

HAKING

PRACTICAL PROTECTION

IT SECURITY MAGAZINE

WHEN I'M X64: BOOTKIT THREAT EVOLUTION

PERFORMING A HISTORY INJECTION
AGAINST THE CHROME WEB BROWSER
CHAPTCHAS, WHAT THEY ARE
AND HOW TO USE THEM
STRIPING SSL ENCRYPTION

Vol.7 No.02
Issue 02/2012(50) ISSN: 1733-7186

PLUS

INTERVIEW WITH RICHARD JOHNSON
(IL)LEGAL: SMART METERS, DUMB REGULATORS



It's here! Penetration testing for Students



**Click here
To enter the
early bird list**

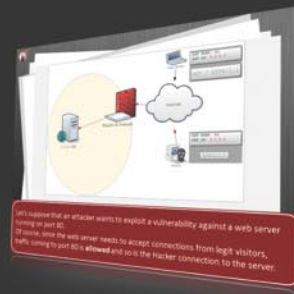


80% of beginners remain beginners or give up completely

We know the pain of being a beginner. You either don't have the foundational skills or you don't have a clear path to follow. Don't give up. There is a better way. Our course will teach you basics of networks and web apps.

It's not just about 1337 instructors

Expert teachers hardly remember what took them to the expert status. It's a fact. There is no way to effectively teach beginners other than help them building strong foundations and showing them the correct path.



You can do it

If you keep studying without a clear learning path you are probably wasting time. Secret is path and perseverance. Better a single step in the correct direction than 10 random steps. Our course will save you months of struggling and frustrations.

You gotta see this.

www.elearnsecurity.com



Still hacking virtual machines?



Coliseum Lab is here!

The most epic web app hacking lab
you have ever seen

CLICK HERE

14 educational challenges
in a multi-platform
environment.

Epic!

www.coliseumlab.com



HAKIN9 team

Editor in Chief: Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Managing Editor: Marta Jabłońska
marta.jablonska@hakin9.org

Editorial Advisory Board: Julian Evans, Aby Rao, Aleksandr Matrosov, Eugene Rodionov, Federico „Glamis” Filacchione, Satish Bommisetty, Praful Agarwal, Sulabh Jain, Christopher M. Frenz, Hamidreza Moheballi

DTP: Ireneusz Pogroszewski
Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Proofreaders: Bob Folden, Nick Malecki

Top Betatesters: Nick Baronian, John Webb, Ivan Burke

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 Marciniaik


CEO: Ewa Dudzic
ewa.dudzic@software.com.pl

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

Publisher: Software Press Sp. z o.o. SK
02-682 Warszawa, ul. Bokerska 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

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.

Dear all,

We are happy to announce that you are reading 50th number of Hakin9 Magazine!

We want to thank all the authors, beta testers, proofreaders who were helping us to prepare every issue. Big thank you, also for you, dear readers. Hakin9 would not be the same without your comments and good advice.

First article „When I'm x64: Bootkit Threat Evolution in 2011". At the end of each year it's traditional in security to offer a retrospective view of security-related events in the past 12 months and predictions of likely trends in the threat/anti-threat landscape for the upcoming year. You will learn how major bootkit families have evolved, their differences and resemblances and how attacks against 64-bit operating systems have become increasingly effective. If you are interested in digital forensics and want to learn about iPhone Forensics in detail read an article „iPhone Forensics On iOS 5". Imagine a computer which is protected with OS level password – we can still access the hard disk data by booting a live CD or by removing the hard disk and connecting it to another machine. So it is not easy to take out the chips and dump data in it. HTTPs is not an unknown terminology. Hyper Text Transfer Protocol Secure is a secure version of the Hyper Text Transfer Protocol (HTTP) which is a combination of HTTP with SSL (Secure Socket Layer)/TLS (Transport Layer Security). SSL comes preinstalled in BackTrack, the version which we used for this article in BackTrack Version 4. Read more in „Stripping SSL Encryption". We all know Captchas and any of us have used them. Have you ever wondered if they are useful, secure, accessible? Federico Filacchione will try to debunk some myth related to them and help you understand how you can use Captchas on your web application. How to view the data stored in Chrome's History database or inject falsified entries into Chrome's History database? You will learn from „Performing a History Injection Against the Chrome Web Browser". The chrome history files are actually simple to view and interact with in that they are SQLite databases and can be readily viewed with programs such as the SQLite Database Browser.

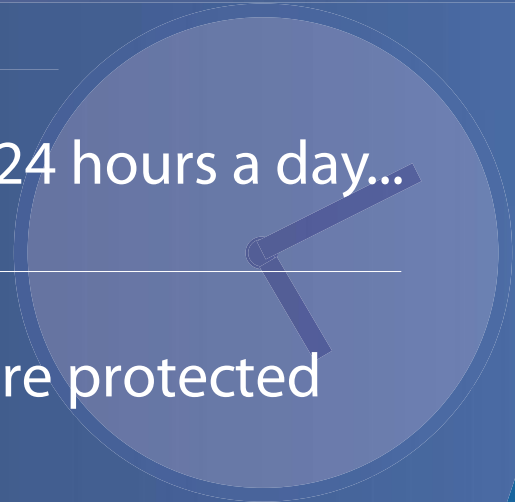
Last but not least – Il(legal) column. This time smart metering – one of the hottest topics in the Energy and Utilities sector in Europe and North America. We also recommend an interview with Richard Johnson. He is a computer security specialist, who spends his time playing in the realm of software vulnerability analysis. Richard currently fills the role of principal research engineer on Sourcefire's Vulnerability Research Team, offering 10 years of expertise in the software security industry.

Once again we would like to thank all of you, our long term contributors, authors and everyone from Hakin9 Team. We are looking forward for next 50 issues!

All the best,
Marta & Hakin9 Team

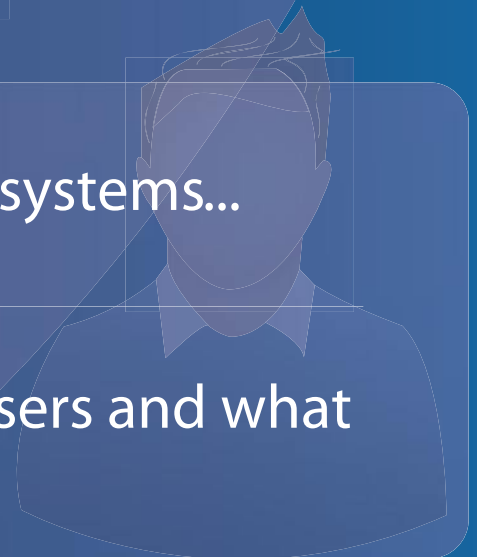
Be reactive...

- Your systems are being attacked 24 hours a day...
- You understand the threats and are protected against them...



Be proactive...

- My users' behaviour threatens our systems...
- I understand what motivates my users and what threats are coming my way...



ID Theft Protect provides information on threats from a user perspective.

IN BRIEF

08 Latest News From IT Security World

By Armando Romeo, eLearnSecurity and ID Theft Protect

As usual specialists from companies eLearn Security and ID Theft protect will share with us latest news from IT security world. Read it to up-date yourself.

BASICS

10 When I'm x64: Bootkit Threat Evolution in 2011

By Aleksandr Matrosov, Eugene Rodionov

It's traditional in security (almost considered compulsory in PR circles) at the end of each year to offer a retrospective view of security-related events in the past 12 months and predictions of likely trends in the threat/anti-threat landscape for the upcoming year. We suspect that by the time this article appears, you'll be sick and tired of crystal balls, but by the end of 2011 we had noted and documented some particularly interesting growth trends in complex threats, especially those targeting the Microsoft Windows 64-bit platform and bootkits in particular (Matrosov).

16 iPhone Forensics on iOS 5

By Satish Bommisetty

Mobile device forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions. The use of phones in crime was widely recognised for many years, but the forensic study of mobile devices is changing every day because of new technologies and advanced mobile operating systems. In this article we will learn about iPhone forensics and the technical procedures & challenges involved in extracting data and artefacts from a live iPhone and iPhone backups. iPhone forensics can be performed on the backups made by iTunes or directly on the live device. In iPhone Forensics, the goal is to extract data and artefacts from an iPhone or backup without altering any information. Having knowledge of different data acquisition types involved in mobile device forensics (http://en.wikipedia.org/wiki/Mobile_device_forensics) would give a better foundation for this article.

22 Striping SSL Encryption

By Praful Agarwal and Sulabh Jain

HTTPS is not an unknown terminology. Hyper Text Transfer Protocol Secure (HTTPS) is a secure version of the Hyper Text Transfer Protocol (HTTP) which is a combination of HTTP with SSL (Secure Socket Layer)/TLS

(Transport Layer Security) (...) This command is the main cause of the attack, as this command basically faking the data packets in the network, poisoning the network clients, making them believe that attacker's machine is now the gateway of the network. After this command is executed, the network traffic, from all the machines in the network, will be redirected to the attacker's machine. And because IP forwarding has been enabled on the attacker's machine, the redirected traffic will be forwarded to the real default gateway to complete the Request-Response cycle. In this situation, the attacker is becoming the man in the middle. Now comes the SSL Strip, open a new console in BackTrack and type in the command `sslstrip -I 8080`.

26 CAPTCHAs, What They Are and How To Use Them

By Federico "Glamis" Filacchione

You've of course always used them. They're those strange letters and numbers below pretty every registration form that exist on the Internet. CAPTCHAs are everywhere, sure, but are they useful? Are they secure? Are they accessible? We'll look at how they're implemented, we'll try to debunk some myth related to them and understand how you can use CAPTCHAs on you web application, and be safe and sound. The CAPTCHA is a challenge-response type of test. It means that the user is given a text, and he's asked to insert that text again in a form. Since this action requires a thought, it's presumed that only a human being can execute the test, and not a bot (which is a program built to behave like a human, but not at that level). There are several types of CAPTCHAs, let's see what the differences are. As said before, graphic-only tests are inaccessible for visually impaired persons. Even using the audio and graphic test the risk to cut out a large number of users is high. This can happen if the user has not the ability to hear the test (hardware problems, lacking of a specific plug-in, etc.) or if the test is taken in a noisy environment.

DEFENSE

30 Performing a History Injection Against the Chrome Web Browser

By Christopher M. Frenz

Over the course of the last couple of decades computers have arisen to a position of prominence across many aspects of people's personal and business life. The chrome history files are actually simple to view and interact with in that they are SQLite databases and can be readily viewed with programs such as the SQLite Database Browser (<http://sqlitebrowser.sourceforge.net/>). In fact viewing the history files which such a program is highly recommended as you can use the browser to not

only view the various tables and the data inside the tables, but also to view the schema that was used to construct the database. The Chrome user data file locations vary somewhat depending on what operating system you are running (Table 1). The above script does raise some further questions about the validity of using link data from Chrome's browser history file. In addition to the commonly asked questions of how do we know if the suspect was the one using the computer at the time and not another person, forensic examiners may also need to consider the additional question of whether or not the data found in the browser history is legitimate or not.

(IL)LEGAL

34 Smart Meters Dumb Regulators

By Drake

Smart metering is the one of the hottest topics in the Energy and Utilities sector in Europe and North America. The historic situation with energy utility billing, predominantly electricity and gas, has been that the estimations of usage have generally been estimated based upon one or two reliable readings from customer meters per year. Consequently, "smart" meters hold a deal or promise. More accurate charging for energy used could be good for the consumer, allowing them to make more informed choices about when and how they choose to use electricity, for example. This is hugely important, as electricity, except that derived from nuclear power stations, costs different amounts to produce at different times of day.

36 Iptables

By Hamidreza Mohebbali

Network security is a primary consideration in any decision to host a website as the threats are becoming more widespread and persistent every day. One means of providing additional protection is to invest in a firewall. Though prices are always falling, in some cases you may be able to create a comparable unit using the Linux iptables package on an existing server for little or no additional expenditure. Creating an iptables firewall script requires many steps, but with the aid of the sample tutorials, you should be able to complete a configuration relatively quickly.

INTERVIEW

52 Interview with Richard Johnson

By Aby Rao

Richard Johnson is a computer security specialist who spends his time playing in the realm of software vulnerability analysis. Richard currently fills the role of principal research engineer on Sourcefire's Vulnerability Research Team, offering 10 years of expertise in the software security industry. Current responsibilities include research on exploitation technologies and automation of the vulnerability triage and discovery process. Past areas of research include memory management hardening, compiler mitigations, disassembler and debugger design, and software visualization. Richard has released public code for binary integrity monitoring, exploit mitigations, program debugging, and reverse engineering and has presented at more than 20 conferences worldwide since 2004. Richard is also a co-founder of the Uninformed Journal and a long time resident of the Hick.org ranch.



Coliseum

Virtual labs
100% practical hands on
training
by eLearnSecurity

FIND OUT
14 educational challenges

- ✓ Real world scenarios
- ✓ No set-up time
- ✓ Play on MS SQL Server
- ✓ Got stuck? We support!

SYMANTEC

A recent law suit has been filed against Symantec for one of their latest Norton products – Norton Utilities. James Gross is seeking class action status and claims their product is Scareware. One can download the Norton software for free and run a diagnostic scan but must pay to have any of the problems resolved. Gross hired a forensics expert to study the software and he discovered that the software always reports the system as having issues no matter what.

Symantec intentionally designed its Scareware to invariably report, in an extremely ominous manner, that harmful errors, privacy risks, and other computer problems exist on the users' PC, regardless of the real condition of the consumer's computer. – states the claim.

Gross is seeking \$5,000,000 in damages and class action status for the suit while Symantec stands by their product and is seeking to fight the accusations in court.

Source: Schuyler Dorsey

SYMANTEC ATTACK

Symantec is also the victim of a recent attack. A hacker successfully stole the source code to their anti-virus software. Symantec announced that their network was not breached but the source was taken from a third-party.

Our own network was not breached, but rather that of a third-party entity... we are still gathering information on the details and are not in a position to provide specifics on the third-party involved. Presently, we have no indication that the code disclosure impacts the functionality or security of Symantec's solutions.

Symantec also reported that the source code stolen belonged to EndPoint Protection 11 and Antivirus 10.2 which are both more than five years old. They believe that this theft presents no immediate danger to Norton customers and that it will not benefit hackers in any way since anti-virus is simply based on signature detection.

Source: Schuyler Dorsey

POS HACK

A group of Romanian hackers has been recently accused of racking up over \$3 million in fraudulent charges. The group has been digitally stealing credit card data from point-of-sale systems in Subway restaurants and 50 other small retailers since 2008, adding up to 80,000 victims. The attacks were made possible by the point-of-sale systems being open and accessible from the

internet. The attackers port-scanned entire blocks of IP addresses looking for a specific remote access software.

Once the attackers gained access to the system, they deployed their own Trojans and loggers which recorded all credit card data coming into the system. All of the data was then offloaded to web servers controlled by the attackers and then uploaded to sendspace.com. Some of the accused then took the credit card information and made fake credit cards for fraudulent purchases.

As the case continues, security experts stress the importance of PCI compliance. The attackers used simple password guessing techniques to crack into these systems and it could have been averted.

Source: Schuyler Dorsey

U.S. CHAMBER OF COMMERCE

The Chamber of Commerce was breached by a group of Chinese hackers and they may have had access to the systems for over a year. The attackers had full access to information on its three million members and successfully downloaded much of the information.

What was unusual about it was that this was clearly somebody very sophisticated, who knew exactly who we are and who targeted specific people and used sophisticated tools to try to gather intelligence, – David Chavern, Chief Operating Officer at the Chamber of Commerce.

This has also raised a lot of questions with regard to company security policies. Many companies are great at controlling their inbound traffic but fail to effectively control outbound traffic. Changes to these rules could greatly minimize the risk and damage if a break –in were to occur.

Some information pointed to it being a state-sponsored attack but this is unconfirmed. Many security experts have speculated about state-sponsored attacks over the past several years as we approach the possibility of cyber warfare.

Source: Schuyler Dorsey

SMART CARDS HACKED

AlienVault has reported that Chinese hackers have successfully found a way to hack U.S. government smart cards. The hackers are using a malware called Sykipot in an attempt to steal data from the Department of Defense. The malware captures PINs used by the smart cards which gives the hackers full access to that information.

AlienVault stated *Like we have shown with previous Sykipot attacks, the attackers use a spear phishing*

campaign to get their targets to open a PDF attachment which then deposits the Sykipot malware onto their machine... Then, unlike previous strains, the malware uses a keylogger to steal PINs for the cards. When a card is inserted into the reader, the malware then acts as the authenticated user and can access sensitive information. The malware is controlled by the attackers from the command & control center.

The malware specifically targets ActivIdentity, the PKI used within the government defense departments. With this information, experts speculate that the Department of Defense was specifically targeted but as to what information they were going after remains unknown.

Source: Schuyler Dorsey

ANDROID.QICSOMOS OPEN SOURCE TROJAN IN CIRCULATION

Security researchers have uncovered a new Android malware that claims to detect Carrier IQ software on an Android device. Android.Qicsomos is a modified version of an open source project which was used to detect Carrier IQ with additional code to dial a PRS SMS number. On start-up the rogue software displays information about the device and will show a message which indicates the Carrier IQ rootkit was not found. Users are then presented with a button to uninstall the Android app, but it's at this point that the rogue app sends an unauthorized PRS SMS. Read this article for more information on PRS SMS rogue apps (<http://www.julianevansblog.com/2012/01/the-sms-premium-rate-service-trojan-mobile-app.html>).

Source: ID Theft Protect

CARBERP TROJAN STEALING UKASH E-CASH VOUCHERS

The Carberp Trojan is targeting Facebook users in an attempt to steal login credentials. Carberp allows its developers to anonymously exploit Facebook users who use Ukash e-cash vouchers. Carberp replaces a user Facebook page and redirects (using a MitB attack vector) the user to a fake page notifying the user that their Facebook account is locked. The user is asked for personal information including a Ukash 20 Euro to unlock the account. The page claims the e-cash voucher will be added to the Facebook account; however it is transferred to the Carberp bot master which then uses it to convert into cash. Ukash like most e-cash voucher systems offers anonymity, so it's easy for the fraudsters to sell the e-cash vouchers anywhere anytime without being traced.

Source: ID Theft Protect

LIVE BLOGGING WEBSITE COVERITLIVE VICTIM OF DATA BREACH

CoveritLive was recently the victim of a hacking attack. The company confirmed that the attack happened earlier this month (Jan), but are not entirely sure on what data was accessed. It did say that no financial data was compromised. The announcement of this data breach came via Twitter. An investigation into how the data was accessed is still in progress. All users have been advised to change their passwords before logging in.

Source: ID Theft Protect

MEMBERS OF TEAMP0IONN HACKER GROUP TARGET T-MOBILE

Hackers from hacker group TeaMp0ioN have recently published login information for T-Mobile staff and administrators. The hackers managed to identify a SQL injection vulnerability on the T-Mobile.com and the *newsroom.T-Mobile.com* websites. The hackers managed to extract names, emails, phone numbers and passwords of all the administrators and staff. The hackers highlighted the manual distribution of passwords via an admin who used the same set of passwords. T-Mobile has recently (16th Jan) confirmed that only the newsroom website was affected.

Source: ID Theft Protect

NSA RELEASES SECURITY ENHANCED ANDROID OS

The *National Security Agency* (NSA) has this week (Jan16th) released a security-hardened version of Google mobile OS Android. The enhanced build is based on SELinux, which was also created by the NSA. The primary function of the hardened Android OS is to limit the behaviour of flawed or malicious Android apps. The hardened OS will no doubt be of considerable interest to mobile developers and security experts. Consumers will not be that interested unless they have been a victim of rogue mobile app and want to learn more. The NSA is waiting for you to call....

Source: ID Theft Protect

When I'm x64:

Bootkit Threat Evolution in 2011

It's traditional in security (almost considered compulsory in PR circles) at the end of each year to offer a retrospective view of security-related events in the past 12 months and predictions of likely trends in the threat/anti-threat landscape for the upcoming year.

What you will learn...

- How major bootkit families have evolved, their differences and resemblances
- How attacks against 64-bit operating systems have become increasingly effective

What you should know...

- Basic understanding of Windows architecture
- Basic understanding of the PC boot process.

We suspect that by the time this article appears, you'll be sick and tired of crystal balls, but by the end of 2011 we had noted and documented some particularly interesting growth trends in complex threats, especially those targeting the Microsoft Windows 64-bit platform and bootkits in particular (Matrosov).

Figure 1 is a (pretty much self-explanatory) diagram depicting the evolution of bootkit threats over time. The left-hand column represents Proof-of-Concept bootkits that have played an important part in the development of this type of threat but haven't had the same impact *in the wild* as widespread malware like Olmarik (TDL4).

eEye's Bootroot was an NDIS backdoor that used customized boot sector code to compromise the kernel during loading.

Vbootkit targeted Vista, the security of which was weakened by two inherent assumptions: that there was no possibility that malware could take hold before the Vista loader kicked in, and that once an executable's integrity has been checked on loading, its image in memory will not change before it's actually loaded (Kumar). Vbootkit version 2 extends the hooking of Int 13h and subsequent patching in memory to Windows 7 (Softpedia).

The Stoned bootkit, of course, has no direct connection with 1987's boot sector/partition sector infector Stoned (a.k.a. New Zealand), arguably one of the most successful viruses (in terms of longevity) of

all time. Its name is, however, clearly quite deliberate: Stoned Bootkit author Peter Kleissner describes Stoned as *probably the first bootkit?* and used a variation on the famous Stoned message *Your PC is now Stoned* in a BlackHat presentation describing the bootkit (Kleissner 2009). Development of a 64-bit version is described in a subsequent document (Kleissner 2011).

The right-hand column shows bootkits that have had more impact *in the wild*. Mebroot has been used by a number of botnets including Torpig (Sinowal). Mebratix writes itself to the MBR, displacing the original MBR code to another sector (the sector varies according to

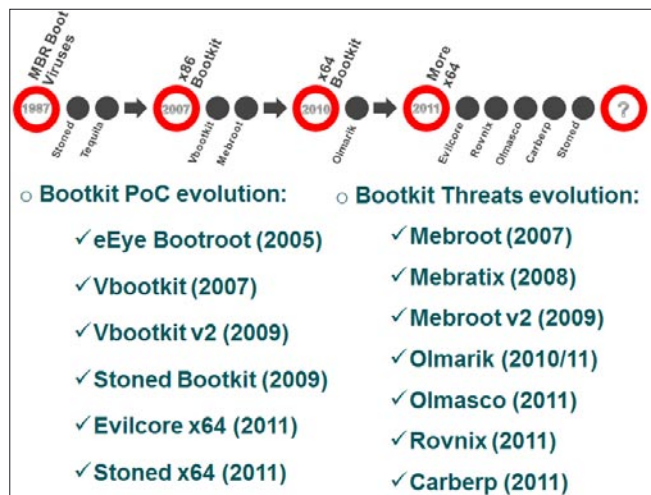


Figure 1. Bootkit threat evolution. (Modified from TDSS part 1: The x64 Dollar Question)

the variant) in something very close to classic BSI (boot sector infector) style: Brain, the great-granddaddy of PC viruses, did something very similar (Harley, Slade, Gattiker, 2001).

As we've focused quite a lot of research attention on other entries in that column, they get sections to themselves.

TDL4 bootkit

As we predicted in 2010, TDL4 (Win32/Olmarik) has been evolving over 2011. Its developers attempted to bypass the KB2506014 security update, which addressed a vulnerability allowing abuse of WinPE mode.

TDL4 could be referred to as the first widely spread bootkit targeting 64-bit systems. In order to get control before the OS loader does, it overwrites MBR code while leaving the partition table untouched. This can be seen in Figure 2.

When the malicious boot code receives control it locates TDL4's hidden storage and continues the boot process using the malware's bootkit components.

A particularly striking feature of TDL4 is that it has implemented various techniques to load its kernel-mode driver on x64 systems (64-bit versions of Microsoft Windows Vista and Windows 7) despite their enforced kernel-mode code signing policy and implement kernel-mode hooks even with kernel-mode patch protection policy enabled.

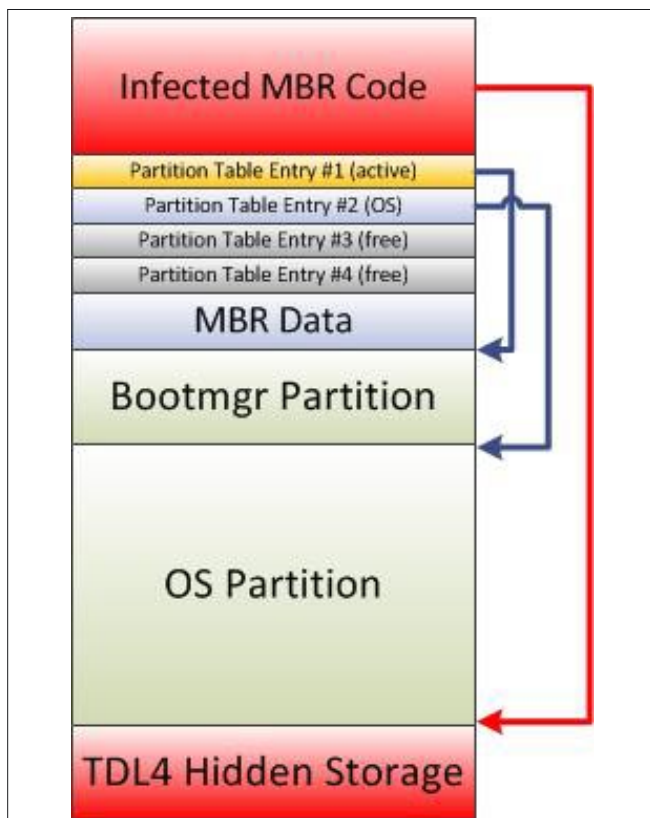


Figure 2. TDL4-infected partition schema

The dropper is unable to load the kernel-mode driver on x64 operating systems it is unable to load the kernel-mode driver, as the driver isn't signed. To get round this limitation the dropper wrote all its components directly to the hard disk, sending `IOCTL_SCSI_PASS_THROUGH_DIRECT` requests to a disk class driver. Having obtained the disk's parameters, it created an image of its hidden file system in the memory buffer and then wrote it to the hard drive. Subsequently, it modified the MBR so that its malicious components were loaded at boot time. Then it rebooted the system, calling the `ZwRaiseHardError` routine and passing `OptionShutdownSystem` as its fifth parameter. This routine resulted in the system's displaying a Blue Screen of Death and rebooting.

In our report *The Evolution of TDL4: Conquering x64* we described a method used by the TDL4 bootkit to load its malicious unsigned driver on 64-bit systems. Subsequently, Microsoft released a patch addressing the vulnerability in x64 OS's (Windows Vista and later) exploited by TDL4, which centred on the way in which the OS checked the integrity of loaded modules.

On unpatched systems there were three BCD (*Boot Configuration Data*) options that determined how the OS checked integrity of the kernel-mode modules:

On a patched system only two of these are left: `BcdLibraryBoolean_DisableIntegrityCheck` and `BcdLibraryBoolean_AllowPrereleaseSignatures`. `BcdOSLoaderBoolean_WinPEMode` BCD option is no longer used in the initialization of code integrity policy. The routine `BllmgQueryCodeIntegrityBootOptions` in `winload.exe` returns the value that determines code integrity policy, and the function was probably added in order to increase the size of the export directory so that the TDL4 bootkit was unable to replace it.

Subsequently, however, an updated version of the TDL4 bootkit worked around this patch by introducing modifications in the `ldr16` component in order to reintroduce the ability to successfully infect x64 architecture.

The intention was to modify the `ld32` or `ldr64` components of `kdcom.dll`, according to the system targeted. Rather than switching into WinPE mode, this version of TDL4 patched `I_CheckImageHashInCatalog`, a routine used to validate the integrity of the modules being loaded by `winload.exe`.

When the `I_CheckImageHashInCatalog` routine fails to verify the integrity of a module, the value `0xC0000428` (`STATUS_INVALID_IMAGE_HASH`) is returned, preventing the system from booting. However, the bootkit patched this routine so as to make it return `0x0000C428` instead of `0xC0000428`. This latest value is not an error code per se (error codes in kernel-mode normally have the most significant bit set to 1), so the replacement of `kdcom.dll` was not detected by the operating system.

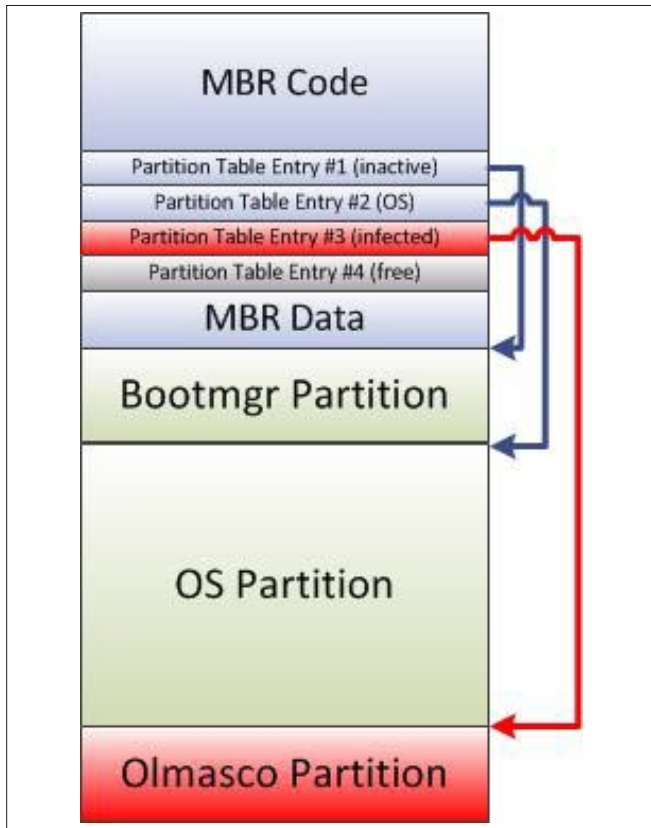


Figure 3. Olmasco-infected partition schema

Olmasco Bootkit

At the beginning of 2011 a brand new bootkit, Win32/Olmasco (also known as MaxSS), appeared in the wild. This new malware family is based on enhanced techniques developed and evolved within the TDL4 family. Unlike TDL4, Win32/Olmasco modifies the partition table of the disk rather than patching MBR code. It looks for an empty entry in the partition table and free space at the end of the hard drive in order to create a new hidden partition containing payload and configuration information. Figure 3 illustrates the way in which partitions are laid out on the infected hard drive: Figure 3.

Here is the beginning of the Win32/Olmasco VBR (Volume Boot Record): Figure 4.

The VBR of the malicious partition mimics the VBR of the legal NTFS partition: this approach makes Win32/Olmasco stealthier and therefore more difficult to detect.

```

0000 EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  WPNTFS .....
0010 00 00 00 00 00 00 F8 00 00 3F 00 FF 00 53 AC FF 00  ....°..?. .SM .
0020 00 00 00 00 00 00 80 00 00 9C 53 00 00 00 00 00 00  ....A.A.bs.....
0030 39 05 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00  9.....
0040 F6 00 00 00 01 00 00 00 8B 62 C8 E9 B8 4B 28 D5  Ÿ.....лbLш-K(-
0050 00 00 00 00 FA 31 C0 8E D0 BC 00 7C FB 0E 1F 0E  ....*1LO|-.|v...
0060 07 66 60 88 16 00 7E C6 06 04 7E 1E B4 48 BE 04  .F`И..~|..~.+H-.
0070 7E CD 13 B0 50 0F 82 71 01 83 2E 13 04 14 A1 13  ~=-.P.Bq.Г....б.
    
```

Figure 4. Olmasco VBR

ZeroAccess Rootkit

Early in 2011 a new 64-bit ZeroAccess (Win32/Sirefef) modification appeared in the wild. Unlike TDL4 and Win32/Olmasco, Sirefef doesn't implement bootkit functionality. Although there is a version of the malware targeting 64-bit systems, it doesn't contain a kernel-mode driver. The only ZeroAccess version which does include a driver targets x86 systems only. For this reason, the machine-infection algorithm is different for 32- and 64-bit Operating System versions.

On x86 systems ZeroAccess behaves like the TDL3 rootkit. It infects a kernel-mode boot start driver by completely overwriting the driver with its own code. As a result, at boot time the system loads the malicious driver, not the original, legitimate code. In order to protect itself against security software and to conceal the fact that the driver was overwritten it sets up low-level hooks in the storage device driver stack.

On a 64-bit OS, however, since there is no kernel-mode 64-bit driver, the malware drops consrv.dll library into the `systemroot\system32` directory and registers it as part of the Windows Subsystem, which has to be loaded by the *Session Manager Subsystem (smss.exe)* process (trusted system process) during system startup. The list of subsystems which need to be loaded is stored under the Required value of the `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems` registry key. If one of the components of *required* subsystems is missing the system is rendered unbootable. Thus, removing the threat by deleting `consvr.dll` without applying corresponding changes to the registry key will break the system.

Rovnix and Carberp bootkits

In 2011 a new bootkit – Win32/Rovnix – established a new trend: modification of the VBR and Bootstrap Code (Win32/Rovnix, Win32/Carberp). Using such a technique allowed malware to bypass many security and antivirus programs since the feature makes detecting and removing these threats more difficult.

Interestingly enough, the bootkit builder for VBR bootkits like Win32/Rovnix was offered for sale. For instance, Win32/Carberp – one of the most dangerous banking trojans – was upgraded to include bootkit functionality. Its developers started testing the for-sale bootkit in the end of the summer.

```

seg000:01EA halt:                                ; CODE XREF: seg000:00751j
seg000:01EA                                     ; sub_B0+351j ...
seg000:01EA             hlt
seg000:01EA sub_191         endp
seg000:01EA
seg000:01EB ; -----
seg000:01EB             jmp     short halt
seg000:01EB ; -----
seg000:01ED aBoot         db '\boot',0          ; DATA XREF: seg000:008E1r
seg000:01F3             db     0

```

Figure 5. Loading the file "\boot"

The bootkit component of Carberp is almost identical to that of the Rovnix bootkit, which we discussed in more technical detail in the slide deck *Defeating x64: Modern Trends of Kernel-Mode Rootkits* accompanying a talk given at the Ekoparty 2011 Security conference.

However, the installer had changed significantly. In addition to installing the bootkit it now tried to exploit several vulnerabilities in order to escalate its privileges. This was necessary as Carberp requires administrative privileges in order to install the bootkit. Primarily, Carberp targets corporate users using RBS (*Remote Banking Systems*) software which often lacks administrative privileges, so that an attack relying purely on social engineering doesn't cut the mustard. The installer exploited the following vulnerabilities in the system software in order to escalate privilege:

- MS10-073: a *win32k.sys* `KeyboardLayout` vulnerability in Windows 2000 and XP, originally exploited by Stuxnet. It worked by loading a specially crafted keyboard layout file, making it possible to execute arbitrary code with SYSTEM privileges. Privilege escalation occurred while dispatching input from the keyboard using the `NtUserSendInput` system service in the *Win32k.sys* module.

- MS10-092: a Task Scheduler vulnerability also exploited by Stuxnet, which actually worked on 64-bit systems as well. It worked because in order to protect the integrity of the job configuration files Task Scheduler calculated a CRC32 checksum, fine for detecting unintentional errors but making it easy to create another message with the same checksum.
- MS11-011 (*win32k.sys* `SystemDefaultEUDCFont` vulnerability described at <http://support.microsoft.com/kb/2393802>).
- .NET Runtime Optimization vulnerability triggered by insecure permissions in the service's .EXE file (see <http://osvdb.org/show/osvdb/71013>).

Carberp and its relationship with the Black Hole bootkit has been discussed at some length in the ESET Threatblog and a white paper.

Rootkit Hidden Storage

We are now seeing that more and more complex threats have started using their own hidden storage and avoid relying on services provided by OS. This approach allows malware to keep its payload and configuration data secret where antivirus and security software is less likely to find it, as well as evading security measures

```

unsigned int __stdcall RkFsLoadFile(FS_DATA_STRUCT *a1, PDEVICE_OBJECT DeviceObject, const char *FileName,
{
    unsigned int result; // eax@1

    result = RkFsLocateFileInDir(&a1->root_dir, FileName, FileEntry); // locate file in the root dir
    if ( (result & 0xC0000000) != 0xC0000000 )
    {
        result = RkFsReadFile(a1, DeviceObject, FileEntry); // read the file from the hard drive
        if ( (result & 0xC0000000) != 0xC0000000 )
        {
            result = RkFsCheckFileCRC32(FileEntry); // verify file integrity
            if ( result == 0xC000003F )
            {
                MarkBadSectorsAsFree(a1, FileEntry->pFileEntry); // free occupied sectors
                RkFsRemoveFile(a1, &a1->root_dir, FileEntry->pFileEntry->FileName); // remove corresponding entry
                RkFsFreeFileBuffer(FileEntry);
                RkFsStoreFile(a1, DeviceObject, &a1->root_dir); // update directory
                RkFsStoreFile(a1, DeviceObject, &a1->bad_file);
                RkFsStoreFile(a1, DeviceObject, &a1->bitmap_file); // update bitmap of occupied sectors
                RkFsStoreFile(a1, DeviceObject, &a1->root); // update root directory
                result = 0xC000003Fu;
            }
        }
    }
    return result;
}

```

Figure 6. Removing a corrupted file

References

- <http://www.slideshare.net/matrosov/defeating-x64-modern-trends-of-kernelmode-rootkits>
- <http://resources.infosecinstitute.com/tdss4-part-1/>
- http://www.eeye.com/eEyeDigitalSecurity/media/ResearchPapers/eEyeDigitalSecurity_Pixie-Presentation.pdf?ext=.pdf
- <http://www.blackhat.com/presentations/bh-europe-07/Kumar/Presentation/bh-eu-07-kumar-apr19.pdf>
- <http://news.softpedia.com/news/Windows-7-Vbootkit-2-0-Attack-Tool-Goes-Open-Source-111063.shtml>
- <http://www.blackhat.com/presentations/bh-usa-09/KLEISSNER/BHUSA09-Kleissner-StonedBootkit-SLIDES.pdf>
- <http://stoned-vienna.com/downloads/The%20Art%20of%20Bootkit%20Development.pdf>
- <http://www.cs.ucsb.edu/~seclab/projects/torpig/index.html>
- *Viruses Revealed: David Harley, Robert Slade, Urs Gattiker; Osborne 2001*
- http://go.eset.com/us/resources/white-papers/The_Evolution_of_TDL.pdf
- <http://blog.eset.com/2011/04/15/kb2506014-kills-tdl4-on-x64>
- <http://blog.eset.com/2011/05/10/the-co-evolution-of-tdl4-to-bypass-the-windows-os-loader-patch-kb2506014>
- <http://blog.eset.com/2011/10/18/tdl4-rebootedhttp://www.eset.eu/encyclopaedia/win32-sirefef-a-trojan-dropper-pmax-a-horse-trojandropper>
- http://go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf
- <http://blog.eset.com/2010/10/15/win32k-sys-about-the-patched-stuxnet-exploit>
- <http://go.eset.com/us/resources/white-papers/carberp.pdf>

integral to the operating system. It is interesting and instructive to compare (briefly, on this occasion) the hidden file systems of the most sophisticated of these threats: Win32/Olmasco, TDL4 and ZeroAccess.

TDL4 Hidden Storage

As the successor of TDL3 and TDL3+ this family of malware inherits almost all the features of its

predecessors regarding the storage of payload modules. It reserves some space at the end of the hard drive for housing its file system, the contents of which are protected by low-level hooks and an RC4 stream cipher. TDL4 uses the same technique for allocating space on a hard drive for its file system as its predecessor; namely, it starts at the last but one sector of the hard drive and grows towards start of the disk space. However, there are changes in the layout of the file system compared to that used in TDL3. ESET's white paper *Evolution of TDL: Conquering x64*. includes in-depth analysis of TDL4. The bootkit protects the contents of its file system by encrypting its blocks. Like TDL3 it uses the RC4 encryption algorithm, a stream cipher with varying key length. However, TDL4 uses as a key the 32-bit integer LBA of the sector block being encrypted.

Win32/Olmasco Hidden Storage

The Win32/Olmasco developers went even further in the design and implementation details of its hidden file system. Generally, Olmasco's system resembles a schema which is used by TDL4 but with additional enhancements:

- A supporting hierarchy with files and folders;
- Verification of file integrity to check if its components are corrupted;
- Better management of internal file system structures.

```

unsigned int __stdcall RKFLoadSystemFiles(FS_DATA_STRUCT *FsData, char *Data)
{
    FS_DATA_STRUCT * _FsData; // esi@1
    unsigned int result; // eax@3
    FS_LIST_ENTRY_STRUCT *u4; // ecx@6
    FS_LIST_ENTRY_STRUCT *FileEntry; // [sp+14h] [bp+8h]@3

    _FsData = FsData;
    if ( _FsData && Data )
    {
        FsData->field0 = (int)Data;
        FsData->root.pFileEntry = (FS_FILE_ENTRY_STRUCT *)Data;
        FsData->root.FileBuffer = (int)Data;
        FileEntry = &FsData->bad_file;
        result = RKFSLocateFileInDir(&_FsData->root, "$bad", &_FsData->bad_file);
        if ( (result & 0xC0000000) == 0xC0000000 )
        {
            _FsData->field0 = 0;
            _FsData->root.pFileEntry = 0;
            _FsData->root.FileBuffer = 0;
            return result;
        }
        result = RKFSLocateFileInDir(&_FsData->root, "$bitmap", &_FsData->bitmap_file);
        if ( (result & 0xC0000000) == 0xC0000000 )
        {
            u4 = FileEntry;
LABEL_7:
            _FsData->bad_file.pFileEntry = 0;
            _FsData->root.FileBuffer = 0;
            _FsData->root.pFileEntry = 0;
            _FsData->field0 = 0;
            u4->pDirEntry = 0;
            return result;
        }
        result = RKFSLocateFileInDir(&_FsData->root, "\\", &_FsData->root_dir);
        if ( (result & 0xC0000000) == 0xC0000000 )
        {
            FileEntry->pDirEntry = 0;
            u4 = &_FsData->bitmap_file;
            _FsData->bitmap_file.pFileEntry = 0;
            goto LABEL_7;
        }
        result = 0;
    }
    else
    {
        result = 0xC0000000u;
    }
    return result;
}

```

Figure 7. Olmasco file system initialization

Resources

- *TDSS part 1: The x64 Dollar Question*
By Aleksandr Matrosov, Eugene Rodionov & David Harley, April 2011
Considers and contrasts the distribution and installation of the TDL3 and TDL4 bootkits.
- *TDSS part 2: Ifs and Bots*
By Aleksandr Matrosov, Eugene Rodionov & David Harley, April 2011
Looks in more depth at the internals of the TDSS malware.
- *TDSS part 3: Bootkit on the other foot*
By Aleksandr Matrosov, Eugene Rodionov & David Harley, April 2011
The last part of the series describes the TDSS loading process.
- *Rooting about in TDSS*
By Aleksandr Matrosov & Eugene Rodionov, October 2010
This article for Virus Bulletin describes a utility for dumping the TDSS rootkit's file system. Originally published in *Virus Bulletin*, October 2010.*
- *Win32/Carberp: When You're in a Black Hole, Stop Digging*
By Aleksandr Matrosov, Eugene Rodionov, Dmitry Volkov and David Harley, December 2011
This paper consolidates information published by ESET and Group-IB researchers on Russian malware that attacks Russian RBS (Remote Banking Systems) transactions: now updated to version 1.1 to include additional material.
- *Modern Bootkit Trends: Bypassing Kernel-Mode Signing Policy*
By Aleksandr Matrosov and Eugene Rodionov, October 2011
This presentation continues the authors' consideration of modern bootkit techniques for evading kernel mode code signing policy as applied to currently In-the-Wild malware.
- *Defeating x64: The Evolution of the TDL Rootkit*
By Aleksandr Matrosov and Eugene Rodionov, May 2011
A presentation for Confidence 2011, held in May 2011 in Krakow, on the analysis and implications of the latest generation of the TDL rootkit (TDL4).

Unlike the TDL4 hidden file system, which is only capable of storing files, the system implemented in Win32/Olmasco could store both files and directories.

For instance, the VBR of Win32/Olmasco's hidden partition includes code to load a file with *boot* name from the root directory '\': Figure 5.

In addition, upon reading a file from the file system Win32/Olmasco performs some checks in order to detect corruption of file contents. An additional field was introduced into the data structure with CRC32 checksumming of the file contents. If Win32/Olmasco detects that a file is corrupted it removes the corresponding entry from the file system and frees the occupied sectors. This is shown in Figure 6.

Since the file system implemented in Win32/Olmasco is more mature than that implemented in TDL4, it therefore requires more efficient management in terms of free space usage and manipulation of data structures. Therefore some special files were introduced to help support file system integrity :

- \$bad
- \$bitmap

Figure 7 shows the code for Win32/Olmasco's file system initialization routine.

Both of these files are at the same hierarchical level in the root directory and are not accessible for payload (i.e., they're for system use only). The purpose of these files is to store information about unused space and

sectors containing corrupted data (just as files with the same names are used in NTFS).

Another reason for introducing these files is to make the file system resemble NTFS more closely, so as to confuse security software.

ZeroAccess Hidden Storage

Yet another malware family that implements its own hidden storage is the ZeroAccess rootkit. Like TDL4 and Win32/Olmasco ZeroAccess stores its payload and configuration information in a custom hidden file system. In this case, however, the contents of the hidden storage are saved into a file in the OS file system rather than directly into hard drive sectors.

For this reason, ZeroAccess creates the directory `WindowsDir\%NtUninstallKB_BotId%\BotId` into which payload and configuration information is stored. In this case, the malware behaves like a file system filter driver: it redirects payload read/write operations to corresponding files in the system, as well as performing transparent file encryption.

ALEKSANDR MATROSOV, EUGENE RODIONOV
Aleksandr Matrosov and Eugene Rodionov are ESET malware researchers based in Moscow, where they are both also lecture at the National Research Nuclear University. David Harley is an established security author and independent researcher and holds the position of Senior Research Fellow at ESET North America.

iPhone Forensics On iOS 5

Mobile device forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions.

What you will learn...

- iPhone Forensics in-detail
- Types of iPhone Forensics
- Technical challenges involved in iPhone Forensics
- Data extraction from a passcode protected and a Non-JailBroken device
- Data extraction from an encrypted iPhone backup
- Usage of iPhone Forensic tools

What you should know...

- Background of digital forensics
- Data acquisition types involved in mobile device forensics

The use of phones in crime was widely recognised for many years, but the forensic study of mobile devices is changing every day because of new technologies and advanced mobile operating systems. In this article we will learn about iPhone forensics and the technical procedures & challenges involved in extracting data and artefacts from a live iPhone and iPhone backups. iPhone forensics can be performed on the backups made by iTunes or directly on the live device. In iPhone Forensics, the goal is to extract data and artefacts from an iPhone or backup without altering any information. Having knowledge of different data acquisition types involved in mobile device forensics (http://en.wikipedia.org/wiki/Mobile_device_forensics) would give a better foundation for this article.

Forensics on Live Devices

Researchers at Sogeti Labs have released open source forensic tools (with the support of iOS 5) to recover low level data from the iPhone. Details below outline their research, giving an overview of the usage of iPhone forensic tools.

iPhone 4 GSM model with iOS 5 is used for forensics. Steps involved in iPhone forensics:

- Creating & Loading forensic toolkit on to the device without damaging the evidence
- Establishing communication between the device and the computer

- Bypassing the iPhone passcode restrictions
- Reading the encrypted file system
- Recovering the deleted files

Creating & Loading forensic toolkit

Imagine a computer which is protected with OS level password – we can still access the hard disk data by booting a live CD or by removing the hard disk and connecting it to another machine. When we compare computers to the iPhone, it is an embedded device. So it is not easy to take out the chips (hard disk) and dump data in it. The iPhone makes chip dumping even more complicated by encrypting the data during storage. In order to perform iPhone forensics, we use Live CD approach. As the iPhone has only one serial port, we are going to load custom OS over USB to access hard disk (NAND chip) of the device. But the problem here is, iPhone only loads firmware which has been signed by Apple.

In order to create and load the forensic toolkit, we need to understand iPhone functions at operating system level. iOS (previously known as iPhone OS) is the operating system that runs on all Apple devices

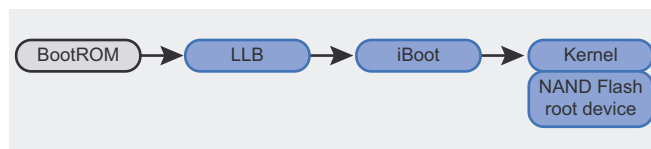


Figure 1. Boot up sequence in Normal mode

like iPhone, iPod, Apple TV and iPad. iOS is a zip file (ships as an *.ipsw* file) that contains boot loaders, kernel, system software, shared libraries & built in applications.

When an iPhone boots up, it walks through a chain of trust which is a series of RSA signature checks among software components in a specific order as shown in Figure 1.

The BootRom is a *Read only memory* (ROM) and it is the first stage of booting an iOS device. BootRom contains the Apple root certificates to signature check the next stage.

The iPhone operates in 3 modes – Normal Mode, Recovery Mode, DFU mode.

In Normal mode, BootRom starts off some initialization stuff and loads the *low level boot* loader (LLB) by verifying its signature. LLB signature checks and loads the stage 2 boot loader (iBoot). iBoot signature checks the kernel & device tree and kernel signature checks all the user applications.

In DFU mode, the iPhone follows the boot sequence with a series of signature checks as shown in Figure 2. BootRom signature checks the second level boot loaders (iBSS, iBEC). Boot loader signature checks the kernel and kernel signature checks the Ramdisk.

During iOS update, Ramdisk gets loaded into RAM and it loads all other OS components. In Forensics, we will create a custom Ramdisk with all our forensic tool kit and load the custom Ramdisk in iPhone volatile memory. Signature checks implemented at various stages in the boot sequence do not allow loading our custom Ramdisk. To load our custom Ramdisk we have to bypass all these signature checks. In the chain of trust boot sequence, if we compromise one link, we can fully control all the links that follow it. The hacker community have found several vulnerabilities in BootRom by which we can flash our own boot loader and patch all other signature checks in all the subsequent stages. Apart from signature checks, every stage is also encrypted. These encryption keys can be grabbed using JailBreaking tools.

Building custom Ramdisk

First we will build a custom Ramdisk with all our forensic tools and patch the Ramdisk signature checks in kernel. Later, we will use jailbreaking tools to load our kernel by patching BootRom signature checks.

With the open source forensic toolkit released by Sogeti Labs, we can build Ramdisk only on MAC OS X.

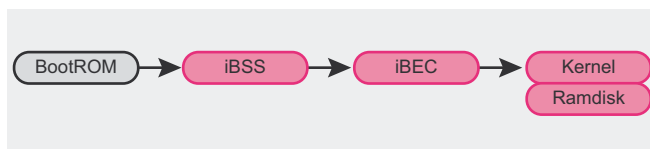


Figure 2. Boot up sequence in DFU mode

During this article, Ramdisk is built on MAC OS X 10.6. The entire forensic toolkit contains python scripts, a few binaries and a few shell scripts.

In order to run the tools, we first need to install all the dependencies (Use the below listed commands from OS X terminal). Download and install Xcode 4. It installs the required compilers (ex: gcc).

Download ldid, grant execute permissions and move it to `/usr/bin` directory. ldid is used for signing the binaries.

```

curl -O http://networkpx.googlecode.com/files/ldid
chmod +x ldid
sudo mv ldid /usr/bin/
  
```

Download and install OSXFUSE. OSXFUSE allows you to extend Mac OS X's native file handling capabilities via a third-party file system.

```

curl -O -L https://github.com/downloads/osxfuse/osxfuse/
OSXFUSE-2.3.4.dmg
hdiutil mount OSXFUSE-2.3.4.dmg
sudo installer -pkg /Volumes/FUSE\ for\ OS\ X/Install\
OSXFUSE\ 2.3.pkg -target /
hdiutil eject /Volumes/FUSE\ for\ OS\ X/
  
```

Download & install python modules – pycrypto, M2crypto, construct and progressbar.

```

sudo ARCHFLAGS='-arch i386 -arch x86_64' easy_install pycrypto
sudo easy_install M2crypto construct progressbar
  
```

Download and install Mercurial (<http://mercurial.selenic.com/>) to check out the source code from the repository.

```

hg clone https://code.google.com/p/iphone-dataprotection/
cd iphone-dataprotection
  
```

Compile `img3fs.c` which is located in `img3fs` folder. This script is used to encrypt and decrypt Ramdisk and kernel. If you run into a problem while running this command, edit the makefile in `img3fs` folder and change the compiler path.

```

make -C img3fs/
  
```

Download `redsn0w` which is a famous *JailBreaking tool*. `Keys.plist` file inside `redsn0w` contains the encryption keys to decrypt Ramdisk and Kernel.

```

curl -O -L https://sites.google.com/a/iphone-dev.com/
files/home/redsn0w_mac_0.9.10b4.zip
unzip redsn0w_mac_0.9.10b4.zip
cp redsn0w_mac_0.9.10b4/redsn0w.app/Contents/MacOS/Keys.plist.
  
```


To patch the signature checks in kernel, supply iOS 5 ipsw file to `kernel_patcher.py`. iOS 5 ipsw file can be downloaded from www.getios.com which maintains all iOS versions for all Apple devices.

```
python python_scripts/kernel_patcher.py IOS5_IPSW_FOR_
YOUR_DEVICE
```

The above python script creates a patched kernel and a shell script to create Ramdisk.

```
sh ./make_ramdisk_n88ap.sh
```

Running the shell script downloads the forensic tool kit (`ssh.tar.gz`) and adds it to the Ramdisk. The Ramdisk image is just a plain HFS+ file system which is native to Mac OS, making it fairly simple to add files to it. All the steps mentioned above create a patched kernel and a custom Ramdisk with forensic tools.

Note

I have created the patched kernel and the custom Ramdisk for iPhone 4. You can directly download these files and skip all the above steps. Download Link: http://www.4shared.com/folder/dKmG68Im/iPhone_Forensics.html.

Loading Forensic Toolkit

In order to load forensic toolkit, supply iOS 5 ipsw file, patched kernel and custom Ramdisk to `redsn0w` tool. Connect the device to computer using USB cable and run the below command. Follow the steps displayed by `redsn0w` to boot the device in DFU mode. In DFU mode, `redsn0w` exploits the BootRom vulnerability and loads patched kernel & custom Ramdisk on to the device.

```
./redsn0w_mac_0.9.10b4/redsn0w.app/Contents/MacOS/redsn0w
-i IOS5_IPSW_FOR_YOUR_DEVICE -r myramdisk.dmg -k
kernelcache.release.n88.patched
```

If the process fails with the *No identifying data fetched error*, make sure that the host computer is connected to the internet. After `redsn0w` is done, the Ramdisk boots in verbose mode. Upon successful boot up, iPhone displays OK message.

Establishing Device to Computer Communication

Once booted with custom Ramdisk, networking capabilities (like WI-FI) are not enabled by default. So a different way is chosen to communicate with the device by following the approach that Apple took with iTunes. USBMUX is a protocol used by iTunes to talk to the booted iPhone and coordinate access to its iPhone services by other applications. USB multiplexing

provides TCP like connectivity over a USB port using SSL. Over this channel iTunes uses the AFC service to transfer files. But here we use this channel to establish a SSH connection and get shell access to the device.

SSH works on port 22. `Tcprelay.py` script redirects port 22 traffic to 2222 port.

```
python usbmuxd-python-client/tcprelay.py -t 22:2222
1999:1999
```

SSH is now accessible at localhost:2222.

```
ssh -p 2222 root@localhost
password: alpine
```

At this point, we get access to the file system. To make things even more complicated, every file is encrypted with its own unique encryption key tied to particular iOS device. Furthermore, the data protection mechanism introduced with iOS 4 adds another layer of encryption that does not give access to the protected files & keychain items when the device is locked. Data protection is the combination of using hardware based encryption along with a software key. Every iPhone (>3gs) contains a special piece of hardware (AES processor) which handles the encryption with a set of hardcoded keys (UID, GUID). The OS running on the device cannot read the hardcoded keys but it can use the keys generated by UID (0x835 and 0x89B) for encryption and decryption. The software key is protected by a passcode and is also used to unlock the device every time the user wants to make use of the device. So in order to access the protected files, we first have to bypass the passcode.

Bypassing the iPhone Passcode Restrictions

Initially (< iOS 4), the passcode was stored in a file which could be removed directly over SSH. Since the introduction of data protection (from iOS 4 on), the passcode is used to encrypt protected files and keychain items on the device. So in order to decrypt the data, we have to supply a valid passcode.

Passcode validation is performed at two levels one at springboard and another one at kernel level. A brute force attack performed at springboard level locks the device, introduces delays and may lead to data wipe-out. However these protection mechanisms are not applicable at kernel level (AppleKeyStore method) leading to brute force attacks. To make brute force attacks less practical, the passcode key derived from the user passcode is tied to hardware UID key. So a brute force attack can only be performed on the device. It is not possible to prepare pre-compute values (like rainbow tables) offline. `Demo_bruteforce.py` script can be used to brute force the 4 digit passcode.

```
python python_scripts/demo_bruteforce.py
```

Port 1999 opened with tcprelay.py is used by the brute force script. It connects to the custom restored_ external daemon on the Ramdisk, collects basic device information (serial number, UDID, etc.), unique device keys (keys 0x835 and 0x89B), downloads the system keybag and tries to brute force the passcode (4 digits only).

Table 1 illustrates the time required to brute force different types of passcodes.

Reading the Encrypted File System

Upon successful passcode brute force, the script automatically downloads the keychain. The keychain is a SQLite database which stores sensitive data on your device. The keychain is encrypted with a hardware key. The keychain also restricts which applications can access the stored data. Each application on your device has a unique *application-identifier* (also called entitlements). The keychain service restricts which data an application can access based on this identifier. By default, applications can only access data associated with their own application-identifier. Later Apple introduced keychain groups. Now applications which belong to same group can share the keychain items. There are two ways to access all the keychain items. One way is, by writing an application and making it as a member of all application groups. The other way is by writing an application and granting com.apple.keystore.access-keychain-keys entitlement.

Keychain database contents can be extracted using keychain_tool.py.

```
python python_scripts/keychain_tool.py -d [UDID]/
keychain-2.db [UDID]/[DATAVOLUMEID].plist
```

Table 1. Brute force time estimation

Passcode Complexity	Brute force time
4 digits	18 minutes
4 alphanumeric	51 hours
5 alphanumeric	8 years
8 alphanumeric	13,000 years

Table 2. Backup Directories

Operating system	Backup Location
Windows XP	C:\Documents and Settings\[user name]\Application Data\Apple Computer\MobileSync\Backup\
MAC OS X	~/Library/Application Support/MobileSync/Backup/ (~ represents user's home directory)
Windows 7	C:\Users\[user name]\AppData\Roaming\Apple Computer\MobileSync\Backup\

To dump the iPhone file system, execute the dump_data_partition shell script.

```
./dump_data_partition.sh
```

The script reads the file system from the device and copies it to UDID directory as an image (.dmg) file. The image file can be opened using the modified HFExplorer that will decrypt the files *on the fly*. To decrypt it permanently, emf_decrypter.py script can be used.

```
python python_scripts/emf_decrypter.py [UDID]/
[data_DATE].dmg
```

emf_decrypter.py decrypts all files in the file system image. To view the decrypted files, mount the file system with below command.

```
Hdituil mount [UDID]/[data_DATE].dmg
```

As soon as the file system is decrypted, there are various files of interest available such as the mail database, the SMS database and location history, etc...

Recovering the Deleted Files

Deleting a file on iPhone, only deletes the file reference. So it is possible to recover the deleted files. To recover the deleted files run emf_undelete.py script.

```
python python_scripts/emf_undelete.py [UDID]/
[data_DATE].dmg
```

With this technique it is possible to recover valuable data like call logs, deleted images, deleted SMS, deleted contacts, deleted voicemail and deleted emails.

Forensics On iPhone Backups

In Forensics we sometimes may end up with the suspect's computer but not the actual iPhone. In this

cf115784cee5e9e0...b12678ef21d7059	Today, 5:22 PM	4 KB	Document
cfa63c2cb0fd8084...5ea3be8d1ba6529	Today, 5:22 PM	4 KB	Document
d1c4ed94e4c4421...201ac5571ee5236	Today, 5:22 PM	4 KB	Document
d1f062e2da26192...274bfd8d07821e4	Today, 5:22 PM	70 KB	Document
d29f4fbbalca2a95d...c1b9c967df6e02d5	Today, 5:22 PM	926 KB	Document
d351344f01cbe49...1fb7ea5614e7c2e5	Today, 5:22 PM	4 KB	Document
dfd4021bd3cd373...675dd96136a1b1c	Today, 5:22 PM	406 KB	Document
e7532f808c1e24e...6ad43b43b7d79f50	Today, 5:22 PM	4 KB	Document
ea4f4a1a45ab93a9...8d298d78686dd4	Today, 5:22 PM	4 KB	Document
edcbc482cd751c4...b2347b80b43b173	Today, 5:22 PM	4 KB	Document
ede88d31c5904c...f0b1cd8310a6ea93	Today, 5:22 PM	4 KB	Document
f7bbe63e61427d2...5726b81289cfa38	Today, 5:22 PM	4 KB	Document
f23461ec2e507af1...1fb5080608024b5	Today, 5:22 PM	4 KB	Document
fb7786ced1add24...8e1ed041e24d52a4	Today, 5:22 PM	4 KB	Document
fb520955c981895...90a46a1ced8c2e9c	Today, 5:22 PM	8 KB	Document
fdad2f81cc0b838d...b14da7ef2d835f3c	Today, 5:22 PM	4 KB	Document
fea8934e2dd1f9e0...b297cf6cc74fd017	Today, 5:22 PM	Zero KB	Document
info.plist	Today, 5:22 PM	12 KB	XML P...rty List
Manifest.mbdb	Today, 5:22 PM	20 KB	Document
Manifest.mbdbx	Today, 5:22 PM	4 KB	Document
Manifest.plist	Today, 5:22 PM	4 KB	XML P...rty List

Figure 3. iPhone backup in unreadable format

References

- iPhone data protection in depth by Jean-Baptiste Bédune, Jean Sigwald – <http://esec-lab.sogeti.com/dotclear/public/publications/11-hitbamsterdam-iphonedataprotection.pdf>
- iPhone data protection tools – <http://code.google.com/p/iphone-dataprotection/>
- Handling iOS encryption in forensic investigation by Jochem van Kerkwijk – <http://staff.science.uva.nl/~delaat/rp/2010-2011/p26/report.pdf>
- iPhone Forensics by Jonathan Zdziarski – <http://shop.oreilly.com/product/9780596153595.do>
- iPhone forensics white paper – <http://viaforensics.com/education/white-papers/iphone-forensics/>
- Keychain dumper – <http://labs.neohapsis.com/2011/02/28/researchers-steal-iphone-passwords-in-6-minutes-true-but-not-the-whole-story/>
- 25C3: Hacking the iPhone – http://www.youtube.com/watch?v=1F7fHgj-e_o
- iPhone wiki – <http://theiphonewiki.com>

iPhone Forensics Video

<http://www.youtube.com/watch?v=2Fs6ee1yeq4&context=C32aee7aADOEgsToPDskLQueZ3jYDdlXdGSYdCN26>

case, Forensics can also be performed on the backups made by iTunes.

When a user connects the iPhone to a computer, iTunes automatically creates a subfolder with device UDID as the name and stores the backup in a path shown in Table 2. Once the subfolder is created, then each time the device is connected to the computer, iTunes will only update the files in the existing subfolder.

iTunes backups everything on the device along with the device details like serial number, UDID, SIM hardware number and phone number. The backup folder contains a list of files which are not in a readable format as shown in the Figure 3.

Most of these files are property list files and SQLite database files. Below are listed the free tools that can be used to convert the gibberish backup files into a readable format as shown in Figure 4.

- MAC OS X – iPhone Backup Extractor – <http://supercrazyawesome.com/>
- Windows – iPhone Backup Browser – <http://code.google.com/p/iphonebackupbrowser/>

File Name	Date Modified	Size	Type
keychain-backup.plist	Today, 5:23 PM	8 KB	XML P...rty List
Library	Today, 5:23 PM	--	Folder
AddressBook	Today, 5:23 PM	--	Folder
Caches	Today, 5:23 PM	--	Folder
Calendar	Today, 5:23 PM	--	Folder
CallHistory	Today, 5:23 PM	--	Folder
com.apple.itunesstored	Today, 5:23 PM	--	Folder
ConfigurationProfiles	Today, 5:23 PM	--	Folder
Cookies	Today, 5:23 PM	--	Folder
Keyboard	Today, 5:23 PM	--	Folder
Logs	Today, 5:23 PM	--	Folder
Mail	Today, 5:23 PM	--	Folder
MobileInstallation	Today, 5:23 PM	--	Folder
Notes	Today, 5:23 PM	--	Folder
Preferences	Today, 5:23 PM	--	Folder
Safari	Today, 5:23 PM	--	Folder
SMS	Today, 5:23 PM	--	Folder
SpringBoard	Today, 5:23 PM	--	Folder
Voicemail	Today, 5:23 PM	--	Folder
Media	Today, 5:23 PM	--	Folder
DCIM	Today, 5:23 PM	--	Folder
PhotoData	Today, 5:23 PM	--	Folder
Recordings	Today, 5:23 PM	--	Folder
mobile	Today, 5:23 PM	--	Folder
ProvisioningProfiles	Today, 5:23 PM	--	Folder
SystemConfiguration	Today, 5:23 PM	--	Folder
TrustStore.sqlite3	Today, 5:23 PM	16 KB	Document

Figure 4. Extracted backup in readable format

iTunes also provides a way for the users to store backups in a secure way by setting a backup password. When a user sets a backup password, all files in the backup get encrypted. The above listed tools do not work with the encrypted backup files. Backup4.py released by Sogeti labs supports viewing the encrypted backups. In order to decrypt the encrypted backup, we first have to brute force the backup password. This can be done by iterating the password supplied to Backup4.py script. If the supplied password is correct, backup4.py will decrypt the whole backup; placing the decrypted contents in the given extracting path. Later, the decrypted files can be analyzed using iPhone Backup Extractor or iPhone Backup Browser.

```
cd iphone-dataprotection
python python_scripts/backups/backup4.py [backup path]
[extracting path] [backup password]
```

If you run into problems during backup4.py script execution, copy the util, crypto, keystore folders which are available in python_scripts folder to backups folder.

Apple is changing the iTunes backup mechanism with every release of iOS. The current release of Sogeti Forensics tools do not support the iOS 5 backups. It is always challenging to design the scripts to decrypt the latest iOS backups.

With the techniques illustrated in the article it is clear that iPhone Forensics is still possible on the latest version of iOS.

SATISH BOMMISSETTY

Satish Bommisetty is an Information Security Professional with 5 years of experience in Penetration testing of web applications and mobile applications. His blog is located at <http://securitylearn.wordpress.com>. Email: satishb3@hotmail.com



CODENAME: SAMURAI SKILLS COURSE



<< Penetration Test Training Samurai Skills >>

- You will learn Real World Hacking Techniques for Targeting , Attacking , Penetrating your target
- Real Live Targets (Websites , Networks , Servers) and some vmware images
- Course Instructors are Real Ethical Hackers With more than 7
- years Experience in Penetration Testing
- ONE Year Support in Forums and Tickets
- Every Month New Videos (Course Updated Regularly)
- Suitable Course Price for ONE Year Support
- Take Our course at your own pace (any time , any where)
- Our Course is Totally Different from Other Courses (new Techniques)

We have Real World Hacking/Penetration Testing Lab with Over 20 Real Target

Striping SSL Encryption

HTTPs is not an unknown terminology. Hyper Text Transfer Protocol Secure (HTTPS) is a secure version of the Hyper Text Transfer Protocol (HTTP) which is a combination of HTTP with SSL (Secure Socket Layer)/TLS (Transport Layer Security).

What you will learn...

- What tool SSL strip is

What you should know...

- Basic knowledge on HTTP

As we all know, HTTPS allows secure online communication which also includes ecommerce transactions, such as online banking.

Web browsers such as Internet Explorer and Firefox display a padlock icon to indicate that the website is secure, as it also displays *https://* in the address bar.

Web servers and Web browsers rely on the *Secure Sockets Layer* (SSL) protocol to create a uniquely encrypted channel for private communications over the public Internet. Each SSL Certificate consists of a public key and a private key. The public key is used to encrypt information and the private key is used to decipher it. When a



Web browser points to a secured domain, a level of encryption is established based on the type of SSL Certificate as well as the client Web browser, operating system and host server's capabilities. That is why SSL Certificates feature a range of encryption levels such as up to 256-bit.

We would like to introduce a tool called *SSL strip* which is based around a *man-in-the-middle*

attack (MITM), where users in a particular network can be forcedly redirected from the secure HTTPS to the insecure version (HTTP) of a web page.

By acting as a man-in-the-middle, the attacker can compromise any information sent between the user and the supposedly secure webpage. The author of the

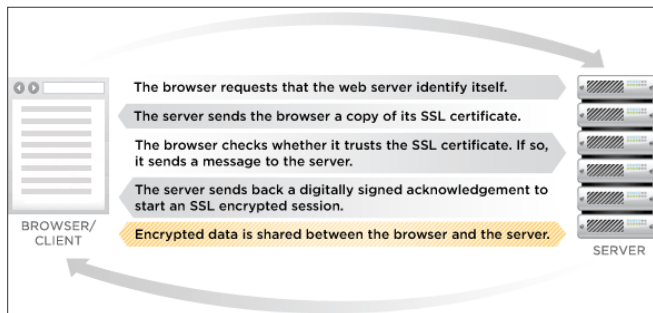


Figure 1. Request and Response Cycle between Client and HTTPS Server



Figure 2. BackTrack First Boot Screen

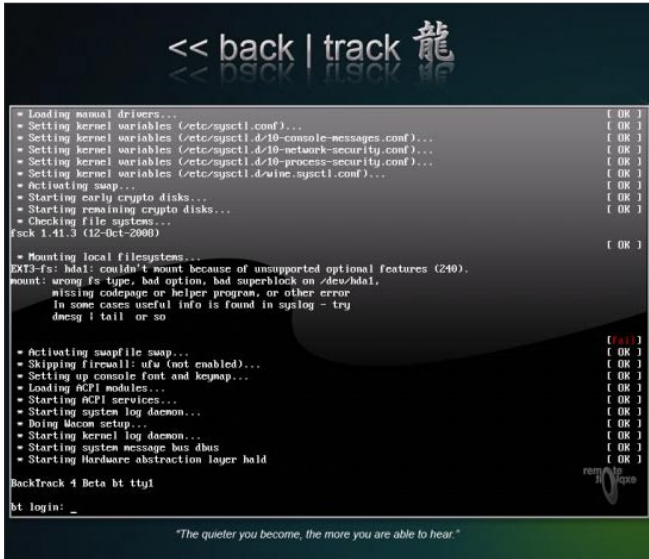


Figure 3. BackTrack Successfully Booted Up, Asking for Login Credentials

exploit claims to have used it to steal data from PayPal, Gmail, Tickermaster, and Facebook – including sixteen credit card numbers and control of more than 100 email accounts.

SSL comes preinstalled in BackTrack, the version which we used for this article is BackTrack Version 4.

BackTrack is a Linux-based penetration testing arsenal that aids security professionals in the ability to perform assessments in a purely native environment dedicated to hacking. Regardless if you're making BackTrack your primary operating system, booting from a LiveDVD, or using your favorite thumbdrive, BackTrack has been customized down to every package, kernel configuration, script and patch solely for the purpose of the penetration tester.

BackTrack is intended for all audiences from the most savvy security professionals to early newcomers to the information security field. BackTrack promotes a quick and easy way to find and update the largest database of security tool collection to-date.



Figure 4. BackTrack Desktop Screen

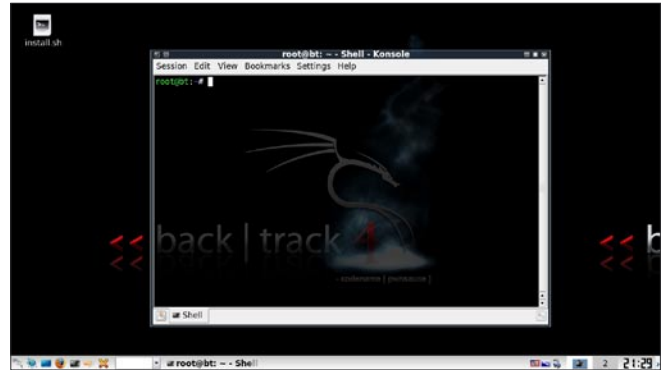


Figure 5. BackTrack with Console

BackTrack can be downloaded from <http://www.backtrack-linux.org/downloads/>. Let's see, how Backtrack works.

After downloading the ISO file of BackTrack, burn a DVD using any disk burning software e.g. Nero. Then insert the burned disk in your PC, make sure that your PC is configured to detect CD/DVD Drive as the First Boot Device. For more info on booting the computer from CD/DVD devices, follow the link http://www.simplyguides.net/guides/boot_from_cd/boot_from_cd.shtml.

You can also make a bootable USB stick for BackTrack. For more information, follow the link http://www.backtrack-linux.org/wiki/index.php/UNetbootin_USB_Installer.

BackTrack will start to boot from the disk. Let the default option be selected and wait for the startup process to begin.

At the login screen, type in `root` as username and `toor` as password. If you still do not get the desktop graphic screen, type `startx`.

To fire up SSL Strip, open console in BackTrack, by click the third icon from left in taskbar at bottom.

In console, type in `echo 1 > /proc/sys/net/ipv4/ip_forward`.

This command enables the data packet forwarding from our machine. By default any modern Linux distributions will have IP Forwarding disabled. This is normally a good idea, as most administrators will not need IP Forwarding, but if we are setting up a Man-In-The-Middle attack, so we will need to enable forwarding. Next command is:

```
iptables -t nat -A PREROUTING -p TCP --destination-port 80 -j REDIRECT --to-port 8080
```

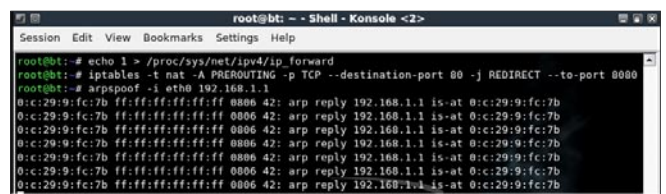


Figure 6. Command Execution in BackTrack Console

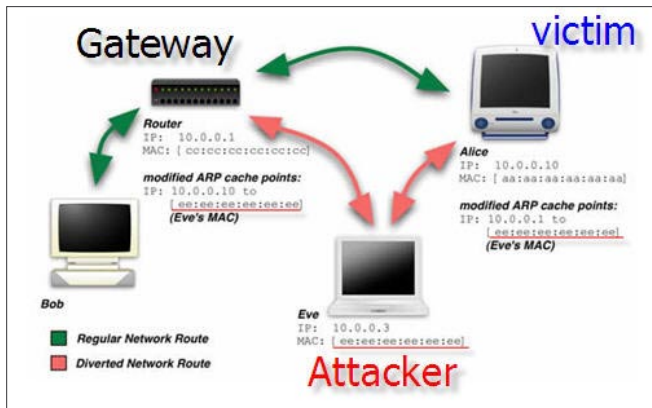


Figure 7. Man-In-The-Middle Attack (ARP Spoofing) Demonstration

The above mentioned command redirects all the data packets travelling in the network which are supposed to reach port number 80 of the server (port number 80 means HTTP server), to port number 8080 (port on which our ssl strip is listening).

Next command is, `arpspoof -i wlan0 192.168.1.1` (where 192.168.1.1 is the IP address of the default gateway of your network). To know about the IP address of the default gateway of your network, type `ipconfig` in console.

This command is the main cause of the attack, as this command basically faking the data packets in the network, poisoning the network clients, making them believe that attacker's machine is now the gateway of the network. After this command is executed, the network traffic, from all the machines in the network, will be redirected to the attacker's machine. And because IP forwarding has been enabled on the attacker's machine, the redirected traffic will be forwarded to the real default gateway to complete the Request-Response cycle.

In this situation, the attacker is becoming the man in the middle.

Now comes the SSL Strip, open a new console in BackTrack and type in the command `sslstrip -l 8080`.

SSL Strip will transparently hijack HTTP traffic on a network, watch for HTTPS links and redirects, then map those links into either look-alike HTTP links or homograph-similar HTTPS links. It also supports

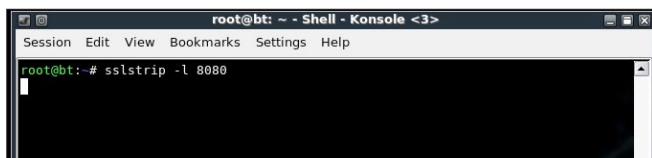


Figure 8. Firing Up SSL Strip Script

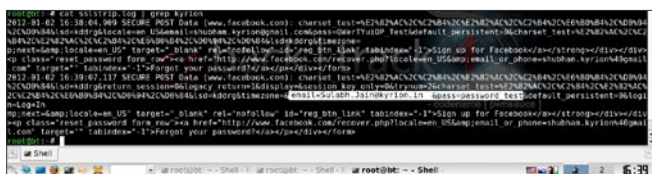


Figure 9. Watching SSL Strip's Log Data in Console

modes for supplying a fav icon which looks like a lock icon, selective logging, and session denial.

Now its time to check the result, in another console type `cat sslstrip.log`. This command will display all the hijacked data including the nasty usernames and passwords if enter by anyone in your network.

We can sense your nasty intentions. Get ready to own the network like never before.

Don't forget the mantra, TALK LESS - LISTEN MORE - HACK EVERYTHING. For us, hacking is not crime, it's our passion and emotion. We are hackers but not criminals for sure.

Disclaimer: Some of the images displayed in the article are copied from internet, the rights for those images remains with the respective copyright holders. As this article is for educational purpose, so its cool to use them.

This was SSL Striping. Wait for the next edition, where we will take you through the thoughts and steps on *Manipulating Chats and Other Network Data* and changing them on the go. Its Praful Agarwal aka SBEZTT and Sulabh Jain, Signing Off.

PRAFUL AGARWAL

+91-9818559358

praful.agarwal@sandrock.in

praful.agarwal@kyrion.in

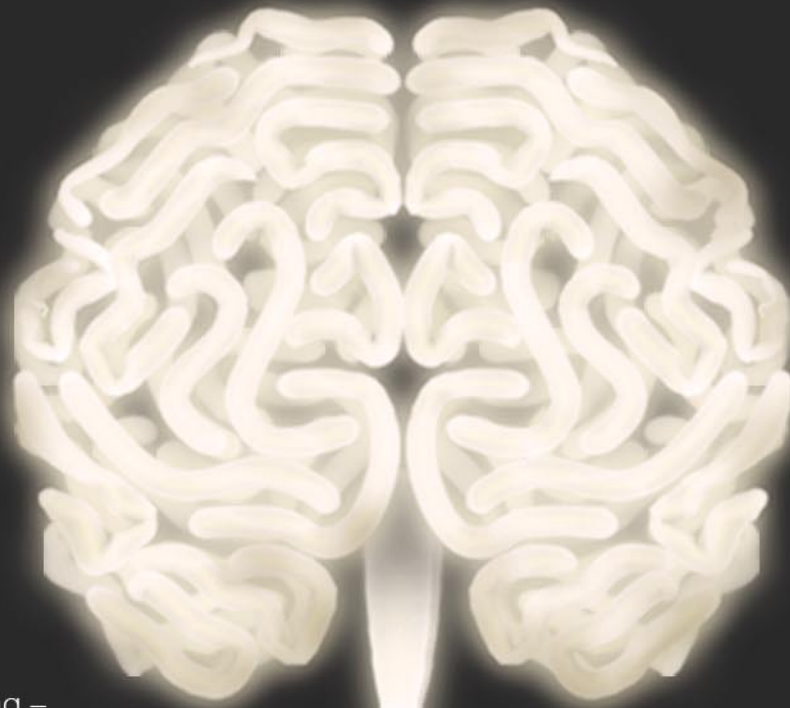
A Seasoned Hacker turned into an Information Security Professional, working on computer codes since the age of 13. Praful is a Research Analyst at Sandrock eSecurities. He also heads the IT department at Kyrion Digital Securities.

SULABH JAIN

sulabh.jain@sandrock.in

sulabh.jain@kyrion.in

Sulabh is Research Analyst – Network Security for Sandrock eSecurities Pvt. Ltd. and Lead Trainer for Kyrion Digital Securities Pvt. Ltd.



Cloud-based training – access content 24/7 from anywhere with ease.

Hands-on labs – gain practical experience from a "hacker's" perspective.

Constantly updated curriculum – new modules added monthly.

Direct mentoring and 1 on 1 instructor interaction.



Content covers:

- Hacking fundamentals
- Recon, network, server, client, and web pentesting
- Pentest structure
- Reverse engineering
- Digital forensics & more!

Teaches the latest offensive security techniques from beginner through cutting edge.

Are you thinking like a
HACKER yet?
www.thehackeracademy.com



CAPTCHAs,

what they are and how to use them

You've of course always used them. They're those strange letters and numbers below pretty every registration form that exist on the Internet. CAPTCHAs are everywhere, sure, but are they useful? Are they secure? Are they accessible? We'll look at how they're implemented, we'll try to debunk some myth related to them and understand how you can use CAPTCHAs on you web application, and be safe and sound.

What you will learn...

- How to use correctly the CAPTCHA tool to improve your website
- Understand better how attacks at web registration forms are made
- A little more security awareness when publish something online

What you should know...

- How to set-up a web registration form and what a CAPTCHA is
- General knowledge of websites accessibility issues
- General knowledge of web security

CAPTCHA is an acronym for Completely Automated Public Turing test to tell Computers and Humans Apart. It's a test invented in 2000 by some researchers in the Carnegie Mellon University, to prevent the illicit use of bots during some web service registration phase. The main use of CAPTCHAs it to prevent a massive registration, for example to a webmail service. In this article we'll describe the different test categories, we'll analyze how much secure is every category, and learn how to avoid the countermeasures used to elude the CAPTCHA and how to set up the correctly to be accessible to everyone.

Chapter 1 – How CAPTCHAs are used

The CAPTCHA is a *challenge-response* type of test. It means that the user is given a text, and he's asked to insert that text again in a form.

Since this action requires a thought, it's presumed that only a human being can execute the test, and not a *bot* (which is a program built to behave like a human, but not at that level).

There are several types of CAPTCHAs, let's see what the differences are.

Graphic CAPTCHA

This is the most common type of test. It's an image in which a text appears more or less distorted, and the user is required to read the text (numbers and/or words) and repeat it in a form.

In Figure 1 you can see how Yahoo! uses the graphic CAPTCHA.

This is pretty easy either to implement it than use it. The user just has to read the text, write it, and he's done. For the provider, in this case of a free webmail service, it's also very easy to set up a procedure that, within a standard form, changes the image every time the page is reloaded.

This is the simplest way to use the CAPTCHA, but is enough? More important, is accessible? The answer to both questions is no. Let's see why.

Audio and graphic CAPTCHA

The second type of the test integrates an audio CAPTCHA to the visual one. This is necessary because since the graphic-only test relies exclusively on sight to be resolved, a visually impaired person cannot register to the service. So in this type of test an audio feature is added to read the text presented (or read another thing, it depends on how the CAPTCHA is used), and allow even a blind person to hear the words and numbers and then insert in the appropriate form.

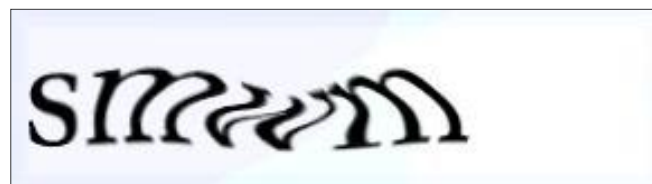


Figure 1. Yahoo! graphic CAPTCHA

In Figure 2 you can see how Google uses this kind of test for the Gmail registration form.

This is a very accessible implementation even because the alt HTML attribute of the CAPTCHA images is *Listen and type the numbers you hear*, so when you click on the wheelchair icon you can see an audio file, in this case some numbers are read, and then complete the test.

As an interesting note the CAPTCHA is localized. In the image you can see the Italian version and the numbers are actually read by an Italian voice. The same happens for the English registration form. Another thing to notice is that in this case the test has *two* ways to be resolved, since either the word presented or the numbers read are valid to pass.

Cognitive CAPTCHA

There is another way to use the CAPTCHA, and it's more complex and really more fascinating when used in social networks. This kind of CAPTCHA asks the user to do a logical operation, instead of just showing a word it asks things like: *how are 3 + 3 / 2?* or *What is the color of the sky, when it's not raining?*

Another option to set up a cognitive CAPTCHA is to use information that the user *and* the service provider already know. This works very well in a social network, and has been implemented, of course, by Facebook.

If Facebook thinks that a login is possible illicit (maybe looking at the usual IP of the user, or something else), it shows the user some photos of his friends, asking to identify who is a particular person.

NOTE 1

GoRumors.com – Facebook Mulls New Social CAPTCHA Security System <http://gorumors.com/social/facebook-social-captcha-system/40583>.

This kind of CAPTCHA is however very hard to set up, since it needs either a very large database of questions, we'll return on this security issue later, or knowledge of user's information. So it's hard to use on non-authenticated forms, like a query on a public database for a WHOIS lookup, for example.

So, how about the accessibility?

Inaccessibility in CAPTCHAs is a real issue, and it has been discussed multiple times over the last year. Even the World Wide Web Consortium published a document in 2005.

NOTE 2

Inaccessibility of CAPTCHA – Alternatives to Visual Turing Tests on the Web, W3C 23/11/2005 <http://www.w3.org/TR/turingtest/>.

The W3C document is not really actual in some passages, but the key findings are really worth reading, and the problems it talks about are real, and must be considered by anyone who uses CAPTCHAs in his services.

As said before, graphic-only tests are inaccessible for visually impaired persons. Even using the audio and graphic test the risk to cut out a large number of users is high. This can happen if the user has not the ability to hear the test (hardware problems, lacking of a specific plug-in, etc.) or if the test is taken in a noisy environment. The Google audio test is very hard to understand even in a normal situation by a non-audio impaired person, this happens because too much security has won over too less accessibility.

Even the cognitive CAPTCHA has a lot of accessibility issues, starting from blind people who cannot see the images. This however is a complex situation, because a social network itself is not the most accessible place on the Internet, isn't it?

Anyway a solution to this can be use information that you already have, i.e. the user's birthday, his favorite soccer team or asking the user a question he has set up in the registration process. This can help to implement a cognitive CAPTCHA and make it accessible to anyone.

Chapter 2 – Are CAPTCHAs secure?

Let's talk security now. As for any kind of tool used to prevent an illicit use of a service, even CAPTCHAs have been attacked, and since they protect mainly webmail registration forms, the attacks came from spammers who wanted to register and use a lot of free email.



Figure 2. Gmail graphic and audio CAPTCHA

Here we'll discuss some of the methods used to break the tests, but consider that this kind of attacks is constantly developing, and researchers are presenting more and more new ideas every year.

NOTE 3

How to break captchas – Blackhat SEO – <http://www.blackhat-seo.com/2008/how-to-break-captchas/>.

Break a CAPTCHA by resolving it automatically

As said, the CAPTCHA was invented to prevent a bot to fulfill a request. But bot have been designed constantly to read the image, and pass the test. This kind of attack is almost old as the CAPTCHA itself.

NOTE 4

Greg Mori, Jitendra Malik – *Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA*. University of California, Berkeley 2003.

Since this kind of attack is pretty similar to a scanner OCR functionality, the easiest way to try to evade the attack is to distort the text in the image, and add more complexity.

In Figure 3 you can see the same Yahoo! CAPTCHA, with more distortion added.

In this case however, security wins over accessibility. Since the more you distort the text the less it became readable, even by a non-visually impaired person.

How many times you have been asked to read an impossible to understand word in a CAPTCHA? Too many...

Breaking the CAPCHTA automatically of course works even on the audio test. A lot of programs are perfectly able to understand spoken words and put down to a file.

Similarly to the graphic counterpart, evading this attack means using the same technique: distortion (see Note 5). You can add a background noise, distort the voice or make the words less intelligible.

NOTE 5

Bohr, Shome, Simon – *Improving Auditory CAPTCHA Security*. ISR Technical Report 2008-31.

But this is not enough, and causes a lot of problems too. A recent study on this topic (see Note 6) shows that even adding distortion to the audio file, the CAPTCHA was automatically resolved 67% the times

for Google, 70% for DIGG and 45% for reCAPTCHA (which is by far the most popular CAPTCHA service used in the Internet). And of course the more noise you add, the less understandable the audio becomes.

Talking about the cognitive CAPTCHA, its weakest links is the database of the questions. Because if combinations of words and numbers are limitless, I need to set up a list of logical questions to ask, and it's easy for an attacker to collect them, resolve them and use the service as he prefers.

NOTE 6

Tam, Simsa, Hyde, Von Ahn – *Breaking Audio CAPTCHAs*. NIPS 2008, MIT Press.

If the cognitive CAPCTHA is based on informations shared between the users and the service provider, this is hard to attack, but this requires a non-public service, so it's not usable in a registration form.

Exploit the CAPTCHA generation algorithm

As any random number generators, the algorithm behind the CAPTCHA can be exploited to know the test before it's presented to the user.

This kind of attack is very hard to complete though. The modern algorithms are well written, open source reviewed, and it's unlikely that an attacker would spend a lot of resources breaking this kind of logic.

However this can happen if a service provider uses a very old CAPTCHA generation system, and that's why a lot of providers use reCAPTCHA, which is very secure on this side.

Manually resolve the CAPTCHA

This can be a contradiction: if I have to automatically attack a CAPTCHA why should I try to attack it by *resolving it manually?*

The answer to the question is easy: if a CAPTCHA is made to be resolved by humans, let the human resolve it! This can be done in two ways.

Embedding the CAPTCHA you want to break in another web service, run by the attacker. For example an attacker can use a Gmail registration form CAPTCHA to allow users to see pornographic videos on the attacker's website. In this case the users will resolve the tests for the attacker.

The other way is to hire people to resolve as much CAPTCHA they can. This case has been quoted in the W3C report (see Note 2), showing that an operator can resolve hundreds of CAPTCHAs every hour.

These kinds of attacks are extremely expensive, and they're not worth the effort, since an automatic bot would always be faster than a human, and it's very complex to set up a *fake* website that uses CAPTCHA

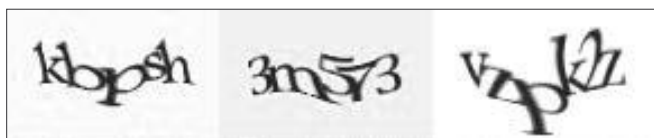


Figure 3. Yahoo! CAPTCHA distorted to avoid automatic reading

from another service. Even the pornographic offer is very useless, since there are a lot of free images and video available everywhere on the net, so it's unlikely that a user would resolve a CAPTCHA to see some tits. This is a little more than an urban legend.

NOTE 7

Whereas it is trivial to write a bot that abuses an unprotected site millions of times a day, redirecting CAPTCHAs to be solved by humans viewing pornography would only allow spammers to abuse systems a few thousand times per day. The economics of this attack just don't add up: every time a porn site shows a CAPTCHA before a porn image, they risk losing a customer to another site that doesn't do this – The Pornography Attack is Not a Concern – Captcha.net.

Chapter 3 – Do I have to monitor the activity on my CAPTCHAs?

The answer to this question is definitely yes. You do need to log everything about your CAPTCHAs, and you need to analyze the log very carefully.

This is needed for both security and accessibility. You need to look at how many successful tests are recorded and how many users are failed the test, and if you implement ad audio and video CAPTCHA, you'll need to know even if the user has passed the test reading the words or hearing the audio.

Analyzing these logs the service provider can understand if the test is really useful, if it's too much difficult to pass, or if it's too weak or too simple. For example if you think that your service will be used by a very small number of visually impaired users, and your logs tell that your CAPTCHA is passed 50% video and 50% audio, this can mean either that the audio part is too weak to attacks, or that the visual one is too difficult to be read. And knowing this you can really improve your service and your user's experience.

Correlating this log with other sources, for example application servers logs or firewalls/IPS logs, you can also have a clear situation on who, when and from where are using (or maybe abusing) your service.

Conclusions

So we have made a tour on the most common types of CAPTCHA you can use, what are the strengths and weaknesses of every type, and how you can detect abusers or attackers.

As any security measure, CAPTCHAs cannot guarantee you 100% of security. This tool is useful, but needs to be implemented very well, you need to think about accessibility, user experience, and these are critical problems if your service has a broad

audience, much more if you are publishing a service in a government or public website.

You can enforce CAPTCHA security by adding complexity, distortion, or difficult logic questions, but you need to be aware that the more complexity you add, the more security you gain, but the less accessible your service became. And that's why a lot of CAPTCHA implementations out there are really a pain to get passed through and this causes frustration in users. And you really don't want that.

Together with a good CAPTCHA, well balanced between security and accessibility, and well monitored, you can also set up another control of you service, based on usage of the service itself. If the web application thinks that a user is doing too many operations than a normal human can do, then it presents a CAPTCHA.

This is exactly what Google does if you try to do a lot of search queries from an automated tool. Of course Google can't show a CAPTCHA every time a user search something, but if the search engine thinks that you're a bot and not a human, it stops everything and presents to you a cognitive CAPTCHA. And this works very well to prevent abuses.

To conclude: always respect your users, think what kind of service you're trying to protect, and keep looking at the logs.

And you'll get a good CAPTCHA!

FEDERICO 'GLAMIS' FILACCHIONE

Federico 'glamis' Filacchione is born and living in Rome, Italy. He works in the IT industry for more than 10 years, most of them trying to spread security awareness and convince his colleagues and bosses that security is not a tool you can buy, but a new way to think about the same old stuff. He loves quoting Schneier's „You can't defend. You can't prevent. The only thing you can do is detect and respond“, but still finds people who doesn't understand the meaning.

You can read his thoughts (in Italian) on glamisonsecurity.com, follow him @glamis on Twitter, or contact him at glamis@glamisonsecurity.com.

Performing

a History Injection Against the Chrome Web Browser

Over the course of the last couple of decades computers have arisen to a position of prominence across many aspects of people's personal and business life.

What you will learn...

- How to view the data stored in Chrome's History database
- How to inject falsified entries into Chrome's History database

What you should know...

- Some basic familiarity with SQL statements would be helpful in understanding the sample code.

The expansion in the role of computers has included both positive aspects as well as negative ones, and as such there has been recent expansion in the field of digital forensics as means of gathering data from computer systems believed to have been involved in the execution or planning of a crime. For example, browser histories have been introduced in various criminal cases, where browser histories describing searches for terms like *handguns*, *silencers*, and *neck break*, along with visits to sites containing content that is descriptive of violent or *questionable* content, were considered damning evidence (http://news.cnet.com/8301-13578_3-10452471-38.html).

Of course there has been much media attention given to these issues, particularly in the area of ways people can protect their privacy by clearing their histories or by surfing the Web in a private browsing mode in order to prevent histories from being recorded, but what of the question of the accuracy of these histories themselves? How reliable can these history files really be considered? In other words, would it be possible for someone to falsify a browser history to make it appear

as if a person had visited a site that he had never before visited? In this article, we will attempt to answer that question by examining the history file for the chrome browser and attempting to use a small script to inject a browser history entry into the browsers history file.

The Chrome History

The chrome history files are actually simple to view and interact with in that they are SQLite databases and can be readily viewed with programs such as the SQLite Database Browser (<http://sqlitebrowser.sourceforge.net/>). In fact viewing the history files which such a program is highly recommended as you can use the browser to not only view the various tables and the data inside the tables, but also to view the schema that was used to construct the database. The Chrome user data file locations vary somewhat depending on what operating system you are running (Table 1).

Once you browse to the relevant directory for your operating system, the file you are looking for is called History (Note: depending on your installation the History file may be in a subdirectory of the destination directory

Table 1. Location of Chrome User Data Files by Operating System.

Operating System	History Location
Linux	/home/USER/.config/chromium
Windows XP	C:\Documents and Settings\USER\Local Settings\Application Data\Google\Chrome
Windows 7/Vista	C:\Users\USER\AppData\Local\Google\Chrome

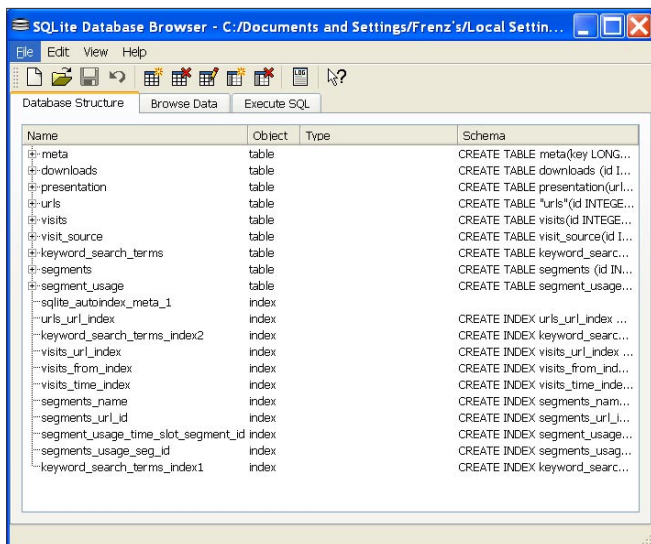


Figure 1. The History database schema

listed in Table 1). Opening the History file in the SQLite Database Browser reveals the schema for the database and, as such, reveals the tables that comprise the database (Figure 1). In particular, it is worth noting the existence of the tables named *urls* (Figure 2) and *visits* (Figure 3) as these are the table entries that must be edited in order to have an entry appear in the browsers history.

The *urls* table contains several fields of importance including an *id* that is uniquely assigned to each URL that is visited as well as a *url* field that list the actual URL (e.g. <http://www.google.com>) that was visited. This table also has field named *title*, which lists the title of the Web page that was visited and a *visit_count* field, which describes how many times a given URL was visited. The *typed_count* field tracks how many times the URL was actually types into the browsers URL bar (vs clicked on) and the *last_visit_time* field corresponds to the last time the URL was visited. It is important to note

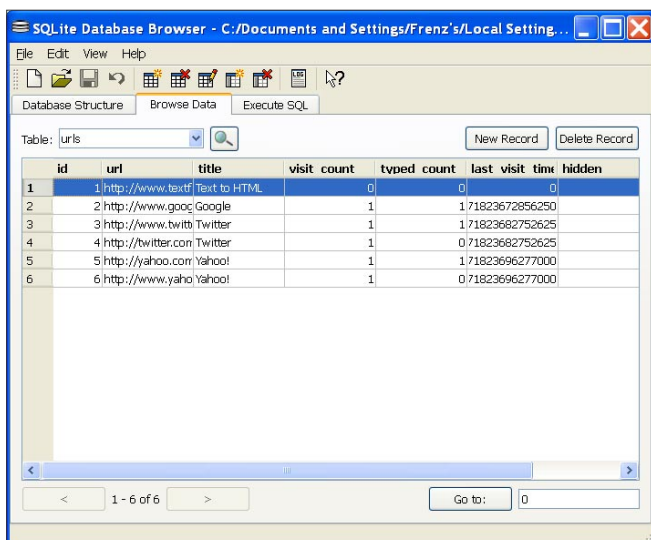


Figure 2. Sample data in the urls table of the History database

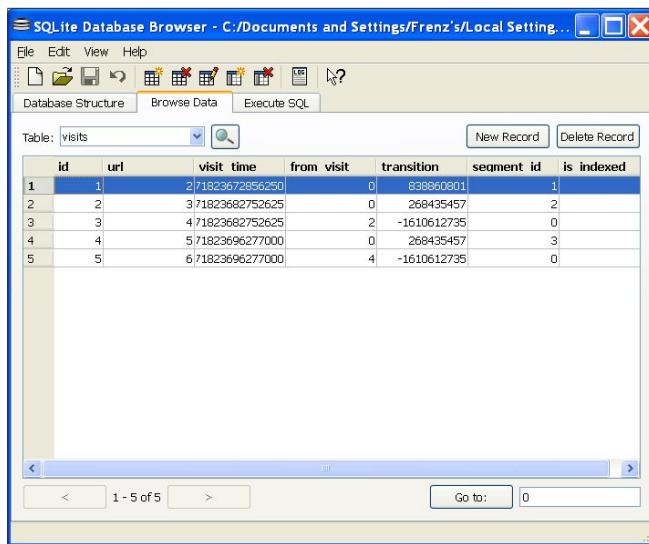


Figure 3. Sample data in the visits table of the History database

that Chrome timestamps are in microseconds since midnight of Jan 1st 1601 (UTC).

The *visits* table contains several fields of importance as well. As with the *urls* table each entry has a uniquely assigned *id* number, but this number is not the same as the one used in the *urls* table, as it corresponds to a particular visit to a URL and not a URL itself. The *url* field, however, does make use of the *id* numbers from the *url* table, and each *url* field entry should correspond to the *id* numbers of a URL in the *url* table. For example, a *url* value of 4 in the *visits* table would correspond to a visit to the URL with the *id* number of 4 in the *urls* table. The *visits* table also stores the time of each visit in the *visit_time* field and demonstrates which visit (if any) initiated the visit to a new URL via a link click, redirection, etc, in the *visit_from* field. The *transitions* field describes how the URL was loaded into the browser.

Injecting Site Visits via a Perl Script

Now that we have an understanding of the most essential data that needs to be modified, in order to create a chrome history entry, let's consider the construction of a simple Perl script that can inject an entry into the Chrome browser history. There are two things to note with this script – 1) it need not be Perl and

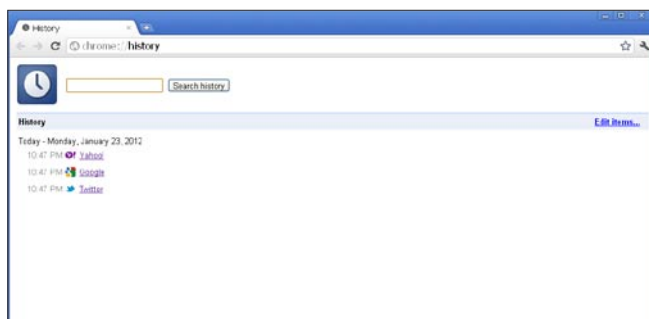


Figure 4. A sample Chrome History tab representing the before state

could be rewritten in any language that supports access to SQLite Databases and 2) The script is written with simplicity in mind in order to help readers understand the basic workings of the Chrome History database, and is not seeking to make the hardest to detect spoofed browser history possible.

This script makes use of the DBI Perl module which is a database interface for Perl and thus will allow the script to read data from and write data to the SQLite database files used to store Chrome's history. Note that when opening databases the script does not specify the full paths since this is somewhat operating system

specific. As such, the script can be modified to include the full path to the database on the system or can be run from inside the same directory that the databases are contained. The script begins by processing the History database file where a SELECT COUNT statement is used to find the next available *id* value in the *urls* table. The script then next defines the data that the user seeks to inject into the History file and proceeds to add an entry to the *urls* table via an INSERT statement. This process is then repeated for the *visits* table, but it is important to note that the *url* field value in the visits table should be set to the *id* value just inserted to *urls* table. Once the

Listing 1. The History Injection Script

```
#!/usr/bin/perl

# Copyright 2012- Christopher M. Frenz
# This script is free software it may be used, copied, redistributed, and/or modified
# under the terms laid forth in the Perl Artistic License

use DBI;

#opens the History database
my $dbh = DBI->connect("dbi:SQLite:dbname=History","","");

#finds the next available id
$id = $dbh->selectrow_array( "SELECT COUNT(*) FROM urls" );
$id=$id+1;

#data to inject
$url="http://www.microsoft.com";
$title="Microsoft.com";
$time=12971823657525436; #Jan 23, 2012 3:20 PM Eastern Standard Time

#modifies urls table
$dbh->do( "INSERT INTO urls(id, url, title, visit_count, typed_count, last_visit_time, hidden, favicon_id)
          VALUES ('$id', '$url', '$title', '1', '1', '$time', '0', '0')" );

#establishes url and id values for visits table
$url2=$id;
$id= $dbh->selectrow_array( "SELECT COUNT(*) FROM visits" );
$id=$id+1;

#modifies visits table
$dbh->do( "INSERT INTO visits(id, url, visit_time, from_visit, transition, segment_id, is_indexed) VALUES
          ('$id', '$url2', '$time', '0', '872415238', '0', '0')" );

#closes History DB
$dbh->disconnect;

unlink("Visited Links") or print "Unable to delete. Reason $!\n";
```

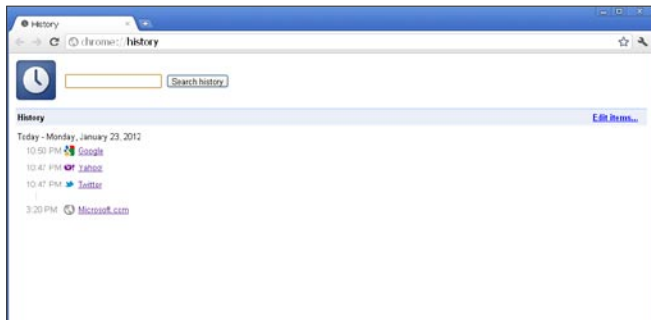



Figure 5. A sample Chrome History tab after the history injection script is executed

modification of the History database is complete, the unlink function is used to delete the file called *Visited Links*. This is an encrypted binary file that chrome uses to store visited links and to determine which links show up in the visited link color vs the non-visited link color. Due to the encrypted nature of the file, editing it would not be an easy undertaking, but luckily, deleting the file forces Chrome to regenerate the file from the History database. Thus by deleting the file, we are forcing Chrome to regenerate the file with our spoofed link and make the link appear in the proper visited link color.

Testing the Script

To test the script, a test environment can be set up by opening the Chrome browser and visiting a few pages to ensure the presence of a browser history. Now open the history tab in the Chrome browser and note the current lists of sites visited and times (Figure 4). Close the browser to release the database and allow the script to access it. Adjust the `$title`, `$url`, and `$time` variables in the script to reflect the data that you want to inject and run the script. Now reopen the browser, and examine the history tab again. Note the new entry (Figure 5).

Conclusions

The above script does raise some further questions about the validity of using link data from Chrome's browser history file. In addition to the commonly asked questions of how do we know if the suspect was the one using the computer at the time and not another person, forensic examiners may also need to consider the additional question of whether or not the data found in the browser history is legitimate or not. Perhaps such histories should have to be corroborated with ISP or server logs before they should be considered admissible evidence in the courtroom?

CHRISTOPHER M. FRENZ

Christopher M. Frenz is the author of the computer programming books "Pro Perl Parsing" and "Visual Basic and Visual Basic .NET for Scientists and Engineers". He actively performs research on a variety of computer related topics, including software security.



[GEEKED AT BIRTH]



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

[IT'S IN YOUR PULSE]

LEARN:

- Advancing Computer Science
- Network Security
- Artificial Life Programming
- Open Source Technologies
- Digital Media
- Robotics and Embedded Systems
- Digital Video
- Serious Games and Simulation
- Enterprise Software Development
- Strategic Technology Development
- Game Art and Animation
- Technology Forensics
- Game Design
- Technology Product Design
- Game Programming
- Technology Studies
- Human-Computer Interaction
- Virtual Modeling and Design
- Network Engineering
- Web and Social Media Technologies

www.uat.edu > 877.UAT.GEEK

Smart Meters

Dumb Regulators

Smart metering is the one of the hottest topics in the Energy and Utilities sector in Europe and North America.

What you will learn...

- Some of the myths about smart meters
- How regulators can make problems worse

What you should know...

- What „smart meters“ are
-

The historic situation with energy utility billing, predominantly electricity and gas, has been that the estimations of usage have generally been estimated based upon one or two reliable readings from customer meters per year. Consequently, “smart” meters hold a deal or promise. More accurate charging for energy used could be good for the consumer, allowing them to make more informed choices about when and how they choose to use electricity, for example. This is hugely important, as electricity, except that derived from nuclear power stations, costs different amounts to produce at different times of day. Smart meters could also allow energy companies to engage more intelligently with their customers, offering tariffs more appropriate to their lifestyle or business activities. Perhaps most importantly, it is seen to serve the *green agenda*.

But smart metering programmes around the world have been beset with issues. In part, this is because they are an example of new technology being used in new ways – with the resulting high level of risk. More importantly, from a security and privacy point of view, smart metering vastly increases the volume of data that utilities companies have on their customers – and also make it much, much more granular. All electrical devices have a *signature* – thus it may be possible to see, from a smart meter reading, not only what electrical devices a household has, but when it uses them, how often, when they are in (and when

they are out), even what films they watch (since TVs draw power based on what’s being shown on screen, even movies have a *signature*). One trade magazine even claimed it was possible to derive the sexual orientation of a household from this data – although they neglected to say how you could tell this from how many times they boiled the kettle, opened the fridge, or when they watched soap operas.

So smart metering presents governments and regulatory agencies with precisely the sort of challenge they are very badly equipped to deal with – a mixture of privacy and novel technology. As a consequence, some smart metering rollouts have faltered – for example in the Netherlands – while others have been the subject of significant consumer outcry – for example, PG & E’s rollout in the United States. Perhaps worse still, because the subject is so poorly understood by the public, there is room for all manner of cranks to state a case for smart metering being the beginning of the end of civilisation ...and I’m not really exaggerating on that last point.

So what are the real issues? Are they valid concerns?

One big concern is that smart meters will be a tool for *Big Brother*; in other words, the authorities will be keen to access the data so they can tell what you get up to in your own home. In fact, one of the poster slogans for the *anti-smart* campaign in the Netherlands was

Slim Meten Slinks Weten, which roughly translates as *Smart Meters know what you're doing*. This may be true to an extent; as we've seen, it's more than technically possible, but why would anyone want to know your domestic habits? I'm inclined to dismiss this; it's probably a big concern if you have a hydroponic marijuana farm, drawing huge amounts of electricity 24/7, for instance, but I don't believe that's going to worry most people.

Another commonly touted concern is an even simpler one – that criminals could, using the data, tell when you're likely to be at home, or out, or asleep. There are three logical problems with this hypothesis. First, there are plenty of other ways to guess this. Second, the data is only ever retrospective – if I can tell that you've been away for a week, how useful is this to me, as a burglar? What's to say you're not back home now, unless I'm checking the data in real time? Which gives us the third problem – why would I bother getting this data, when I could be more certain (and better able to break in) from looking at your property from the street outside? Even if I could compromise the records of, say 10,000,000 residential customers, what could I conceivably do with it? So, for me, this is another *non starter* as a concern.

Another *anti-smart* argument is that energy companies would not be rolling out smart meters if they were really in the consumers' interest. This is perhaps more of a convincing point. But increased accuracy of billing works both ways – the chances are that you, as the consumer, may work out paying more than you have at some times – but less at others. By the same token, the energy company won't be able to *estimate* bills in its own favour. So, in this case, as a consumer, you may lose out, or you may gain.

So, then, following this line of argument, where are the problems? Well, in some respects, I think they may come from governments and regulators themselves. Let me give you an example. In the UK, the Information Commissioner's Office, which regulates Data Protection, has defined personal information, in the context of smart metering as any information which, on its own or in combination with any other set or piece of information, could be used to identify an individual. Under EU law, personal data requires protection. But if you read that definition again, you'll quickly realise that, in fact, pretty much any data from smart metering fits it. This isn't very helpful from the energy companies' perspective – in fact, there is an argument that the fact that the regulator has failed to provide some leadership is retarding roll out (and serious consideration of many security controls).

Some commentators have also raised the spectre of *remote mass disconnect*, whereby somebody, be they

master criminals, or an unfriendly government, decide they are going to *switch off* bits, or all, of a country's utilities networks. Now, many national smart metering programmes have deliberately avoided building in remote disconnect functionality, precisely because it is so legally contentious. For example, how many electricity companies would want a law suit and the resultant publicity because they had terminated supply to a vulnerable customer, perhaps somebody with a kidney dialysis machine?

But this, I think, help illustrates one of the most important risk areas – smart metering facilitates other technologies. Organisations are springing up that offer a range of complementary technologies – one example is *smart plugs*. These are devices which connect your electrical outlets with your devices, and are controllable remotely – e.g. from your mobile phone, work PC, or tablet. So, while a smart metering infrastructure may not allow the Doomsday scenario of, say, China turning off the power in the US, it might, just, create space for another technology that could allow it. And of course, because the *smart plug* is collecting the same sort of personal data, it just won't have the same level of regulatory focus on it.

To coin an old phrase, the road to hell is paved with good intentions. I think there is a convincing argument that smart meters, in and of themselves, do not constitute a significant risk to life and liberty for the individual. But I do think that it is a great exemplar of sclerotic government actions actually creating a problem; by failing to either manage public debate effectively, or provide useful guidance, government agencies and regulators have created their very own, and new, *problem space*. So don't worry to much about smart meters – worry more about the things that come with them.

DRAKE

Drake has worked on information security and strategy with government agencies, the military, financial institutions and other blue chip organisations in Europe, the Middle East, and Africa since Boris Yeltsin was President.

iptables

iptables is a user space application program that allows a system administrator to configure the tables provided by the Linux kernel firewall (implemented as different Netfilter modules) and the chains and rules it stores. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.

What you will learn...

- What iptables are

What you should know...

- Linux kernel firewall

iptables requires elevated privileges to operate and must be executed by user *root*, otherwise it fails to function. On most Linux systems, iptables is installed as `/usr/sbin/iptables` and documented in its *man page*, [2] which can be opened using `man iptables` when installed. It may also be found in `/sbin/iptables`, but since iptables is more like a service rather than an *essential binary*, the preferred location remains `/usr/sbin`.

iptables is also commonly used to inclusively refer to the kernel-level components. *x_tables* is the name of the kernel module carrying the shared code portion used by all four modules that also provides the API used for extensions; subsequently, *Xtables* is more or less used to refer to the entire firewall (v4, v6, arp, eb) architecture.

Introduction

Network security is a primary consideration in any decision to host a website as the threats are becoming more widespread and persistent every day. One means of providing additional protection is to invest in a firewall. Though prices are always falling, in some cases you may be able to create a comparable unit using the Linux iptables package on an existing server for little or no additional expenditure. This chapter shows how to convert a Linux server into:

- A firewall while simultaneously being your home website's mail, web and DNS server.

- A router that will use NAT and port forwarding to both protect your home network and have another web server on your home network while sharing the public IP address of your firewall.

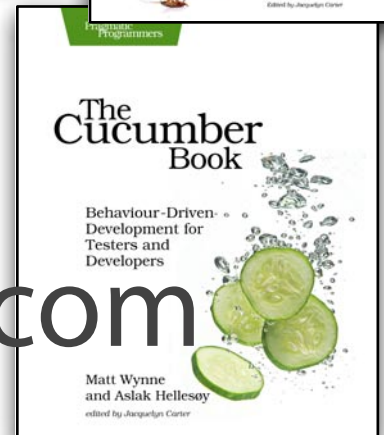
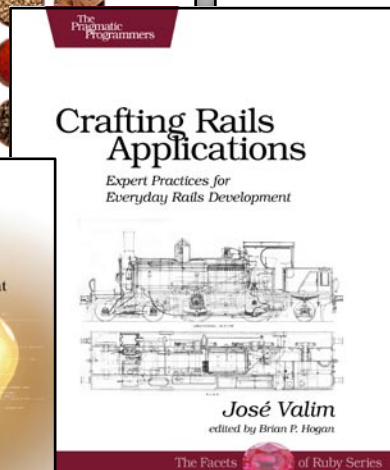
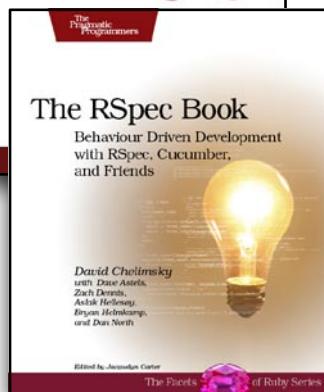
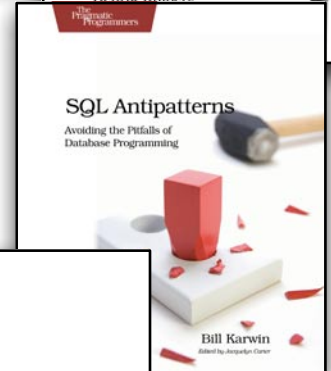
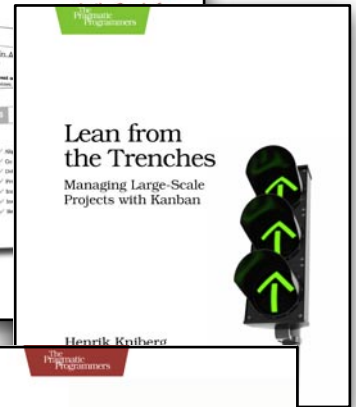
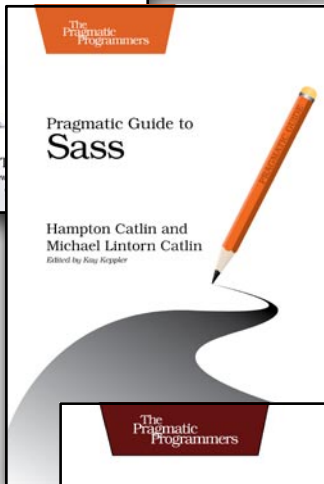
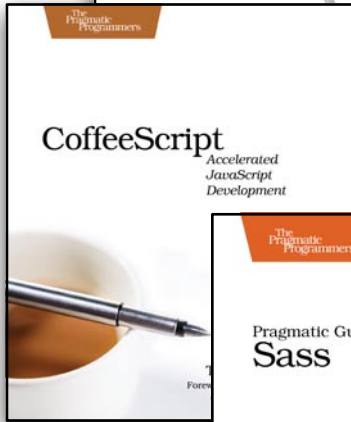
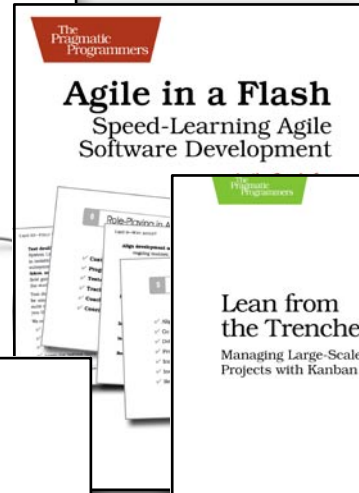
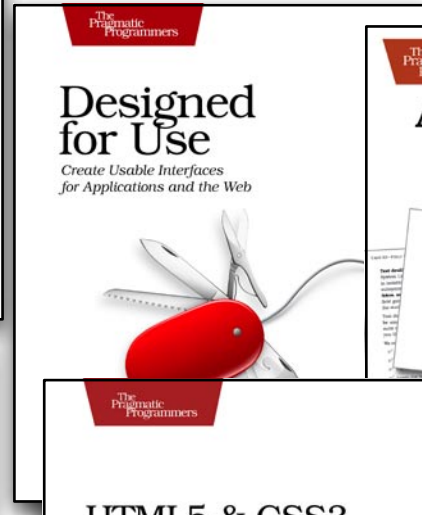
Creating an iptables firewall script requires many steps, but with the aid of the sample tutorials, you should be able to complete a configuration relatively quickly.

What Is iptables?

Originally, the most popular firewall/NAT package running on Linux was ipchains, but it had a number of shortcomings. To rectify this, the Netfilter organization decided to create a new product called iptables, giving it such improvements as:

- Better integration with the Linux kernel with the capability of loading iptables-specific kernel modules designed for improved speed and reliability.
- Stateful packet inspection. This means that the firewall keeps track of each connection passing through it and in certain cases will view the contents of data flows in an attempt to anticipate the next action of certain protocols. This is an important feature in the support of active FTP and DNS, as well as many other network services.
- Filtering packets based on a MAC address and the values of the flags in the TCP header. This is helpful

Keeping You At the
Top of Your Game.
Paper • eBooks • Dropbox



Pragmatic
Bookshelf

www.pragprog.com

in preventing attacks using malformed packets and in restricting access from locally attached servers to other networks in spite of their IP addresses.

- System logging that provides the option of adjusting the level of detail of the reporting.
- Better network address translation.
- Support for transparent integration with such Web proxy programs as Squid.
- A rate limiting feature that helps iptables block some types of *denial of service* (DoS) attacks.

Considered a faster and more secure alternative to ipchains, iptables has become the default firewall package installed under RedHat and Fedora Linux.

Download And Install The Iptables Package

Before you begin, you need to make sure that the iptables software RPM is installed. (See Chapter 6, *Installing Linux Software*, if you need a refresher.) When searching for the RPMs, remember that the filename usually starts with the software package name by a version number, as in `iptables-1.2.9-1.0.i386.rpm`.

How To Start iptables

You can start, stop, and restart iptables after booting by using the commands:

```
[root@bigboy tmp]# service iptables start
[root@bigboy tmp]# service iptables stop
[root@bigboy tmp]# service iptables restart
```

To get iptables configured to start at boot, use the `chkconfig` command:

```
[root@bigboy tmp]# chkconfig iptables on
```

Determining The Status of iptables

You can determine whether iptables is running or not via the service `iptables status` command. Fedora Core will give a simple status message. For example

```
[root@bigboy tmp]# service iptables status
Firewall is stopped.
[root@bigboy tmp]#
```

Packet Processing In iptables

All packets inspected by iptables pass through a sequence of built-in tables (queues) for processing. Each of these queues is dedicated to a particular type of packet activity and is controlled by an associated packet transformation/filtering chain.

There are three tables in total. The first is the mangle table which is responsible for the alteration of quality of service bits in the TCP header. This is hardly used in a home or SOHO environment.

Listing 1. Processing For Packets Routed By The Firewall

```
Queue
Type
Queue
Function
Packet
Transformation
Chain in Queue
Chain Function
Filter Packet
filtering FORWARD
Filters packets to servers accessible by another NIC
on the firewall.
INPUT
Filters packets destined to the firewall.
OUTPUT
Filters packets originating from the firewall
Nat Network
Address
Translation
PREROUTING
Address translation occurs before routing.
Facilitates
the transformation of the destination IP address to
be
compatible with the firewall's routing table. Used
with
NAT of the destination IP address, also known
as destination NAT or DNAT.
POSTROUTING
Address translation occurs after routing. This
implies
that there was no need to modify the destination IP
address of the packet as in pre-routing. Used with
NAT of the source IP address using either one-to-one
or many-to-one NAT. This is known as source NAT,
or SNAT.
OUTPUT
Network address translation for packets generated by
the firewall. (Rarely used in SOHO environments)
Mangle TCP header
modification PREROUTING
POSTROUTING
OUTPUT
INPUT
FORWARD
Modification of the TCP packet quality of service
bits
before routing occurs. (Rarely used in SOHO
environments)
```

The second table is the filter queue which is responsible for packet filtering. It has three built-in chains in which you can place your firewall policy rules. These are the:

- Forward chain: Filters packets to servers protected by the firewall.
- Input chain: Filters packets destined for the firewall.
- Output chain: Filters packets originating from the firewall.

The third table is the nat queue which is responsible for network address translation. It has two built-in chains; these are:

- Pre-routing chain: NATs packets when the destination address of the packet needs to be changed.

- Post-routing chain: NATs packets when the source address of the packet needs to be changed (Listing 1).

You need to specify the table and the chain for each firewall rule you create. There is an exception: Most rules are related to filtering, so iptables assumes that any chain that's defined without an associated table will be a part of the filter table. The filter table is therefore the default.

To help make this clearer, take a look at the way packets are handled by iptables. TCP packet from the Internet arrives at the firewall's interface on Network A to create a data connection.

The packet is first examined by your rules in the mangle table's PREROUTING chain, if any. It is then inspected by the rules in the nat table's PREROUTING

Listing 2. Descriptions Of The Most Commonly Used Targets

target	Description	Most Common	Options
ACCEPT	<code>_ iptables</code> stops further processing. <code>_</code> The packet is handed over to the end application or the operating system for processing		
N/A			
DROP	<code>_ iptables</code> stops further processing. <code>_</code> The packet is blocked		
N/A			
LOG	<code>_</code> The packet information is sent to the <code>syslog</code> daemon for logging <code>_ iptables</code> continues processing with the next rule in the table <code>_</code> As you can't log and drop at the same time, it is common to have two similar rules in sequence. The first will log the packet, the second will drop it. <code>--log-prefix "string"</code> Tells iptables to prefix all log messages with a user defined string. Frequently used to tell why the logged packet was dropped		
REJECT	<code>_</code> Works like the DROP target, but will also <code>return</code> an error message to the host sending the packet that the packet was blocked <code>--reject-with</code> qualifier The qualifier tells what type of reject message is returned. Qualifiers include: <code>icmp-port-unreachable</code> (default) <code>icmp-net-unreachable</code>		
			<code>icmp-host-unreachable</code> <code>icmp-proto-unreachable</code> <code>icmp-net-prohibited</code> <code>icmp-host-prohibited</code> <code>tcp-reset</code> <code>echo-reply</code> DNAT <code>_</code> Used to do destination network address translation. ie. rewriting the destination IP address of the packet <code>--to-destination ipaddress</code> Tells iptables what the destination IP address should be SNAT <code>_</code> Used to do source network address translation rewriting the source IP address of the packet <code>_</code> The source IP address is user defined <code>--to-source <address>[-<address>][:<port>-<port>]</code> Specifies the source IP address and ports
			MASQUERADE <code>_</code> Used to do Source Network Address Translation. <code>_</code> By default the source IP address is the same as that used by the firewall's interface <code>[--to-ports <port>[-<port>]]</code> Specifies the range of source ports to which the original source port can be mapped.

chain to see whether the packet requires DNAT. It is then routed.

If the packet is destined for a protected network, then it is filtered by the rules in the FORWARD chain of the filter table and, if necessary, the packet undergoes SNAT in the POSTROUTING chain before arriving at Network B. When the destination server decides to reply, the packet undergoes the same sequence of steps. Both the FORWARD and POSTROUTING chains may be configured to implement *quality of service* (QoS) features in their mangle tables, but this is not usually done in SOHO environments.

If the packet is destined for the firewall itself, then it passes through the mangle table of the INPUT chain, if configured, before being filtered by the rules in the

INPUT chain of the filter table before. If it successfully passes these tests then it is processed by the intended application on the firewall.

At some point, the firewall needs to reply. This reply is routed and inspected by the rules in the OUTPUT chain of the mangle table, if any. Next, the rules in the OUTPUT chain of the nat table determine whether DNAT is required and the rules in the OUTPUT chain of the filter table are then inspected to help restrict

Listing 3. General Iptables Match Criteria

```
iptables command
Switch
Description
-t <table> If you don't specify a table, then
           the filter table is assumed. As
           discussed
before, the possible built-in tables include: filter,
           nat, mangle
-j <target> Jump to the specified target chain when
           the packet matches the current
           rule.
-A Append rule to end of a chain
-F Flush. Deletes all the rules in the selected
           table
-p <protocoltype>
Match protocol. Types include, icmp, tcp, udp, and
           all
-s <ip-address> Match source IP address
-d <ip-address> Match destination IP address
-i <interfacename>
Match "input" interface on which the packet enters.
-o <interfacename>
Match "output" interface on which the packet exits
In this command switches example
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p
           TCP -j ACCEPT
iptables is being configured to allow the firewall to
           accept TCP packets coming in on
           interface eth0 from
any IP address destined for the firewall's IP
           address of 192.168.1.1. The 0/0
           representation of an IP
address means any.
```

Listing 4. Common TCP and UDP Match Criteria

```
Switch Description
-p tcp --sport
<port>
TCP source port. Can be a single value or a range in
           the format: start-portnumber:
           end-port-number
-p tcp --dport
<port>
TCP destination port. Can be a single value
           or a range in the format:
           startingport:
           ending-port
-p tcp --syn Used to identify a new TCP connection
           request. ! --syn means, not a
           new
           connection request
-p udp --sport UDP source port. Can be a single
           value or a range in the format:
           starting<
           port> port:ending-port
<port>
UDP destination port. Can be a single value
           or a range in the format:
           startingport:
           ending-port
In this example:
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58
           -o eth1 -p TCP \
--sport 1024:65535 --dport 80 -j ACCEPT
iptables is being configured to allow the firewall to
           accept TCP packets for routing
           when they enter on
           interface eth0 from
           any IP address and are
           destined for an IP address of
           192.168.1.58 that is reachable
           via interface eth1. The source port is in the
           range 1024 to 65535 and the
           destination port is port 80
(www/http).
```


unauthorized packets. Finally, before the packet is sent back to the Internet, SNAT and QoS mangling is done by the POSTROUTING chain.

It is now time to discuss the ways in which you add rules to these chains.

Targets And Jumps

Each firewall rule inspects each IP packet and then tries to identify it as the target of some sort of operation. Once a target is identified, the packet needs to jump

over to it for further processing. Listing 2 lists the built-in targets that iptables uses.

Important Iptables Command Switch Operations

Each line of an iptables script not only has a jump, but they also have a number of command line options

Table 5. Common ICMP (Ping) Match Criteria

```
Matches used with ---icmp-type Description
--icmp-type <type> The most commonly used types are
                    echo-reply and echo-request
In this example:
iptables -A OUTPUT -p icmp --icmp-type echo-request
                    -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j
                    ACCEPT
iptables is being configured to allow the firewall to
                    send ICMP echo-requests (pings)
                    and in turn, accept
                    the expected ICMP echo-replies.
Consider another example
iptables -A INPUT -p icmp --icmp-type echo-request \
-m limit --limit 1/s -i eth0 -j ACCEPT
The limit feature in iptables specifies the maximum
                    average number of matches to
                    allow per second. You
                    can specify time intervals in the format /second,
                    /minute, /hour, or /day, or you
                    can use abbreviations so
                    that 3/second is the same as 3/s.
In this example, ICMP echo requests are restricted
                    to no more than one per second.
                    When tuned
                    correctly, this feature allows you to filter
                    unusually high volumes of traffic
                    that characterize denial of
                    service (DOS) attacks and Internet worms.
iptables -A INPUT -p tcp --syn -m limit --limit 5/s
                    -i eth0 -j ACCEPT
You can expand on the limit feature of iptables to
                    reduce your vulnerability to
                    certain types of denial of
                    service attack. Here a defense for SYN flood attacks
                    was created by limiting the
                    acceptance of TCP
                    segments with the SYN bit set to no more than five
                    per second.
```

Listing 6. Common Extended Match Criteria

```
Switch Description
-m multiport --
sports <port, port>
A variety of TCP/UDP source ports separated by
                    commas. Unlike when -
                    m isn't used, they do not have to be within a range.
-m multiport --
dports <port, port>
A variety of TCP/UDP destination ports separated by
                    commas. Unlike
                    when -m isn't used, they do not have to be within a
                    range.
-m multiport --
ports <port, port>
A variety of TCP/UDP ports separated by commas.
                    Source and destination
                    ports are assumed to be the same and they do not
                    have to be within a
                    range.
-m --state <state>
The most frequently tested states are:
ESTABLISHED: The packet is part of a connection that
                    has seen packets
                    in both directions
NEW: The packet is the start of a new connection
RELATED: The packet is starting a new secondary
                    connection. This is a
                    common feature of such protocols such as an FTP data
                    transfer, or an
                    ICMP error.
INVALID: The packet couldn't be identified. Could be
                    due to insufficient
                    system resources, or ICMP errors that don't match an
                    existing data flow.
This is an expansion on the previous example:
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58
                    -o eth1 -p TCP \
--sport 1024:65535 -m multiport --dports 80,443 -j
                    ACCEPT
iptables -A FORWARD -d 0/0 -o eth0 -s 192.168.1.58
                    -i eth1 -p TCP \
-m state --state ESTABLISHED -j ACCEPT
```

that are used to append rules to chains that match your defined packet characteristics, such the source IP address and TCP port. There are also options that can be used to just clear a chain so you can start all over again. Listing 3 through 6 list the most common options.

Here iptables is being configured to allow the firewall to accept TCP packets to be routed when they enter on interface eth0 from any IP address destined for IP address of 192.168.1.58 that is reachable via interface eth1. The source port is in the range 1024 to 65535 and the destination ports are port 80 (www/http) and 443 (https). The return packets from 192.168.1.58 are allowed to be accepted too. Instead of stating the source and destination ports, you can simply allow packets related to established connections using the -m state and --state ESTABLISHED options.

Using User Defined Chains

As you may remember, you can configure iptables to have user-defined chains. This feature is frequently used to help streamline the processing of packets. For example, instead of using a single, built-in chain for all protocols, you can use the chain to determine the protocol type for the packet and then hand off the actual final processing to a user-defined, protocol-specific chain in the filter table. In other words, you can replace a long chain with a stubby main chain pointing to multiple stubby chains, thereby shortening the total length of all chains the packet has to pass through. For example:

```
iptables -A INPUT -i eth0 -d 206.229.110.2 -j fast-  
input-queue  
iptables -A OUTPUT -o eth0 -s 206.229.110.2 -j fast-  
output-queue  
iptables -A fast-input-queue -p icmp -j icmp-queue-in  
iptables -A fast-output-queue -p icmp -j icmp-queue-out  
iptables -A icmp-queue-out -p icmp --icmp-type echo-request \  
-m state --state NEW -j ACCEPT  
iptables -A icmp-queue-in -p icmp --icmp-type echo-reply  
-j ACCEPT
```

Here six queues help assist in improving processing speed. Listing 7 summarizes the function of each.

Saving Your iptables Scripts

The service iptables save command permanently saves the iptables configuration in the /etc/sysconfig/iptables file. When the system reboots, the iptables-restore program reads the configuration and makes it the active configuration.

The format of the /etc/sysconfig/iptables file is slightly different from that of the scripts shown in this chapter. The initialization of built-in chains is automatic and

the string *iptables* is omitted from the rule statements. Here is a sample /etc/sysconfig/iptables configuration that allows ICMP, IPsec (ESP and AH packets), already established connections, and inbound SSH (Listing 8).

It is never a good idea to edit this script directly because it is always overwritten by the save command

Listing 7. Custom Queues Example Listing

```
Chain Description  
INPUT The regular built-in INPUT chain in iptables  
OUTPUT The regular built-in OUTPUT chain in iptables  
fast-inputqueue  
Input chain dedicated to identifying specific  
protocols and shunting the  
packets to  
protocol specific chains.  
fast-outputqueue  
Output chain dedicated to identifying specific  
protocols and shunting the  
packets  
to protocol specific chains.  
icmp-queue-out Output queue dedicated to ICMP  
icmp-queue-in Input queue dedicated to ICMP
```

Listing 8. Inbound SSH

```
[root@bigboy tmp]# cat /etc/sysconfig/iptables  
# Generated by iptables-save v1.2.9 on Mon Nov 8 11:  
00:07 2004  
  
*filter  
:INPUT ACCEPT [0:0]  
:FORWARD ACCEPT [0:0]  
:OUTPUT ACCEPT [144:12748]  
:RH-Firewall-1-INPUT - [0:0]  
-A INPUT -j RH-Firewall-1-INPUT  
-A FORWARD -j RH-Firewall-1-INPUT  
-A RH-Firewall-1-INPUT -i lo -j ACCEPT  
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type  
255 -j ACCEPT  
-A RH-Firewall-1-INPUT -p esp -j ACCEPT  
-A RH-Firewall-1-INPUT -p ah -j ACCEPT  
-A RH-Firewall-1-INPUT -m state --state  
RELATED,ESTABLISHED -j ACCEPT  
  
ACCEPT  
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-  
host-prohibited  
  
COMMIT  
# Completed on Mon Nov 8 11:00:07 2004  
[root@bigboy tmp]#
```

and it doesn't save any comments at all, which can also make it extremely difficult to follow. For these reasons, you're better off writing and applying a customized script and then using the service `iptables save` command to make the changes permanent.

Fedora's iptables Rule Generator

Fedora comes with a program called `lokkit` that you can use to generate a very rudimentary firewall rule set. It

Listing 9. `iptables-save` output to a text file named `firewall-config`

```
[root@bigboy tmp]# iptables-save > firewall-config
[root@bigboy tmp]# cat firewall-config
# Generated by iptables-save v1.2.9 on Mon Nov 8 11:
00:07 2004

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type
255 -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state
RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW
-m tcp --dport 22 -j
ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-
host-prohibited

COMMIT
# Completed on Mon Nov 8 11:00:07 2004
[root@bigboy tmp]#
```

Listing 10. Place statements in the `/etc/rc.local` file

```
# File: /etc/rc.local
# Module to track the state of connections
modprobe ip_conntrack
# Load the iptables active FTP module, requires
ip_conntrack
modprobe ip_conntrack_ftp
# Load iptables NAT module when required
modprobe iptable_nat
# Module required for active an FTP server using NAT
modprobe ip_nat_ftp
```

prompts for the level of security and then gives you the option of doing simple customizations. It is a good place for beginners to start on a test system so that they can see a general rule structure.

Like the service `iptables save` command, `lokkit` saves the firewall rules in a new `/etc/sysconfig/iptables` file for use on the next reboot.

Once you have become familiar with the `iptables` syntax, it's best to write scripts that you can comment and then save it to `/etc/sysconfig/iptables`. It makes them much more manageable and readable.

Recovering From A Lost Script

Sometimes the script you created to generate `iptables` rules may get corrupted or lost, or you might inherit a new system from an administrator and cannot find the original script used to protect it. In these situations, you can use the `iptables-save` and `iptables-restore` commands to assist you with the continued management of the server.

Unlike the service `iptables save` command, which actually saves a permanent copy of the firewall's active configuration in the `/etc/sysconfig/iptables` file, `iptables-save` displays the active configuration to the screen in `/etc/sysconfig/iptables` format. If you redirect the `iptables-save` screen output to a file with the `>` symbol, then you can edit the output and reload the updated rules when they meet your new criteria with the `iptables-restore` command.

This example exports the `iptables-save` output to a text file named `firewall-config` (Listing 9).

After editing the `firewall-config` file with the commands you need, you can reload it into the active firewall rule set with the `iptables-restore` command.

```
[root@bigboy tmp]# iptables-restore < firewall-config
```

Finally, you should permanently save the active configuration so that it will be loaded automatically when the system reboots:

```
[root@bigboy tmp]# service iptables save
```

If desired, you can eventually convert this `firewall-config` file into a regular `iptables` script so that it becomes more easily recognizable and manageable.

Loading Kernel Modules Needed By iptables

The `iptables` application requires you to load certain kernel modules to activate some of its functions. Whenever any type of NAT is required, the `iptable_nat` module needs to be loaded. The `ip_conntrack_ftp` module needs to be added for FTP support and should always be loaded with the `ip_conntrack` module which tracks TCP connection states. As most scripts probably

will keep track of connection states, the `ip_conntrack` module will be needed in any case. The `ip_nat_ftp` module also needs to be loaded for FTP servers behind a NAT firewall.

Unfortunately, the `/etc/sysconfig/iptables` file doesn't support the loading of modules, so you'll have to add the statements to your `/etc/rc.local` file which is run at the end of every reboot.

The script samples in this chapter include these statements only as a reminder to place them in the `/etc/rc.local` file (Listing 10).

Sample iptables Scripts

This section provides some sample scripts you can use to get iptables working for you. Pay special attention to the logging example at the end.

The basic initialization script snippet should also be included in all your scripts to ensure the correct initialization of your chains should you decide to restart your script after startup. This chapter also includes other snippets that will help you get basic functionality. It should be a good guide to get you started.

Note: Once you feel more confident, you can use *Appendix II "Codes, Scripts, and Configurations"*, to find detailed scripts. The appendix shows you how to allow your firewall to:

- Be used as a Linux Web, mail and DNS server
- Be the NAT router for your home network
- Prevent various types of attacks using corrupted TCP, UDP and ICMP packets.
- Provide outbound passive FTP access from the firewall

There are also simpler code snippets in the *Appendix II "Codes, Scripts, and Configurations"*, for Inbound and outbound FTP connections to and from your firewall.

Basic Operating System Defense

You can do several things before employing your firewall script to improve the resilience of your firewall to attack. For example, the Linux operating system has a number of built-in protection mechanisms that you should activate by modifying the system kernel parameters in the `/proc` filesystem via the `/etc/sysctl.conf` file. Using of `/etc/sysctl.conf` to modify kernel parameters is explained in more detail in, *Appendix I "Miscellaneous Linux Topics"*. Here is a sample configuration (Listing 11).

Advanced iptables Initialization

You may also want to add some more advanced initialization steps to your script, including checks for Internet traffic from RFC1918 private addresses. The sample script snippet below outlines how to do this.

More complex initializations would include checks for attacks using invalid TCP flags and directed broadcasts which are beyond the scope of this book.

The script also uses multiple user-defined chains to make the script shorter and faster as the chains can be repeatedly accessed. This removes the need to repeat the same statements over and over again. You can take even more precautions to further protect your network. The complete firewall script in *Appendix II "Codes, Scripts, and Configurations"*, outlines most of them (Listing 12).

Listing 11. Sample configuration

```
# File: /etc/sysctl.conf
#-----
# Disable routing triangulation. Respond to queries out
# the same interface, not another. Helps to maintain
# state
# Also protects against IP spoofing
#-----
net/ipv4/conf/all/rp_filter = 1

# Enable logging of packets with malformed IP addresses
#-----
net/ipv4/conf/all/log_martians = 1
#-----
# Disable redirects
#-----
net/ipv4/conf/all/send_redirects = 0
#-----
# Disable source routed packets
#-----
net/ipv4/conf/all/accept_source_route = 0
#-----
# Disable acceptance of ICMP redirects
#-----
net/ipv4/conf/all/accept_redirects = 0
#-----
# Turn on protection from Denial of Service (DOS) attacks
#-----
net/ipv4/tcp_syncookies = 1
#-----
# Disable responding to ping broadcasts
#-----
net/ipv4/icmp_echo_ignore_broadcasts = 1
#-----
# Enable IP routing. Required if your firewall is
# protecting a
# network, NAT included
#-----
net/ipv4/ip_forward = 1
```


isn't necessary to specify these ports for the return leg as outbound packets for all established connections are allowed. Connections initiated by persons logged into the Web server will be denied as outbound NEW connection packets aren't allowed (Listing 14).

Allowing Your Firewall To Access The Internet

This iptables script enables a user on the firewall to use a Web browser to surf the Internet. HTTP traffic uses TCP port 80, and HTTPS uses port 443.

Note

HTTPS (secure HTTP) is used for credit card transactions frequently, as well as by RedHat Linux servers running up2date. FTP and HTTP are frequently used with yum (Listing 15).

If you want all TCP traffic originating from the firewall to be accepted, then remove the line:

```
-m multiport --dports 80,443 --sport 1024:65535
```

Listing 15. FTP and HTTP are frequently used with yum

```
#-----  
# Allow port 80 (www) and 443 (https) connections from  
# the firewall  
#-----  
iptables -A OUTPUT -j ACCEPT -m state \  
--state NEW,ESTABLISHED,RELATED -o eth0 -p tcp \  
-m multiport --dports 80,443 --sport 1024:65535  
#-----  
# Allow previously established connections  
# - Interface eth0 is the internet interface  
#-----  
iptables -A INPUT -j ACCEPT -m state --state  
ESTABLISHED,RELATED \  
-i eth0 -p tcp
```

Listing 16. Allow only specific ports to have access to your firewall

```
#-----  
# Allow all bidirectional traffic from your firewall to  
# the  
# protected network  
# - Interface eth1 is the private network interface  
#-----  
iptables -A INPUT -j ACCEPT -p all -s 192.168.1.0/24  
-i eth1  
iptables -A OUTPUT -j ACCEPT -p all -d 192.168.1.0/24  
-o eth1
```

Listing 17. Protect the home network

```
#-----  
# Load the NAT module  
#  
# Note: It is best to use the /etc/rc.local example in  
# this  
# chapter. This value will not be retained in the  
# /etc/sysconfig/iptables file. Included only as a  
# reminder.  
#-----  
modprobe iptable_nat
```

```
#-----  
# Enable routing by modifying the ip_forward /proc  
# filesystem file  
#  
# Note: It is best to use the /etc/sysctl.conf example  
# in this  
# chapter. This value will not be retained in the  
# /etc/sysconfig/iptables file. Included only as a  
# reminder.  
#-----  
echo 1 > /proc/sys/net/ipv4/ip_forward  
#-----  
# Allow masquerading  
# - Interface eth0 is the internet interface  
# - Interface eth1 is the private network interface  
#-----  
iptables -A POSTROUTING -t nat -o eth0 -s 192.168.1.0/  
24 -d 0/0 \  
-j MASQUERADE  
#-----  
# Prior to masquerading, the packets are routed via  
# the filter  
# Allowed outbound: New, established and related  
# connections  
# Allowed inbound : Established and related  
# connections  
#-----  
iptables -A FORWARD -t filter -o eth0 -m state \  
--state NEW,ESTABLISHED,RELATED -j ACCEPT  
iptables -A FORWARD -t filter -i eth0 -m state \  
--state ESTABLISHED,RELATED -j ACCEPT
```

Allow Your Home Network To Access The Firewall

In this example, `eth1` is directly connected to a home network using IP addresses from the 192.168.1.0 network. All traffic between this network and the firewall is simplistically assumed to be trusted and allowed.

Further rules will be needed for the interface connected to the Internet to allow only specific ports, types of connections and possibly even remote servers to have access to your firewall and home network (Listing 16).

Masquerading (Many to One NAT)

As explained in *Chapter 2, "Introduction to Networking"*, masquerading is another name for what many call many to one NAT. In other words, traffic from all devices on one or more protected networks will appear as if it originated from a single IP address on the Internet side of the firewall.

Note

The masquerade IP address always defaults to the IP address of the firewall's main interface. The advantage of this is that you never have to specify the NAT IP address. This makes it much easier to configure iptables NAT with DHCP.

You can configure many to one NAT to an IP alias, using the POSTROUTING and not the MASQUERADE statement. An example of this can be seen in the static NAT section that follows.

Keep in mind that iptables requires the `iptables_nat` module to be loaded with the `modprobe` command for the masquerade feature to work. Masquerading also depends on the Linux operating system being configured to support routing between the internet and private network interfaces of the firewall. This is done by enabling IP forwarding or routing by giving the file `/proc/sys/net/ipv4/ip_forward` the value 1 as opposed to the default disabled value of 0.

Once masquerading has been achieved using the POSTROUTING chain of the nat table, you will have to configure iptables to allow packets to flow between the two interfaces. To do this, use the FORWARD chain of the filter table. More specifically, packets related to NEW and ESTABLISHED connections will be allowed outbound to the Internet, but only packets related to ESTABLISHED connections will be allowed inbound. This helps to protect the home network from anyone trying to initiate connections from the Internet (Listing 17).

Note

If you configure your firewall to do masquerading, then it should be used as the default gateway for all your servers on the network.

Port Forwarding Type NAT (DHCP DSL)

In many cases home users may get a single DHCP public IP address from their ISPs. If a Linux firewall is also your interface to the Internet and you want to host a Web site on one of the NAT protected home servers, then you will have to use port forwarding. Here the combination of the firewall's single IP address, the remote server's IP address, and the source/destination port of the traffic can be used to uniquely identify a traffic flow. All traffic that matches a particular combination of these factors may then be forwarded to a single server on the private network.

Port forwarding is handled by the PREROUTING chain of the nat table. As in masquerading, the `iptables_nat` module has to be loaded and routing has to be enabled for port forwarding to work. Routing too must be allowed in iptables with the FORWARD chain, this includes all NEW inbound connections from the Internet matching the port forwarding port plus all future packets related to the ESTABLISHED connection in both directions (Listing 18).

Static NAT

In this example, all traffic to a particular public IP address, not just to a particular port, is translated to a single server on the protected subnet. Because the firewall has more than one IP address, I can't recommend MASQUERADE; it will force masquerading as the IP address of the primary interface and not as any of the alias IP addresses the firewall may have. Instead, use SNAT to specify the alias IP address to be used for connections initiated by all other servers in the protected network.

Note

Although the nat table NATs all traffic to the target servers (192.168.1.100 to 102), only connections on ports 80,443 and 22 are allowed through by the FORWARD chain. Also notice how you have to specify a separate `-m multiport` option whenever you need to match multiple non-sequential ports for both source and destination. In this example the firewall:

- Uses one to one NAT to make the server 192.168.1.100 on your home network appear on the Internet as IP addresses 97.158.253.26.
- Creates a many to one NAT for the 192.168.1.0 home network in which all the servers appear on the Internet as IP address 97.158.253.29. This is different from masquerading. You will have to create alias IP addresses for each of these Internet IPs for one to one NAT to work (Listing 19).

Troubleshooting iptables

A number of tools are at your disposal for troubleshooting iptables firewall scripts. One of the best methods is to log all dropped packets to the `/var/log/messages` file.

Checking The Firewall Logs

You track packets passing through the iptables list of rules using the LOG target. You should be aware that the LOG target:

- Logs all traffic that matches the iptables rule in which it is located.
- Automatically writes an entry to the `/var/log/messages` file and then executes the next rule.

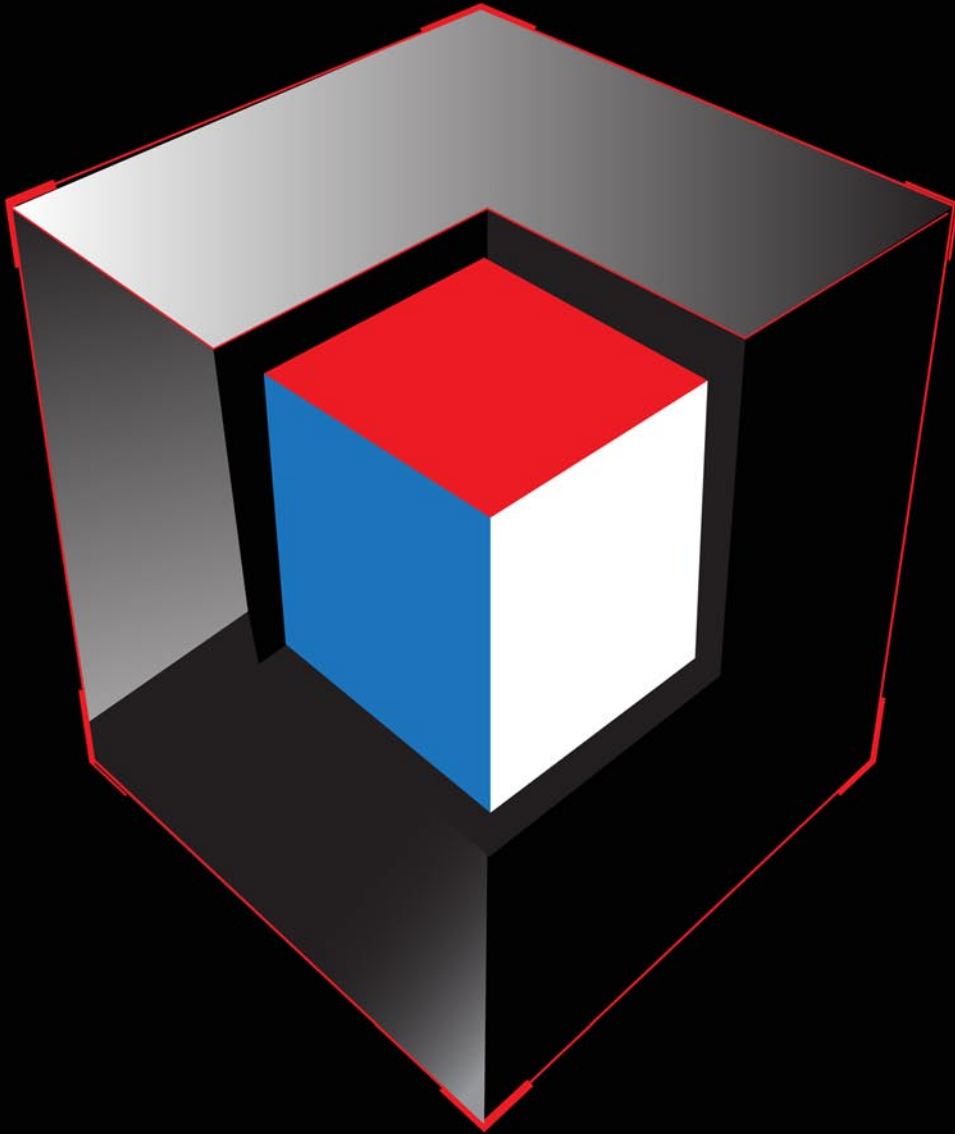
If you want to log only unwanted traffic, therefore, you have to add a matching rule with a DROP target immediately after the LOG rule. If you don't, you'll find

yourself logging both desired and unwanted traffic with no way of discerning between the two, because by default iptables doesn't state why the packet was logged in its log message. This example logs a summary of failed packets to the file `/var/log/messages`. You can use the contents of this file to determine which TCP/UDP ports you need to open to provide access to specific traffic that is currently stopped (Listing 20). Here are some examples of the output of this file:

- Firewall denies replies to DNS queries (UDP port 53) destined to server 192.168.1.102 on the home network.

Listing 18. *Established connection in both directions*

```
#-----  
# Load the NAT module  
#  
# Note: It is best to use the /etc/rc.local example in  
#       this  
#       chapter. This value will not be retained in the  
#       /etc/sysconfig/iptables file. Included only as a  
#       reminder.  
#-----  
modprobe iptable_nat  
#-----  
# Get the IP address of the Internet interface eth0  
#       (linux only)  
#  
# You'll have to use a different expression to get the  
#       IP address  
#       for other operating systems which have a different  
#       ifconfig output  
#       or enter the IP address manually in the PREROUTING  
#       statement  
#  
# This is best when your firewall gets its IP address  
#       using DHCP.  
# The external IP address could just be hard coded  
#       ("typed in  
#       normally")  
#-----  
external_int="eth0"  
external_ip="'ifconfig $external_int | grep 'inet addr'  
#       | \  
#       awk '{print $2}' | sed -e 's/.*://'"  
#-----  
# Enable routing by modifying the ip_forward /proc  
#       filesystem file  
#  
# Note: It is best to use the /etc/sysctl.conf example  
#       in this  
#       chapter. This value will not be retained in the  
#  
# /etc/sysconfig/iptables file. Included only as a  
#       reminder.  
#-----  
echo 1 > /proc/sys/net/ipv4/ip_forward  
#-----  
# Allow port forwarding for traffic destined to port 80  
#       of the  
#       firewall's IP address to be forwarded to port 8080 on  
#       server  
# 192.168.1.200  
#  
# - Interface eth0 is the internet interface  
# - Interface eth1 is the private network interface  
#-----  
iptables -t nat -A PREROUTING -p tcp -i eth0 -d  
#       $external_ip \  
#       --dport 80 --sport 1024:65535 -j DNAT --to  
#       192.168.1.200:8080  
#-----  
# FORWARD chain.  
# Connections on port 80 to the target machine on the  
#       private  
#       network must be allowed.  
#-----  
iptables -A FORWARD -p tcp -i eth0 -o eth1 -d  
#       192.168.1.200 \  
#       --dport 8080 --sport 1024:65535 -m state --state NEW  
#       -j ACCEPT  
iptables -A FORWARD -t filter -o eth0 -m state \  
#       --state NEW,ESTABLISHED,RELATED -j ACCEPT  
iptables -A FORWARD -t filter -i eth0 -m state \  
#       --state ESTABLISHED,RELATED -j ACCEPT
```

HITBSECCONF 2012

AMSTERDAM

May 20th - 25th @ Okura Hotel Amsterdam

REGISTER ONLINE

<http://conference.hitb.org/hitbsecconf2012ams/>

```
Feb 23 20:33:50 bigboy kernel: IN=wlan0 OUT=
MAC=00:06:25:09:69:80:00:a0:c5:e1:3e:88:08:00
SRC=192.42.93.30
DST=192.168.1.102 LEN=220 TOS=0x00 PREC=0x00 TTL=54
ID=30485 PROTO=UDP SPT=53
DPT=32820 LEN=200
```

- Firewall denies Windows NetBIOS traffic (UDP port 138)

```
Feb 23 20:43:08 bigboy kernel: IN=wlan0 OUT=
MAC=ff:ff:ff:ff:ff:ff:00:06:25:09:6a:b5:08:00
SRC=192.168.1.100
```

Listing 19. Create alias IP addresses

```
#-----
# Load the NAT module
#
# Note: It is best to use the /etc/rc.local example in
#       this
# chapter. This value will not be retained in the
# /etc/sysconfig/iptables file. Included only as a
#       reminder.
#-----
modprobe iptable_nat
#-----
# Enable routing by modifying the ip_forward /proc
#       filesystem file
#
# Note: It is best to use the /etc/sysctl.conf example
#       in this
# chapter. This value will not be retained in the
# /etc/sysconfig/iptables file. Included only as a
#       reminder.
#-----
echo 1 > /proc/sys/net/ipv4/ip_forward
#-----
# NAT ALL traffic:
#####
# REMEMBER to create aliases for all the internet IP
#       addresses below
#####
#
# TO: FROM: MAP TO SERVER:
# 97.158.253.26 Anywhere 192.168.1.100 (1:1 NAT -
#       Inbound)
# Anywhere 192.168.1.100 97.158.253.26 (1:1 NAT -
#       Outbound)
# Anywhere 192.168.1.0/24 97.158.253.29 (FW IP)
#
# SNAT is used to NAT all other outbound connections
#       initiated
# from the protected network to appear to come from
# IP address 97.158.253.29
#
# POSTROUTING:
# NATs source IP addresses. Frequently used to NAT
#       connections from
# your home network to the Internet
#
# PREROUTING:
# NATs destination IP addresses. Frequently used to NAT
# connections from the Internet to your home network
#
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#-----
# PREROUTING statements for 1:1 NAT
# (Connections originating from the Internet)
iptables -t nat -A PREROUTING -d 97.158.253.26 -i eth0 \
-j DNAT --to-destination 192.168.1.100
# POSTROUTING statements for 1:1 NAT
# (Connections originating from the home network
#       servers)
iptables -t nat -A POSTROUTING -s 192.168.1.100 -o eth0 \
-j SNAT --to-source 97.158.253.26
# POSTROUTING statements for Many:1 NAT
# (Connections originating from the entire home network)
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 \
-j SNAT -o eth0 --to-source 97.158.253.29
# Allow forwarding to each of the servers configured
# for 1:1 NAT
# (For connections originating from the Internet.
#       Notice how you
#       use the real IP addresses here)
iptables -A FORWARD -p tcp -i eth0 -o eth1 -d
#       192.168.1.100 \
-m multiport --dports 80,443,22 \
-m state --state NEW -j ACCEPT
# Allow forwarding for all New and Established SNAT
#       connections
# originating on the home network AND already
#       established
# DNAT connections
iptables -A FORWARD -t filter -o eth0 -m state \
--state NEW,ESTABLISHED,RELATED -j ACCEPT
# Allow forwarding for all 1:1 NAT connections
#       originating on
# the Internet that have already passed through the
#       NEW forwarding
# statements above
iptables -A FORWARD -t filter -i eth0 -m state \
--state ESTABLISHED,RELATED -j ACCEPT
```

Listing 20. Determine which TCP/UDP ports you need

```
#-----
# Log and drop all other packets to file /var/log/
#           messages
# Without this we could be crawling around in the
#           dark
#-----
iptables -A OUTPUT -j LOG
iptables -A INPUT -j LOG
iptables -A FORWARD -j LOG
iptables -A OUTPUT -j DROP
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

stopped and the `/etc/init.d/iptables` script running without the typical [OK] or [FAILED] messages.

If you have just installed iptables and have never applied a policy, then you will face this problem.

Unfortunately, running the `service iptables save` command before restarting won't help either. You have to create this file.

```
[root@bigboy tmp]# service iptables start
[root@bigboy tmp]#
[root@bigboy tmp]# touch /etc/sysconfig/iptables
[root@bigboy tmp]# chmod 600 /etc/sysconfig/iptables
[root@bigboy tmp]# service iptables start
Applying iptables firewall rules: [ OK ]
[root@bigboy tmp]#
```

Conclusion

A firewall is a critical part of any establishment that connects to an unprotected network such as the Internet, but a firewall is never sufficient. Web site security involves not just protection from corrupted packets or maliciously overwhelming volumes of traffic, but also involves daily data backups to help recovery from device failures, regular application patching, enforced password policies, restricted and monitored physical access to your servers, reliable power and cooling, secured cabling, redundant hardware, and, probably most importantly, well trained and motivated employees. Security should be viewed as anything that contributes to the desired risk-free functioning of your site, and it is well worth the money to invest in and learn from a book that specializes in the topic.

```
DST=192.168.1.255 LEN=241 TOS=0x00 PREC=0x00 TTL=64 ID=0
DF PROTO=UDP SPT=138
DPT=138 LEN=221
```

- Firewall denies Network Time Protocol (NTP UDP port 123)

```
Feb 23 20:58:48 bigboy kernel: IN= OUT=wlan0
SRC=192.168.1.102
DST=207.200.81.113 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=0
DF PROTO=UDP SPT=123
DPT=123 LEN=56
```

The traffic in all these examples isn't destined for the firewall; Therefore, you should check your INPUT, OUTPUT, FORWARD, and NAT related statements. If the firewall's IP address is involved, then you should focus on the INPUT and OUTPUT statements. If nothing shows up in the logs, then follow the steps in *Chapter 4, "Simple Network Troubleshooting,"* to determine whether the data is reaching your firewall at all and, if it is not, the location on your network that could be causing the problem.

As a general rule, you won't be able to access the public NAT IP addresses from servers on your home network. Basic NAT testing requires you to ask a friend to try to connect to your home network from the Internet.

You can then use the logging output in `/var/log/` messages to make sure that the translations are occurring correctly and iptables isn't dropping the packets after translation occurs.

iptables Won't Start

The iptables startup script expects to find the `/etc/sysconfig/iptables` before it starts. If none exists, then symptoms include the firewall status always being

HAMIDREZA MOHEBALI

Interview With Richard Johnson

Richard Johnson is a computer security specialist who spends his time playing in the realm of software vulnerability analysis. Richard currently fills the role of principal research engineer on Sourcefire's Vulnerability Research Team, offering 10 years of expertise in the software security industry. Current responsibilities include research on exploitation technologies and automation of the vulnerability triage and discovery process. Past areas of research include memory management hardening, compiler mitigations, disassembler and debugger design, and software visualization. Richard has released public code for binary integrity monitoring, exploit mitigations, program debugging, and reverse engineering and has presented at more than 20 conferences worldwide since 2004. Richard is also a co-founder of the Uninformed Journal and a long time resident of the Hick.org ranch.

Hakin9: I've heard that you start programming at a young age.

Richard Johnson: Yes, my dad was in the military and we lived in Germany when I was young. He was involved in a Nuclear Biological Chemical Program and they had to be able to communicate back the U.S. so I had early exposure to the concept of networking. My dad was a bit of a hobbyist in computers and we had a Commodore 64 and he started getting on BBS's. I learned how to program BASIC at first out of books and then started modifying games. Later, I also started to get on BBS's myself and just kind of ran with it. From there I found people who were very interested in exploring technology and I was excited about the potential to be able to connect with other people and have access to information about technology. I guess when you grow up travelling you want to figure out how to stay connected. It was inspiring.

Hakin9: Instead of playing games?

RJ: Of course I liked games when I was young so I started to modify games to my liking. You know, BASIC is not compiled so it's the first open source I had access to. I went from games to programming, dialup BBS's, and then finally on to the Internet. First thing I did when I got to the Internet was look for a BBS because that is what I was familiar with. I eventually found a telnet

hacking BBS where I could build my knowledge about phreaking and hacking.

Hakin9: And it determined your future?

RJ: Yeah, it pretty much did, it was exciting, fun, and enabling. It's powerful for young person to be able to manipulate technology so that early exposure eventually led to my career path.

Hakin9: Tell us a little bit about Sourcefire's Vulnerability Research Team, its composition and role in the company?

RJ: The role of VRT is to provide all of the intelligence and threat information that goes into our core products. This includes the vulnerability research that goes into developing signatures for our Sourcefire Next-Generation IPS (NGIPS) as well as malware analysis that supports our anti-malware products. We also have part of the team, including myself, that is dedicated to researching new technologies, threats, and trends. We get information from public sources, like the open source community mailing lists, as well as participating in partnership programs to get early threat reports. We also have people who build the VRT infrastructure, like our fuzzer farms and the interfaces to other open source projects like Razorback™, so that we can develop these prototypes rapidly and try to figure out what works, what doesn't and what

we should pursue so we can eventually put it in our products.

Hakin9: Sourcefire's Vulnerability Research Team is composed of security experts, how do they keep their knowledge up-dated?

RJ: Within the VRT we all have our own approaches to staying educated on the latest technology. Most importantly, we try to share this information within the group so the efforts of one person's knowledge can be valuable to the rest of the team. Personally, I am an active member in the security community. I speak at security conferences in order to try to network with industry leading researchers and I stay up to date by researching and reading absolutely everything that is coming out from this group specifically around vulnerability research. It's a social community and collaboration on ideas is common. Once you establish the direction you want to go then you typically branch off and make it your own and say here is what I think we should be doing to attack this problem that's out there.

Hakin9: Is it happens sometimes that you recruit skilled specialists, you know you just got an email or in this social community you just heard from one guy or you get those vulnerability in your programme and so on is it happens sometimes? like when some hacker tries to hack Microsoft servers, Microsoft software and they try to recruit him, so he could create better security solutions?

RJ: Actually, we just recently hired an individual who is very active in the open source Snort® Community. We were receiving emails from him with new signatures or some improvement he found in an existing signature. He would send in patches and he was already contributing to the group so we felt that it was appropriate to extend him an offer and brought him on board. Finding the right people for the team is an extremely difficult, so when we see somebody who can demonstrate their capability and is willing to be cooperative and establish a relationship we take notice.

Hakin9: When you hire new employees what kind of skill-set do you look for?

RJ: We do network security, vulnerability research, and malware analysis so the skillsets involved would include a strong knowledge of network protocols, programming skills in a variety of languages, debugging, exploit development, reverse engineering, and the ability to create sandboxes for malware analysis. We look for people who have a strong logical understanding and problem solving abilities, and creative thinkers who are

able to think outside the box. We are a very dynamic group and there aren't a lot of rules and structures around limits of what we can do, so we are looking for people that are going to bring new ideas to the table and drive us in new directions.

Hakin9: So just some little contribution in this open group is a good way to show abilities.

RJ: Of course, that shows that somebody out there is already taking a great interest in our product and is willing to contribute and often times has ideas in ways to improve it. We are always looking for new innovation and technology, that's one of the main advantages of Sourcefire.

Hakin9: When you hire someone, some specialist do you allow him to contribute in other social project like for example you have employee and he wants to contribute to the Bitcoin Project.

RJ: As far as contributing to open source projects, that's absolutely encouraged. In fact, most of our new projects are incubated as open source. I've been talking about Razorback at a lot of recent conferences, and Razorback is a project the VRT developed that is currently an open source solution for analyzing files for exploits or malware on the wire. This project came out of a business meeting we had with a customer. We were discussing some of the current security problems they had and we decided to try to do some research and tackle the problem of targeted exploits in rich document formats. The first way we can test whether it's effective is to give it away and see what people think about it. That's the philosophy Sourcefire has had since the beginning because Snort was written by our CTO as an open source project.

Hakin9: Do you train your team in any specific way in area of IT security or just recruit guys who take care of it themselves?

RJ: We are a highly skilled team and offer our own training sessions to other companies. We expect new members of the team to bring their own skills to the table and we extend upon that through collaboration and keeping up with the latest security research. Every vulnerability is going to come to us at some point to put in our product for detection so we have a very good knowledge of the latest vulnerabilities. If people are interested in researching a new aspect of computer security they are encouraged to do so and have budget to do that specifically.

Hakin9: So it is kind of individual training.

RJ: We don't go as a team all together to pay somebody else to train us, but if there are areas of knowledge that

people would add to their own personal capabilities it's encouraged.

Hakin9: Since Sourcefire's VRT supports Snort, can you tell us the market penetration of Snort and how it measures up with their competitors?

RJ: According to Gartner, an independent technology research organization that is respected by global business, we are a leading vendor for our market space. According to their reports, we are considered to have the best technological coverage of any IPS available and we are also the only vendor to align to their new definition of Next-Generation Network IPS. We are not a large corporation that has a small team handling this aspect of their business, this is our business and our team is dedicated to security solutions. As far as competition, our commercial NGIPS with Snort as well as FireSIGHT™ technology at the foundation, creates a much more context-aware and higher quality rate of protection for enterprises, and with this kind of innovation we stand alone.

Hakin9: Tell us a little bit about the Adobe Acrobat X sandbox and how you got involved.

RJ: I have history of doing mitigation review and also some design. It pretty much started at my very first talk at Black Hat in 2004. I did a joint talk with another individual, Peter Silberman, on all of the mitigation technology that was available on Linux or Windows at the time. We were talking about compiler mitigation like ProPolice, Stackguard as well as host mitigations such as PaX. Later, I worked with Microsoft for about five years and as part of my job I helped them review the mitigations in Windows. A co-worker from Microsoft went to the Adobe security team about a year before I left Microsoft so my old colleague contacted us and requested that we review the new sandbox. I delivered a report to them before they released the product and afterward I gave a couple of public talks at security conferences that gave an overview of the design and a few of the remaining weaknesses in the sandbox including an ASLR bypass.

Hakin9: In one of your presentations you mentioned about Windows sandboxing, what about other platforms, have they been extended to other platforms?

RJ: No, the Adobe sandbox was only implemented on the Windows platform. Windows provides API's and access controls they utilized in the design of their sandbox. Currently, they have not implemented a sandbox for the Apple or Linux versions of their software.

Hakin9: Are there any plans to introduce it to Mac or Linux?

RJ: You would have to ask Adobe. If they wanted to they could, but Adobe gets exploited mostly on the Windows platform so they addressed the problem that was most prevalent. They haven't give me any indication as to whether they plan to improve the existing implementation or include a sandbox with future versions for other platforms.

Hakin9: Is Adobe doing enough to secure Adobe Reader/Acrobat products?

RJ: They are moving in the right direction. They are being more proactive than most vendors and I think they have responsibility to because PDF is one of the most prevalent document formats out there. I think they recognise their responsibility and the fact that people are exploiting vulnerabilities in Adobe Reader so they are taking the right steps to begin the process of making their products more secure and also doing a good job at information sharing to help vendors better secure the customers who use those products. Anybody who works in mitigation's understands that it is a cat and mouse game. Historically, every mitigation has been defeated and incrementally improved.

Hakin9: Can you explain Vulnerability research workflow?

(Reference: Harder, better, faster, stronger paper at BlackHat 2010)

RJ: This is actually my main interest at Sourcefire. I've been working on automation for vulnerability discovery and triage for quite some time. I gave a talk with Lurene Grenier at Black Hat USA 2010 that previewed a workflow solution for vulnerability research. This is a system that end to end allows you to generate bugs through fuzzing and then automatically analyze them to an extent where an analyst can have as much information as possible when writing signatures for our products. It's basically an approach that realises that the problem is not necessarily generating bugs but actually having the process around the vulnerability discovery to effectively take advantage of that information and put it back in our product. Normally, we have so much information coming in that prioritization must take place. My interest is automating as much as possible so the limited human resources we have can handle the amount of information.

Hakin9: Do you think that developing the automatic part of this analysis is something that will improve the overall analysis

RJ: My goal is to allow each of the people in my team to be more effective which in turn will allow us to put

more of this threat intelligence into our product. Even if we don't count the fuzzing part, the processing of bugs and the analysis of the vulnerabilities and providing information to the analyst who writes the signatures is very important to our core business. The vulnerabilities we discover allows us to have coverage of vulnerabilities nobody else has which can be key when working with networks that are attacked with zeroday exploits. We actively try to find these problems before anybody else does.

Hakin9: Aren't you afraid that extending the automatic analyse will lead to some situations that this analysing machine will be hacked for example?

RJ: No, this is not dealing with malware this is dealing with software vulnerabilities so there is no risk of launching a malicious payload. We identify the vulnerability and we need to understand the exact conditions that trigger the bug so we can build protection into our products.

Hakin9: How does Data flow analysis help businesses create more secure products?

RJ: Dataflow analysis is a method for understanding the relationship between areas of memory. We utilize this method as part of our ability to track tainted memory. Memory that has been influenced by an external input such as file or network data is considered to be untrusted or *tainted*. If untrusted data influences a program's execution and leads to a crash, there is a potential security vulnerability. We use taint analysis to determine the exploitability, severity, and exact conditions that lead to a crash. This information allows analysts to write accurate signatures rapidly.

Hakin9: What is the current state of fuzzing tools? Can you recommend a couple of tools.

RJ: There are two main approaches. Dumb fuzzing is a random insert of corruption into a file or network stream packet and effectively it still finds hundreds of vulnerabilities in all the modern formats today, so one way to do fuzzing is to put as many resources as possible and manipulating these files and running these through as many iterations as possible. There have been various attempts on improving dumb fuzzing such as using feedback to influence the manipulations, but even the simplest of fuzzers are still effective. The other method would be the use of a framework for developing a fuzzer that's very specific to a format or protocol. There are not a lot prepackaged tools out there available that come with formats predefined, but there are some excellent frameworks and libraries and the one that I specifically use and would recommend over most would be the Peach Framework that's developed by Michael Eddington.

Hakin9: Information visualization falls into Human Computer Interaction (HCI) domain, what got you interested in that field?

RJ: When dealing with run-time analysis traces you end up with an extremely large amount of data, especially when you trying to track memory and the data as opposed to just the code. So I started getting into looking at graphs and basic visualisations and then I started reading a little bit more about the science of visualisations and how the brain process is so much more information visually than it can possibly accomplish analytically. I spent about 6 months or so researching this area and had a wonderful opportunity to speak at MIT and present my ideas on how to handle multifaceted data problems through visualisation. You can look at extreme amounts of data at one time visually in a graph or a picture compared to looking at the same data in a giant spreadsheet.

Hakin9: It's a good way to solve some problems or to generate some new approach for still existing problems in security?

RJ: Just last week I was speaking at the Internet Security Days conference in Germany and the main problem they were discussing was malware propagation. There were some great new examples of visualisation. There was a gentleman from the Japanese government who showed how they watch global propagation of worms and they had very unique visualisation that was highly intuitive. It doesn't give you specific knowledge, but it gives you general knowledge and general knowledge is how you figure out what's actionable. Then you take specific knowledge from that area of interest usually push it off to analysts to drill deeper. When you need a high level view of a very large set of information, visualisation is the way to go.

Hakin9: Vulnerability management can potentially generate tremendous amount of data, how can it be visually simplified for better decision-making?

RJ: One example is if you want to associate levels of priority in a visualisation. If you were to look at your network and you discover that there is a worm propagating your going to want to respond in order of the importance of target that's being attacked. So for example, if you work in a organisation where you have technical support, sales, engineers, and corporate managers and there is a worm on your network, there are certain assets that would have more valuable data which needs to be protected quicker. Visualisation could be utilised to determine the propagation of the attack in real time and the endpoint target is just an IP address but if you can associate that organizational

value based upon the information the target contains, then you know where you should be fixing problems first.

Hakin9: What is MoFlow?

RJ: Moflow is the framework that the vulnerability research workflow is based upon. Moflow is a fairly comprehensive project that covers the the libraries and the code that do the fuzzing and the analysis software that allows you to execute traces and determine the data flow propagation. It has graphical user interface were you can visualise traces and programs in a graph, and finally also connects to reporting database that allows you to just see a history of we fuzzed this and this was the cause of the bug. Basically bug discovery and triage broken into logical steps. This project is something very important to me because this platform will allow all the people in the VRT who write signatures to work smarter and faster and gain more information through automated analysis than they have now.

Hakin9: Your team has been developing Razorback, can you tell us more about your exciting work?

RJ: The Razorback is a framework for assisting in the deep inspection of targeted client side attacks. What this does it allows you to take apart the actual file attachments and analyse them in a distributed clustered framework for vulnerabilities, malware, and also allows you to find embedded documents. So you have a container object like a PDF file, and the PDF file has flash file inside of it some anti-virus programs wont be able to determine that there is also some Javascript inside the flash file and its this Javascript that's the exploit. That embedded document is going to bypass most security protection available today so we developed Razorback to approach the problem of really complicated file format or client side attacks or targeted attacks in particular. Last year Symantec Internet Threat Security Report had a statistic in there and I believe it's 93% of actively exploited in the wild today are client side.

Hakin9: How long it takes to analyse those threats?

RJ: Since we basically have this architecture that involves several steps in a process each one of those steps can be performed on one or more sets of hardware. For example we have taken the open source version of Snort and removed all of the detection capability from it so all it really does is harvest files, it does packet capture and the stream assembly and then we'll determine if there is file in that stream so if there is a HTTP session it will extract all the files that you may

have downloaded, if there is mail SMTP connection it will extract the attachments, if there your application crashes we can take that crash dump and push it into a system like Moflow. The capture is meant to be as lightweight as possible and the analysis is distributed so it can be scaled to match the resources appropriate for your network

Hakin9: What was the first feedback after releasing this programme?

RJ: It has been actively developed and there is a small community slowly growing. People seem to think that it is a good approach for this problem and it's a matter of getting more exposure and building a community just like Snort and hopefully soon it will find its place right alongside installations of Snort.

Hakin9: How actively involved are you in Information Security community locally and internationally?

RJ: I am very involved, I have been presenting publicly every year for the last 10 years on new research topics. It really is a personal passion of mine to pass on this knowledge. When I began my journey into computer security I was a young kid and the only way to gain that knowledge back then was through communication and other people mentoring you. There are definitely a couple of people in my life that provided the inspiration and the patience for that and so I have taken that into my professional life and made sure I am giving back to the community. I believe strongly in open sourcing of knowledge as well as source code. I try to stay as involved as possible, so this year alone I have spoken at a 5 conferences and presented on three unique research topics. I was the first person who discuss the Adobe Reader sandbox and show weaknesses in it. I also published code that provide mitigation's for return oriented programing and just in time shell code which are the latest exploitation methodologies out there. Now I am presenting Razorback. I try to share at least one new set of ideas publicly each year and encourage people to give me feedback and build that network to share knowledge.

ABY RAO

Passware Password Recovery Kit Forensic 11.0

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

Now with Mac User Password Recovery!

Passware Kit Forensic includes over 30 password recovery tools, Encryption Analyzer Professional, Search Index Examiner, FireWire Memory Imager, and a Portable Version to provide immediate password recovery for any protected file detected on a PC or over the network while scanning. It recovers or resets passwords for more than 200 different types of files, as well as decrypts hard drives, PGP archives, and unlocks Windows 7 and Mac OS Lion Administrator accounts. Many types of passwords are recovered or reset instantly.



Key Features

- Scans computers and network for password-protected files
- Recovers passwords for **200+ file types** Updated
- Unlocks hard drives protected with **BitLocker** and **TrueCrypt**
- Retrieves electronic evidence in a matter of minutes from a Windows Desktop Search Database
- Includes a **Portable Version** that runs from a USB thumb drive and recovers passwords without installation on a target PC
- Acquires memory images over FireWire Updated
- Recovers Mac user login passwords from computer memory New!

Advanced Features

- Instant recovery for many password types
- Acceleration with distributed computing **(Distributed Password Recovery)**
- Multiple-core CPUs and nVidia GPUs acceleration
- **Tableau TACC** hardware acceleration
- 8 different password recovery attacks (and any combination of them) with an easy-to-use setup wizard
- Detailed reports with MD5 hash values



After losing my password to important encrypted documents, I thought it was the end of the world. Thanks for saving my work, Passware.

Conor LaHiff, LaHiff & Company.

5 editions for consumers, small business, professional, corporate, and forensic users.

Starting from **\$49!**

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

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

Contacts
 Nataly Koukoushina
 media@lostpassword.com
 Phone: +1 (650) 472-3716 x 101



Join Today Free!



Go Premium to support & enjoy the full potential!

New

Astalavista - The IT News and Security Community

- Forum Posts SHOW
- Downloads SHOW
- Events HIDE
- Official Blog SHOW
- News SHOW
- Jobs SHOW

Astalavista has taken another step into the future.

Stay Up-to-date

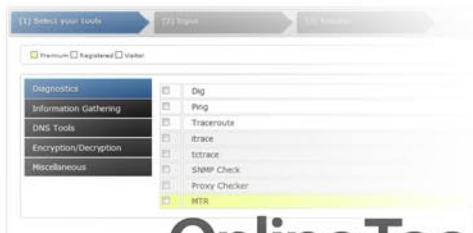
With our relaunch we focus even more on the IT & Security world.

Our continuous news stream on the main page gives you all the information you need – 24/7. What do you think about that? Give us a shout on our Astalavista Blog, you find it by clicking on the first news item on our news stream.

Now 25% OFF!

Join Today
Use coupon: **hakin9astadiscount**
www.astalavista.com

Go Premium!



Online Tools

The new **Online Tools** overview page features nearly 50 tools covering typical IT needs, like Whois, Dig, Proxy List or Encryption.

The **Rainbow tables** section lets you hash your plain text in more than forty different types and crack your hashes. The **blacklist checker** runs your domain against the most important black lists and checks if your IP/Domains are flagged as spam.



Wargames

Wargames by its broad definition is a military drill under real life conditions. It is about testing strategies without the actual combat.

The **"World Gold Reserve"** is where most of the world's gold is stored. The combat in IT is virtual. Here the purpose of a wargame server is to allow you to practice hacker tricks without damaging anything or violating the law. The aim is to find gaps in security and to learn the necessary precautionary actions to prevent this.

Go Premium to support & enjoy the full potential!

Astalavista.com

IT News and Security Community