identity documents such as a passport, utility bill(s), *Social Security Number* (SSN), birth certificate and/or driving licence. See Figure 1 for US statistics on fraud methods.

More than 11 million US adult consumers became victims of identity fraud in 2009, up from nearly 10 million in 2008. The number of fraud victims

rose for the second year in a row. On the other hand, victims' out-of-pocket costs and the time required to resolve fraud have decreased. Out-of-pocket costs can include unreimbursed losses, lost wages due to time taken off work, and possible legal fees for those victims attempting to prosecute. (Source: Javelin Survey 2009)

US banks have stepped up their efforts in counteracting fraud and minimizing the cost and inconvenience suffered by consumers. Most victims don't experience any out-of-pocket costs, but those who did suffered an average cost of \$373. The average time to resolve the fraud for these victims was 21 hours. Due to the zero-liability fraud protection offered by most banks and credit card companies, most victims will only have to pay out-of-pocket expenses to cover their time in resolving fraud, not for reimbursing fraudulent charges. (Source: Javelin Survey 2009)

Types of Identity Theft/Fraud

Over the past few years identity theft/fraud has evolved. The list below shows some of the most recent fraud types which encompass both the cyber world and the real world. The fraud trend suggests identity theft/fraud will continue rising in the cyber world but will continue to cost

banks and insurers as well in the real world:

- *Card fraud abroad is on the rise* – this is where your cloned card is used in a country where they do not use chip and pin – beats any chip and pin system,
- *Keyloggers* – malware – this is the biggest threat to online security since the first virus (called Elk Cloner) was discovered in 1982 – have a keylogger on your machine that logs every keystroke and you have a major problem and worse still you might not even know you have one.
- *Scareware* – antivirus xxxx (the xxxx stands for the year) and antivirus360 are just two versions. This method uses search engines to advertise (including a sponsored link) a fake security product often with a similar name to a leading brand – i.e. antivirus360 which is similar to Norton 360 – in fact scareware is just that – software that scares you into downloading it and then informing you that you have malware on your computer. It will also inform you to remove the viruses and purchase a licence key. The problem is, it doesn't remove the viruses – it

continues to infect your computer!
 · *Flash exploits* – you click on a banner advertisement which unknown to you downloads a malicious program to your computer. The program could download a keylogger, virus or stop your computer from working altogether.

Figure 1 highlights the most common identity fraud/theft types. New identity fraud methods are appearing all the time.

A typical identity fraud scheme

One of the most recent fraud types was how a card fraud can impact your daily life. A husband with his wife and son were just about to pay for lunch when their debit card was declined. The husband arrived home and logged onto his online bank account to find he was overdrawn by \$400.

Over a period of 24 hours 40 transactions were made by his debit card. Worse was to follow – 2 days later there were two packages on his doorstep and on the following day he got a further seven packages.

A close inspection found that all the packages were from different companies, ranging from health and beauty products to diet pills to coffee. In total it added up to \$335, mostly in shipping fees.

There isn't proof that the individual (or individuals) actually gained financially from this scheme. The fraudster doesn't really care whether they get the package; all they want is the credit for sending somebody to actually sign up.

In the US, the *Better Business Bureau* (BBB) indicates they haven't seen this type of scheme before. What is very scary is that all the fraudster needs is a person's name, address and card number to pull this off this fraud type.

Identity theft/fraud solutions

Identity theft/fraud is reasonably simple to identify and stop. The credit reference agencies provide credit monitoring services which provide real time protection of an individual's credit profile. In some countries i.e. UK individuals can apply for Protective Registration which validates with a variety of proof

of identity documents that the individual is who they say they are. This happens before any new credit is granted but the downside is ALL credit applications take considerably longer to be processed.

There has been a rise in particular in the US of companies claiming they not only protect your credit but your online profile (non-credit related information). Online monitoring companies in the US are using fraud monitoring technologies which check thousands of non-credit related databases to detect and stop any incidences of identity theft.

In addition these companies are able to provide real time alert facilities (mainly in the US) which notify you for example on whether there is any payday loan activity associated with your name, date of birth and or social security number. These types of loans are short term with high interest and can be obtained from most lenders without a credit enquiry. Fraudsters will often focus on taking out these types of loans by writing post-dated fraudulent checks and using stolen identity documents.

A number of US companies also offer additional services which might include searching thousands of known criminal (deep web searching) for illegal selling

or trading of an individual's personal information. These searches would look for SSN, credit card numbers and CC dumping).If anything is found these companies goes about charging you to take the necessary steps to resolve the problem.

The monitoring services are an invaluable protection (albeit expensive options) for individuals concerned with having their identity stolen in the real and cyber world but should be used alongside existing online security such as anti-virus; specification and anomaly based behaviour detection; firewall; anti-spyware; anti-keylogger; encryption modules (including a sandbox) and password managers.

The Evolution of Phishing

The first recorded use of the term *phishing* was actually made as far back as 1996. The term has evolved from the variant of *phishing* and can be described as *baiting users to steal financial information and passwords*. The very first phishing attack was on AOL whereby a fraudster would pose as an AOL staff member and send an instant message to a potential victim asking them to reveal their personal password. The fraudster

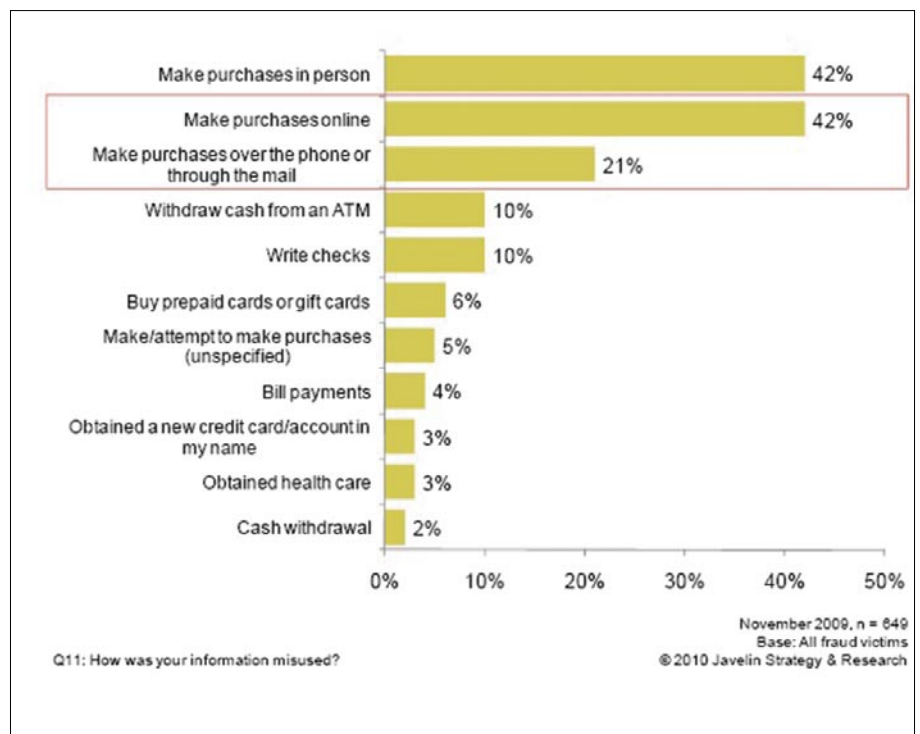


Figure 1. What Are the Most Common Methods of Fraud?

ID fraud expert says...

could then access the victims email account to perpetrate criminal acts such as verifying account data and sending out spam data to the victims address book.

Attack Vector One – Email

The actual phishing method i.e. sending a user a fake link hasn't evolved much over the past few years. What has evolved though is the attack vector. In the early days of phishing cybercriminals would send emails pertaining to be from a reputable company or family friend – the user wouldn't hesitate to click on the link and be directed to a fake website.

The victim would be none the wiser concerning the legitimacy of the website they had just visited. More often than not the website would collect a username, password and other information data such as a passcode (for online banking) and credit card details.

The email attack vector has evolved by using the social web i.e. Facebook, Twitter

and MySpace to attract new victims. The Web 2.0 world is also fast becoming a primary focus for fraudsters especially considering the number of potential users/victims and obvious financial opportunities.

Attack Vector Two – IM

Email isn't the only method – *Instant Messaging* (IM) is also a popular attack vector using Windows Live Messenger and Skype (and the popularity of Web 2.0) to attract new victims. Everyone today wants lots of friends in the cyberworld – so it's no surprise to see unsuspecting people being lured into accepting friend connections.

Some of the tricks deployed by hackers include hacking a hotmail or Windows Live account using keyloggers and Trojan droppers. This would allow them to send out for example a malicious URL redirector to the individuals address book. Most recipients will trust the email address and some

will no doubt click on the malicious URL – so everyone catches a virus – so the infection continues with other email accounts sending out to their address books.

Another method includes receiving an IM with an embedded malicious Flash or image file link. Click on the link and malware is installed on the target machine. The malware can pose as an Adobe Flash Player update but in reality it's designed to log a user's browsing history, in particular their Google search queries within Firefox. This information is then uploaded to a hacker-controlled server.

Phishing – the human problem

A common technique for spammers and phishers is to use the *self-sending* email method. This is where the *from* line as well as the *to* line appear to have your e-mail address. This type of attack method is commonly used with webmail services

Table 2. Most common types of Identity Theft/Fraud in the real world

Fraud Type	Fraud Action	Fraud Result
Application Fraud (CNP)	Stolen or false documents (like utility bills and bank statements) are used to open an account in an individual's name. Alternatively, the fraudster may use counterfeit documentation for identification purposes.	Fraudster will use your credit/debit card (quoting the number found on the bill) over the Internet, by telephone, fax and mail order. This is called Card-Not-Present (CNP) fraud.
Account ID take-over	A fraudster obtains your key personal information, and is able to take over the running of your financial account.	The fraudster will pretend to be you in an attempt to deceive your bank or credit card company to arrange payments to be taken from your bank account.
Mail non-receipt card fraud	The credit/debit card is stolen in-transit. Normally occurs when a card is renewed/or lost/stolen.	As above but also the fraudster may also instruct the bank to change various details of the account, such as the address, and then ask for new cards and cheque books to be issued. Properties with commercial letterboxes like flats and student residence halls are at particular risk of this type of fraud.
Social Security Number (SSN) identity theft	Someone steals your SSN and obtains employment in your name.	Fraudster can damage individual's credit rating. Also employer reports wages earned to the IRS under your SSN leaving you to pay income taxes on these earnings. Further, an identity thief's use of your SSN can cause you to lose life sustaining benefits.
Medical identity theft	Someone steals your identity and either obtains medical insurance in your name or uses your current medical insurance policy to obtain treatment or prescriptions.	You can be denied health coverage or lose your current health coverage because of false information placed in your medical record.
Credit card identity theft	This can occur when your credit card is cloned (or dumpster diving) by copying your card information with a special device.	The fraudster uses your credit information to obtain credit in your good name. Victims don't always find out they have been a victim until it is too late.

such as Yahoo, Hotmail and AOL. This method is another form of email spoofing (see section titled 'Email Spoofing' for further information).

Social Engineering

Phishing is an example of a social engineering technique used to fool users, and exploits the poor usability of current web security technologies. Most phishing occurs because an individual has either inadvertently clicked on a URL in an email or IM conversation. There is also the small part that manipulation plays. This is similar to a confidence trick which attempts to deceive the target into disclosing personal sensitive confidential information to commit a fraudulent act.

There is no solution to social engineering. Most of the Internet Security vendors don't consider education and awareness important and feel that users will click on what they want when they want regardless of whether it is trusted. There are signs that education and awareness will become important. There are only so many security modules you can add and most people don't really understand what the current crop of modules i.e. firewall, spamfilter, browser filters etc do for them.

There is also another underlying issue with consumer security. Updates

– Those people who have anti-virus and a firewall rarely update. Various figures have been talked about – with some security researchers suggesting the figure is nearer sixty per cent. Consider social engineering and you can see why Phishing is here to stay and probably will be with us for as long as the Internet is with us.

The Phishing threat to Business

Successful phishing attacks can cause serious damage to a business. While preventing phishing attacks entirely isn't possible, companies can take measures to ensure attackers' efforts are unsuccessful.

Over the past few months (Q4 2009) we have seen several large companies such as Google and Microsoft fall victim to phishing attacks. The single biggest weakness with the enterprise world is the 'user'. The user might be a company employee or consumer – it actually doesn't matter which because both are interrelated.

Mitigating the Phishing Risk at Administrator level

There are a number of phishing risks for administrators – which include, cross site scripting (XSS), HTML referrer header spoofing and email spoofing.

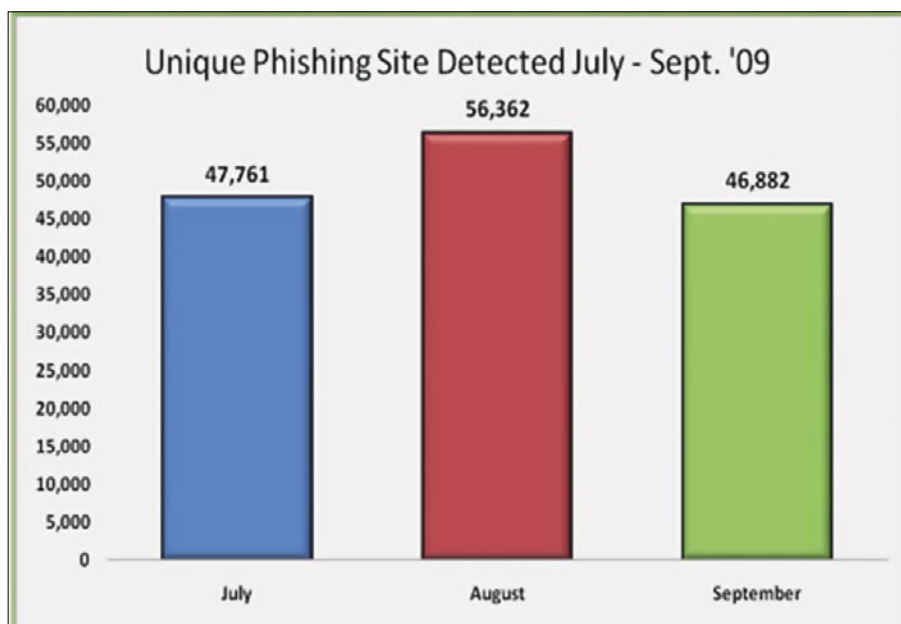
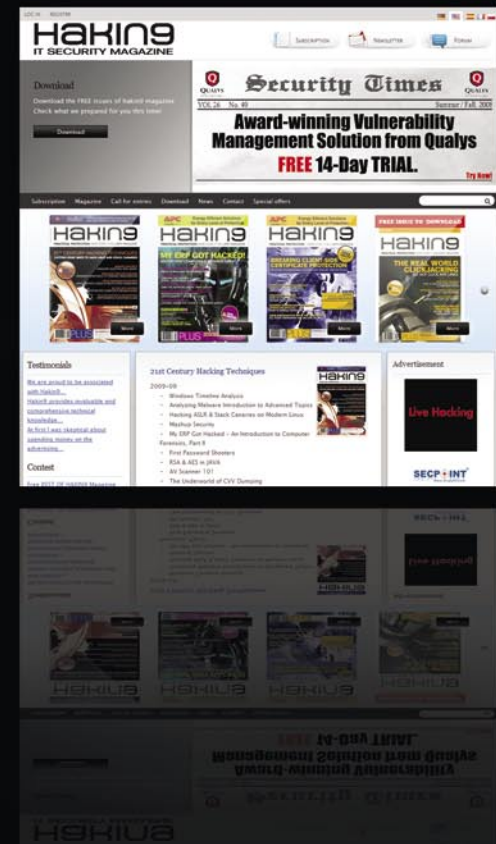


Figure 2. The number of unique phishing websites detected by the APWG during the third quarter of 2009 reached a new record in August with 56,362 (Source: APWG)

Subscribe
to our
newsletter



and get :

- free issues of Hakin9
- recent information on new release
- free articles

and more

WWW.HAKIN9.ORG/EN

ID fraud expert says...

Cross Site Scripting

It's critical to review the website for cross site scripting (XSS) vulnerabilities in the network as phishers often target unvalidated forms and URL redirectors which take unsuspecting victims to fake web pages.

Most readers will know about the XSS tracing vulnerability exploit which exploits Java, ActiveX, Flash and many other controls that allow the use of the HTTP TRACE request. The cross site scripting (XSS) hack provides a hacker the opportunity to exploit cookies and authentication credential information on a victim PC.

A simple step in stopping this type of attack would be to disable the PUT, DELETE, CONNECT and TRACE methods on the Web server. If the methods are needed then it isn't too difficult to setup limited access to trusted users.

iFrame attack

Another major problem for enterprise has to be iFrame attacks. Phishers can use the iFrame to create website overlays (in other words create a separate page which looks like the original) which then makes it very difficult for users to know whether the web page/site is genuine or behaving as it should.

It's difficult enough to know a web page is real let alone trust the web page

isn't behaving differently to the way it was programmed. To make sure that the browser window snaps out of any iFrame attack, administrators need to employ the TARGET_top directive.

Also worth mentioning – so often these days, websites are using pop ups to gather data i.e. asking you what you thought about the page you just visited but sadly attackers are also exploiting the pop-ups for phishing attack.

HTTP Referrer header spoofing

Another worthy security tip should be to implement a simple HTTP referrer header check to see where any suspecting request actually came from. The biggest headache for administrators is that the referrer header is very easily spoofed, but it's possible to shut the door on spam emails as this will identify and close the URLs in any spam emails – in effect this will deny a rogue page/website from using a referrer redirector to force a user's browser from sending fake referrer headers.

Email Spoofing

Another growing problem for businesses is email spoofing and the lack of authentication. Every email will have a *Sender Policy Framework* (SPF) record which is configured in the DNS

to validate every SMTP server request. The SPF agents will reject emails not sent from the servers listed in the SPF records with some email programs i.e. Outlook also will be able to flag them as *spam*. Worth remembering also is that SPF is that it is not universally supported so it might prove difficult to deploy SPF on email servers in your business environment.

Spoofing email isn't necessary spam but it is something that can damage a person's authority and also the business brand/identity. In some cases emails can purport to be from a genuine email address but actually they were not sent by the owner at all.

The damage is obvious here. Mass spoof emailing isn't going to go away and may become more prevalent, especially as the 'genuine' email address will not be picked up as spam because it assumes the sender actually sent it.

Administrators will no doubt be aware that they will also need to maintain and review configuration of all account activities.

The Future

The *Evil Twins* will evolve over the coming years to encompass the growing smart phone generation. Cyber criminals have already started to realize the target potential of sending premium rate (and malicious) URLs to mobile phone users using SMS.

As more and more of the population use smart phones, they will also use Web 2.0, email and pay for bills and check bank balances on the move using their mobiles – so too will the cybercriminals who will look to exploit the variety of mobile operating platforms and application security holes to steal sensitive personal or business information.

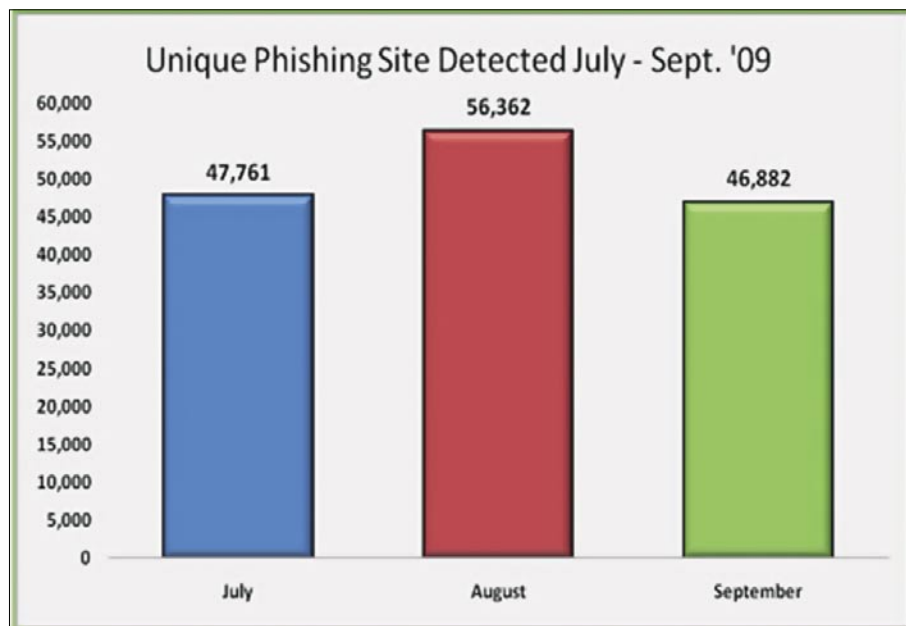


Figure 3. The number of unique phishing reports submitted to APWG in the third quarter of 2009 reached an all-time high of 40,621 (Source: APWG)

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>



100% PURE HACKER

Heorot.net

Heorot.net provides training for penetration testers of all skill levels. Developer of the De-ICE.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 Pen-Test 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



VINTEGRIS S.L

VINTEGRIS S.L is a company dedicated to IT security in Spain. We focus on development of authentications, web access control, password management and synchronization, and digital signature systems, to integrate into the IT of our customers. We also perform integration of third-party recognized security products. Most of our consultants are CISA and CISSP certified and our company is ISO/27001 certified.

<http://www.vintegris.com>
e-mail: info@vintegris.com



Netsecuris

Netsecuris is a professional provider of managed information security and consulting services that focuses on ensuring the security of your networks and systems. Services include managed firewall/intrusion prevention, managed email security, network penetration testing, vulnerability assessments, and information systems risk assessments.

<http://www.netsecuris.com>
email: sales@netsecuris.com



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/>

JOIN OUR EXCLUSIVE CLUB AND GET:

- **Hakin9 one year subscription**
- **classified ad for duration of your subscription**
- **discount on advertising**

You wish to have an ad here?
Join our EXCLUSIVE&PRO CLUB!

For more info e-mail us at en@hakin9.org or go to www.hakin9.org/en

EXCLUSIVE&PRO CLUB

SPECIAL REPORT

Black Hat 2010

JAMES BROAD



From the moment you walked in to the venue it was evident that this was Black Hat. All the signs were there the iconic fedora clad silhouette against a dramatic red background, the vendors with boxes of swag and a schedule that included some of the most interesting topics and best quality training available. But while the frigid February winds howled through the busy streets of Arlington, Virginia outside a different Black Hat crowd was completing their registration. Trench coats, common at both Black Hat and Defcon, covered the business suits and corporate casual of government employees and security contractors.

Gregory Schaffer, Assistant Secretary, Office of Cyber Security and Communications keynote address kicked off the conference on Tuesday morning. Mr. Schaffer explained that the voice of security professionals has not been heard by government and corporate leadership, but we as security professionals need to keep pushing the security agenda. This theme was echoed throughout the five conference tracks over two days; Application Security, The Big Picture, Hardware, Forensics and Privacy, and Metasploit.

The Application Security track focused on programming errors and flaws in logic that facilitate exploitation as well as trends on secure coding that prevents exploitation laden code. While this tracks focus was on exploiting poorly written code, its underlying message was to

those programmers that are striving to write secure code for future applications. From the *Connection String Parameter Pollution Attacks* presented by Chema Alonso and Jose Palazon on the morning of day one to the presentation *Interpreter Exploitation* by Dionysus Blazakis that closed out the conference on day two this track detailed the real threats that exist in current software and the trends that need to be taken to begin to correct this trend.

The Big Picture focused on sweeping trends in technology that impact worldwide computer usage and communications. In their presentation *Whose Internet is it, Anyway?* Andrew Fried, Ben Butler and Richard Cox, new threats to a connected world were exposed. This presentation was followed by Nicholas Percoco defining the state of global security in his *Global Security Report 2010*. These two presentations opened many eyes and minds in the DC area, and surely caused an update in many organizations security plans.

Joe Grand explained how recent hardware vulnerabilities can be exploited on otherwise secure systems in *Hardware is the New Software*. He explains how the commonly overlooked hardware components are a seldom used vector into today's systems. If you were looking for exposure to physical security topics Deviant Ollam discussed how many technically hardened systems are protected by inferior locks and how to tell the difference between these locks and hardened locks in *The Four Types of Lock*.

Mac computers were the topic of focus for Matthieu in *Advanced Mac OS X Physical Memory Analysis* as part of the Forensics and Privacy track. Tom Cross explained how lawfully implemented wiretaps on network connections could be exploited to provide access to these communications by unauthorized individuals or groups in *Exploiting Lawful Intercept to Wiretap the Internet*

The final track, Metasploit, proved to be quite popular with many people listening to Mike Kershaw explaining how he uses the Metasploit Framework to exploit wireless clients before they connect to corporate virtual private networks in *Wireless isn't dead; Attacking clients with MSF*. HD Moore closed out this track with his presentation, *Metasploit and Money*. Focusing on the Rapid 7 acquisition of the Metasploit Framework, HD explained the future of this framework in the commercial space.

While the DC venue has a higher corporate and government ratio than its Las Vegas cousin, this is a conference that should be on every security professionals calendar. However if you missed the 2010 event much of the content is available on the Black Hat DC 2010 media archives page at <http://www.blackhat.com/html/bh-dc-10/bh-dc-10-archives.html>. Offering white papers, audio and video recordings, source materials and presentation slides, this is a site that offers the next best thing to attending the conference in person, sorry no swag from the web page.

Security at the Mobile World Congress

TAM HANNA

The mobile market has provided yours truly with quite a bit of hakin9 revenue over the years – as already stated multiple times, the combination of inexperienced programmers, unaware users and large amounts of vulnerable (and easy-to-find) devices carries quite a bit of explosive potential. It thus should not surprise anyone that this year's Mobile World Congress had quite a bit of security-related stuff – but what was shown in specific?

The Mobile World Congress can be considered the leading exhibition in mobile. Organized by the GSMA, it is held once a year in Barcelona and attracts up to 50,000 individuals.

Food for the dumb

In a nutshell, all exhibitors showed some kind of *downward trend*. It started with Microsoft's extremely consumer-centric Windows Mobile 7, and went on with various products aimed squarely at thrillseekers – those of us who read technical magazines were largely left in the rain.

Mobile Software

Both F-Secure and Kaspersky were at the MWC – and both of them unveiled similar solutions. Given that nobody really cares about mobile viruses anymore, their security suites moved on to areas like SMS / call spam, remote wiping, theft prevention or recovery/locating of stolen devices.

Theoretically, all stolen mobile phones could be located quickly. That is, if it weren't for carrier disinterest and governmental protection of the privacy of the phone thief...

As usual, smaller vendors did the pioneering work here (anyone remember

BluePill from PalmPowerUps – it more-less pioneered remote wiping for consumer smartphones). Psiloc's latest idea is an application called CryptoLine. When installed onto two Symbian smartphones, it builds an encrypted channel between the two devices – their owners can then perform VoIP calls or send encrypted SMS's to one another.

More anti-theft

As most European countries are at least two years away from an election, theft mitigation becomes a lucrative business. Sagem Orga's *SecuritySuite* does just that: it protects various types of data commonly stored onto USIM cards (SMS, contacts) by backing them up on an operator's server.

Networks get armed

Network Operators also got a new weapon in their arsenal: a company called SandVine showed off their new Policy Traffic Switch. Carriers who integrate a PTS into their network then gain various benefits ranging from bandwidth management, usage statistics to very useful security features.

For example, the system uses a patented algorithm to find and stop spam-sending zombie computers: this is insanely important, as it protects network operators from getting their IP ranges blacklisted.

Other notable features include protection from Denial of Service attacks and, more importantly, port scan protection. This is insanely important for operators with many iPhones: if attackers cannot perform port scans to find vulnerable boxes, worm and virus outbreaks get slowed down.

Chip-level security

Cryptography Research goes one level deeper. They are a fab-less *chip IP vendor*

focussed on the sale and licensing of patents which protect chips from various attacks.

They introduced a new processor which protects the keys used for symmetric algorithms from hardware-level attacks: attackers who desolder the chip in order to access the key in it will now have a more difficult life. Furthermore, new technologies against *Differential Power Analysis* were unveiled. DPA is a technique which aims to find out more about a given encryption system by studying its power intake (which, on almost all CPU's, changes depending on what instruction is being executed).

Countermeasures involve things like adding an additional, random power drain to the chip or modifying the algorithms to be less vulnerable. As the US Government is rumoured to add DPA protection to its equipment purchasing guidelines, this is an area which could get interesting very soon.

News from the Forensics front

Finally, law enforcement and large auditing firms were given a new tool from a company called CelleBrite. It allows data extraction on a physical level from most mobile phones by feeding them a custom firmware and reading out the content of the Flash.

Responsible developers should feel forced to react – read our series on encryption to find out more about making your application forensic-proof...

Conclusion

If this year's Mobile World Congress has proved something, then it's this: money can be made with security solutions for the mobile market. As users get dumber, risks will increase – and so will the amount of money carriers and device manufacturers are willing to spend...

Shmoocon 2010 Round-Up

CHRIS RILEY

Well, my first Shmoocon has come and gone and I can't think of a better way to kick-off the 2010 conference season.



Everybody always said that Shmoocon was a great place to socialize, meet new people and enjoy some great technical content. Well they certainly didn't lead me astray. Despite the snow storm raging across Washington DC, everybody was in high spirits, and the presentations didn't fail to deliver.

Information disclosure via P2P networks: Why stealing an Identity via Gnutella is like clubbing baby seals

Larry Pesce, Mick Douglas

I had the pleasure of seeing a part of this presentation at last years Defcon conference (Larry talked on a few of the more critical points in the FAIL panel). The real addition to this version of the talk was the release of the Cactus Project, a tool for interacting with the Gnutella based P2P network. This tool will come in really handy for organisations and penetration testers who want to check for information disclosure through the P2P sharing tool. Currently the toll is written in Perl, however plans are already in discussion about a possible Python/Ruby version. The information that Larry and Mick discovered ranges from the humorous, to the down-right scary. If you've not already checked your P2P sharing settings, this talk will give you a reason to move to do just that..

Windows File Pseudonyms

Dan Crowley

In this interesting presentation, Dan Crowley from Core Security Technologies talked about how Windows file path and naming normalization can be used to cause unexpected and interesting behavior. Although the attack surface is limited with some of the examples he gave, the premise was certainly interesting and enough to demonstrate that extra care should be taken when implementing filters, access control and IDS/IPS alerting. Especially interesting was the use of Windows 8.3 filenames in bypassing filters (WAF/IDS/ISP) and speeding up brute-force file discovery. The demos he gave showed some good examples of there use in bypassing file upload filters, as well as Local File Inclusion. HE also touched on the use of DOS device references, such as PRN, CON, LPTx, etc... Which is something I don't think many people have considered before. This is certainly something people should be building into their automated and manual testing plans, I know I will be.

Learning by Breaking: A New Project for Insecure Web Applications

Doug Wilson

Doug discussed the release of a new project (*OWASP Broken Web Applications Project*) designed to bring together a number of Web Application testing platforms into a single VMware image. The project includes a number of previously separate learning tools, including DVWA (Damn Vulnerable Web App), OWASP's WebGoat, Mutillidae, as well as some older versions of Wordpress, phpBB and Yazd. Beyond the obvious benefit of bringing these applications together into a single VM, the goal of the project is to provide detailed information and assistance on the vulnerabilities and possible exploit methods. The project is currently looking for assistance, so I'd suggest heading over to the project website.

Guest Stealing... The VMware Way

Justin Morehouse, Tony Flick

Justin and Tony put together a new twist on a previously disclosed Tomcat directory traversal vulnerability. By accessing the VMware management console it was found that the version of Tomcat embedded into VMware server, workstation and ESX(i) was still vulnerable to the attack. Aside from the obvious attack vectors (such as access to the `/etc/passwd` file), it is also possible to navigate to the VMware storage and directly download copies of the running guest systems. Certainly an interesting attack vector. Justin and Tony released a PoC Perl script to automate the downloading of VM guests (`gueststealer.pl`) and an NMAP script has also been released by the people behind `Skullsecurity.org` as well. Full information on the vulnerability is available in the VMware advisory (CVE-2009-3733) VMSA-2009-0015.

Social Zombies II: Your friends need more brains

Tom Eston, Kevin Johnson, Robin Wood

After their excellent Defcon talk, I was interested to see where the sequel was headed. As expected, the dangers of social networking played a big part in the presentation. With more and more people joining up to social networks (Twitter has 6.2 million new members each month), it's become a prime hunting ground for attackers and spammers alike. Whether its Facebook, Twitter, or any other social network, the ploys are almost always the same. Fake account with a sexy picture == danger. Add to this the use of shortened links and people who blindly click anything that seems even mildly interesting, and I'm sure where you can see this headed. To drive that point home, Jessica Biel was named at the *Most Dangerous Celebrity* on the web (August 25th 2009). Kevin discussed his (numerous) Facebook profiles, and his new *KanyeWesify* application (*I'ma let you finish...*). Two newer social network tools that are helping bridge the gap between your on-line and real-life activities, Blippy and FourSquare, were also discussed. However I think this quote from Chris Nickerson says it all. *I joined BLIPPY and all I got was jacked at the ATM.*

GSM: SRSLY?

Chris Paget

Unlike the 26C3 presentation or the same name, this focused a little more on the release of the OpenbootTS LiveCD (a Debian-based LiveCD combining OpenBTS, Asterisk and all other software needed to perform as a GSM base-station and route calls through a VoIP provider). As well as demoing a live IMSI catcher on stage and decoding of calls audio (using a simple Wireshark capture and decoding the forwarded SIP traffic), Chris also touched on the issues of A5/0 (NULL cipher), A5/1, A5/2 (water-down export-grade A5/1) and A5/3 (used for 3G). Technical information on MITM attacks against on GSM were also given (including the use of the 3 digit identification codes used to designate which network is which). In it's simplest terms, the handset will connect to the strongest signal. Even though the attacker may not know the handsets secret code (used for encryption purposes), the attacker can advertise a non-encrypted network (downgrade to NULL cipher). Most handsets will accept this without warning, and don't authenticate the provider. Even though the facility exists to alert the user of a NULL cipher (as with SSL errors/warnings), most handset manufacturers have deactivate this!

SPECIAL REPORT

Exposed! More: Attacking the Extended Web

Nathan Hamiel

Much attention has been paid to Web 2.0 and the increased attack footprint that it brings. This presentation looked closer at the interconnection between the distributed components of many Web 2.0 applications, and where they lack sufficient protections. The vulnerabilities that Nathan discussed were not however your typical exploits and inevitable Proof of Concepts to go with them. Instead he concentrated on the possible abuses of publicly available APIs. With this more generalized view, it is easier to see where these type of vulnerabilities could arise, and why they are so rarely considered security issues. After all, a majority of the examples are simply an attacker twisting the purpose of an API function to do something that the developers never considered. Alongside the theoretical attack possibilities, a number of examples were given (using del.icio.us, and hi5modules.com) to show the true effect of mis-using APIs. To finish things off, a new version of Monkeyfist (PoC Dynamic CSRF Tool) was released with improved functionality.

The friendly Traitor: Our software wants to kill us

Kevin Johnson, Mike Poor, Justin Searle

To finish up the presentations for day 3 (at least for me), the great guys at InGuardians presented some ongoing research into client-side exploitation using extensible features. By looking at the functionality of things like Flash, Java and ActiveX, it was possible to see that client-side security protections are tuned specifically to detect exploitation attempts. However by using the existing features in a malicious fashion, it's possible to exploit clients even with these protections in place. A prime example of this is the use of Flash to defeat the same origin policy. Flash uses the crossdomain.xml file to prevent Flash from communicating across domains. However, in a scan of the top 1 million sites (as listed on Alexa) it was found that over 15,000 sites permitted cross domain Flash access from any other website. This exposes the sites to a number of attack vectors that could by design be prevented. Alongside this, an as yet unpatched issue with the administrative console of the MiFi wireless device was also discussed. The vulnerability can open up the device to access from the Internet. By using a number of flaws together, it is possible to access the device configuration file and use the information stored within to access the device fully. Finishing up, Kevin talked about his favorite rising technology, HTML5. Browsers with support for direct SQL Database, File and Device access, Web Storage, Web Sockets,... With a feature-set like this, who needs enemies!

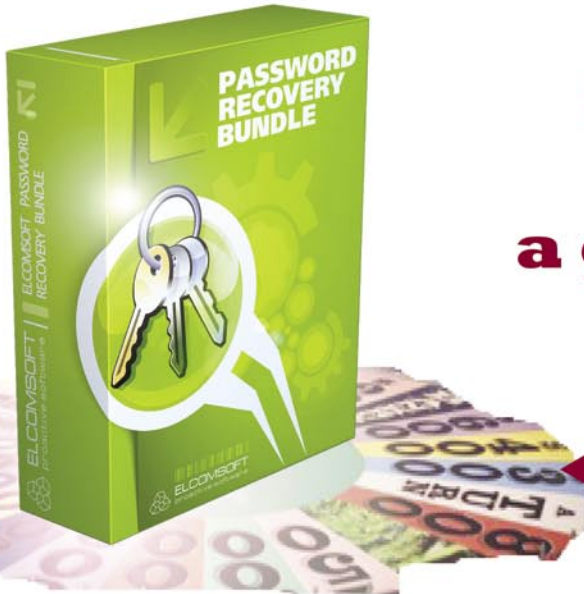
Alongside the great talks that were scheduled, the firetalks (shorter evening talks) really brought forward some great content. The talk by Michael Scheerer on Shodan for penetration testers was particularly interesting, as well as Dave Kennedy's talk on the *Social Engineering Toolkit* (v.04) and Benny Ketelslegers' talk on Sleep Hacking.

The people behind Shmoocon have already released a number of the slides from the conference, so I encourage you to take a look at the presentations page (<http://www.shmoocon.org/presentations.html>) on the Shmoocon website to see the full slides.

My memory of my first ever Shmoocon however, will always be the 15 minute Shmooball fight that developed from the podcasters meet-up. Getting so many podcast groups together (PaulDotCom, Security Justice, Securabit, SMB Minute, Exotic liability, Eurotrash Security, Social Engineering podcast, ...) for a single recording is hard enough at the best of times. Add some ammunition, albeit soft(ish) stressballs, to the mix and it quickly turned from a security Q&A into all out war. Certainly great fun, but I doubt the audio will see the light of day. At least... I hope not!

Further Information

- Shmoocon – <http://www.shmoocon.org/>
- Cactus Project – <http://www.pauldotcom.com/cactusproject.html>
- OWASP BWA – <http://bit.ly/bDywlC>
- Gueststealer – <http://www.fyrmassociates.com/tools/gueststealer-v1.pl>
- Social Zombies II (Slideshare) – <http://bit.ly/ab9i8l>
- OpenBootTS – <http://sf.net/projects/openbootts/>
- MonkeyFist – <http://hexsec.com/misc/monkeyfist>
- Firetalk (videos) – <http://bit.ly/clKHg4>
- Shodan – <http://shodan.surtri.com>



password administration is not a game of chances

17 per cent of users forget their password
once a month, 8 per cent once a week

Password Recovery Bundle is a complete suite of ElcomSoft password recovery tools allows corporate and government customers to unprotect disks and systems and decrypt files and documents protected with popular applications. Based on in-house tests as well as feedback from ElcomSoft valuable customers, these password recovery tools are the fastest on the market, the easiest to use and the least expensive.

- **Hardware-accelerated brute-force** attack based on NVIDIA CUDA; multi-CPU and multi-GPU support.

- The **password cache** automatically stores all discovered passwords in order to unlock other documents protected with the same password momentarily.

- **Dictionary attack** can quickly recover the majority of passwords used by general computer users, and up to 40 per cent of passwords employed in corporate environments.

- Supports **over 100 file formats**, including MS Office, Adobe PDF, Windows logon passwords, ODF, PGP disks, UNIX/Oracle user passwords, WPA/WPA2, Intuit Quicken, and much more.

«When auditing my client's networks and applications for weak passwords, I require a tool set that is dependable and fast. From time to time, I'll also receive a request to recover a lost password protecting a critical document or spreadsheet. Elcomsoft has delivered the desired results each and every time! I want to thank Elcomsoft for providing the best password auditing and recovery tools on the market.»

Kevin Mitnick



77 per cent of users use the same
password to protect various types of data

<http://elcomsoft.com/eprb.html>

Your questions are welcome at sales@elcomsoft.com

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 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:



that was easy.

PC Connection



Enter to **Win a Back-UPS ES 750G!** (A \$99 value)

Also, enter the key code to view other special offers and discounts.

Visit www.apc.com/promo Key Code n519w or Call 888-289-APCC x8253 or Fax 401-788-2797

"The price tag on the new UPS is \$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 five times more energy efficient than any other solution. By saving you \$40 per year in energy costs, our Back-UPS ES pays for itself in two 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



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



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!



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

APC
Legendary Reliability®