# Is Microsoft a Threat to National Security? Policy, Products, Penetrations, and Honeypots

**By**

**Trevor U. Watkins**

Submitted in Partial Fulfillment of the Requirements

**For the Degree of**

**Master of Computing and Information Systems**

**Youngstown State University**

**May 2009**

# Is Microsoft a Threat to National Security? Policy, Products, Penetrations, and Honeypots

## Trevor U. Watkins

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Trevor U. Watkins Student Date

Approvals:

Dr. Graciela Perera, Thesis Advisor Date

Dr. John Sullins, Committee Member Date

Dr. Alina Lazar, Committee Member Date

Peter J. Kasvinsky, Dean of School of Graduate Studies & Research Date

# Abstract

Is Microsoft a threat to national security? This thesis evaluates Microsoft's policies, business model, and products to determine whether Microsoft is a threat to national security. The first part of this thesis investigated Microsoft's policies and products. In the second part of this thesis, two networks were investigated. The first network, which will be known as network "honey," was designed and configured to examine the techniques of hackers. The second network, which will be known as network "X," is a real business enterprise network that was the target for penetration testing. The investigation provided an inside look at the security threats in Microsoft Windows XP SP3, Windows Vista SP1, Microsoft Server 2000 SP4, and Microsoft Server 2003 SP2 operating systems on a network. The results of this investigation serve as a microcosm to a macro-problem. Microsoft Windows networks are too vulnerable to serve as the backbone for any institution or organization's networking infrastructure, especially entities considered to be government critical infrastructures.

# Acknowledgements

I would first like to thank those directly involved with the process of the creation of this thesis. I would like to thank Dr. Graciela Perera for her patience, feedback, suggestions, constructive criticism, and her passion for teaching, which is infectious to all that works with her and for her. I would like to thank Dr. John Sullins and Dr. Alina Lazar for all of their help and guidance, both directly and indirectly to make the experience here at Youngstown State University worthwhile. Last but not least, I would like to thank the late Dr. Karen Duda, for her help with many of the scholarships and grants I received as an undergraduate, which helped enable me to build a computer lab at home, the primary resource for this research.

I would like to give special thanks to my wife Lael H. Watkins, who allowed me to purchase all of the expensive computing equipment used in this research and for her love and support. And more than anyone in the world, I would like to thank my parents, the late Ulysses Watkins and Lenora Watkins for instilling in me the importance of furthering my education and for that Tandy 1000 back in 1987, which is the day I was born again.

# *TABLE OF CONTENTS*

# List of Tables

## List of Figures

# Chapter1: Introduction

Today's critical infrastructures, including banking and finance, energy, defense, communications, transportation, energy, healthcare, and government rely on information technology (IT) to deliver services essential to the public health and safety, commerce, and security. The U.S. Patriot Act defines critical infrastructure as: "systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters (Sec. 1016e) [1]." In 2005, the National Computer Security Survey provided the nation's first large-scale measure of cybercrime. "The effects of these crimes were measured in terms of monetary loss and system downtime [2]". According to this report, "Ninety-one percent of the businesses that provided information to the Department of Justice sustained one or both types of loss. The monetary loss for these businesses totaled $867 million in 2005. Cyber theft accounted for more than half of the loss. Cyber attacks cost businesses $314 million. System downtime caused by cyber attacks and other computer security incidents totaled 323,900 hours. Computer viruses accounted for 193,000 hours and other computer security incidents resulted in more than 100,000 hours of system downtime [2]." It has been estimated that American corporations and businesses collectively have total annual financial losses exceeding 200 and 250 billion dollars per year due to cybercrime. Because the massive growth of the Internet has created a global economy that allows businesses to expand their products and solicit new consumers with a click of a mouse, critical services have become increasingly dependent on computer systems. It is now the case that American corporations and businesses can no longer function without a network of computers, and as these institutions move more of their products and services online, their vulnerability to theft, fraud and other breaches continues to increase. It is also the case that a majority of these companies have a Microsoft Windows network infrastructure.

Reliance on IT makes these critical infrastructures attractive targets of terrorists, criminals and other sophisticated cyber attackers [3]. Throughout the last ten years, there have been increased attacks on these critical infrastructures [3]. Security is obviously a primary concern for each of these entities as they have increased efforts to add security to their network environments, however, current security efforts suffer from the flawed assumption that adequate security can be provided in applications with the existing security mechanisms of mainstream operating systems, specifically Microsoft. Microsoft has a long-standing policy of emphasizing functionality and user friendliness over security in its operating systems. Because Microsoft focuses on "ease of use," in its operating systems, the overall security of any network containing windows boxes or servers can be severely hampered and easy to compromise. The United States Department of Defense for example, is one of the largest consumers of Microsoft's Windows family of operating systems [4]. "There are certainly a small number of organizations with a larger install base, but definitely not one as distributed, inter-connected and solely dependent upon Windows to complete just about every facet of work accomplished. The reliance solely on Windows, from the end-user workstation to the back-end server farm, is a huge risk, which the DoD has shown no desire to mitigate [4]."

With that being said, a question has to be answered. Is Microsoft a threat to national security? This thesis evaluates Microsoft’s policies, business model, and products to determine whether Microsoft is a threat to national security. The first part of this thesis investigates those policies and products. In the second part of this thesis, two networks were investigated. The first network, which will be known as network “honey,” was designed and configured to examine the techniques of hackers to determine the viability of the security of Microsoft products in a Microsoft Windows networking environment. The second network, which will be known as network “X,” is a real business enterprise network that was the target for penetration testing. Network X is a homogenous Microsoft Windows networking environment. The investigation provided an inside look at the security threats in Microsoft Windows XP SP3, Windows Vista SP1, Microsoft Server 2000 SP4, and Microsoft Server 2003 SP2 operating systems on a network. The results of this investigation serve as a microcosm to a macro-problem. Microsoft Windows networks are too vulnerable to serve as the backbone for any institution or organization’s networking infrastructure, especially entities considered to be government critical infrastructures.

The organization of this thesis is as follows. Chapter 1 gives an overview of Windows Security. Chapter 2 gives an overview of Microsoft’s policies and products. Chapter 3 describes network security monitoring and breaks down the setup and configuration of Network Honey in the experiment. Chapter 4 describes penetration tests and breaks down the methodology used to attack the targeted network in the experiment. Chapter 5 encompasses the evaluation of Windows network security and provides the results of the experiment discussed in Chapters 3 and 4. Finally, in Chapter 6, the summary and conclusions drawn from the experiments are discussed.

## 1.1: An Overview Of Windows Security

In 1998 The United States Justice Department along with 20 other states filed suit against Microsoft, in regards to Microsoft’s abuse of power in the bundling of their browser software and operating systems. On May 18, 1998 the trial started. What happened during the testimony of then Vice President for Platforms at Microsoft, Jim Allchin, raised some eyebrows in terms of the security of Microsoft Windows Operating Systems.

Allchin claimed that sharing information with competitors could damage national security and even threaten the U.S. war effort in Afghanistan, which the U.S. engaged in after the 911 attacks [5]. He later acknowledged that some Microsoft code was so flawed it could not be safely disclosed. These bold statements and candid admissions were a part of Jim Allchin's testimony during two days in court before Judge Colleen Kollar-Kotelly, who heard the case of nine states and the District of Columbia seeking stricter penalties for Microsoft's antitrust behavior.

In the sequence of executives that had to testify, Allchin was the final executive lined up to defend Microsoft [5]. Like company Chairman and Chief Software Architect Bill Gates before him, Allchin highlighted the security problems he foresaw that could result from technical information disclosure requirements sought by the nonsettling states. "It is no exaggeration to say that the national security is also implicated by the efforts of hackers to break into computing networks," Allchin testified [5]. "Computers, including many running Windows operating systems, are used throughout the United States Department of Defense and by the armed forces of the United States in Afghanistan and elsewhere [5]."

Unlike the states' proposed remedy, the federal settlement proposal that Microsoft and the Department of Justice agreed to contained a carve-out that permitted Microsoft to withhold API and protocol disclosures if such disclosures would compromise security. The provision was designed to address hackers, viruses and piracy, according to Allchin [5]. In his testimony, Allchin also addressed .Net and countered charges made by rivals, particularly Jonathan Schwartz, senior vice president of corporate strategy and planning at Sun Microsystems Inc. about its interoperability. Charging that Schwartz's testimony oversimplified the interoperability of .Net and Java technology, Allchin claimed the two systems were not perfect equivalents. "Microsoft has invested substantial time and resources in providing great interoperability between .Net and older technologies," Allchin said. "Sun's strategy of promoting '100 percent pure' Java applications discourages interoperability [5]."

During his second day on the stand, Allchin conceded that Microsoft had already identified at least one protocol and two APIs that it planned to withhold from public disclosure under the security carve-out. The protocol, which is part of Message Queuing, contained a coding mistake that would threaten the security of enterprise systems using it if it were disclosed, Allchin said. When Kevin Hodges, attorney for the dissenting states, asked him how many APIs would be exempt, Allchin said he did not know the exact number, but it would include APIs that deal with anti-piracy and digital rights management [5].” Microsoft has already identified APIs involved with Windows File Protection that would be withheld, he said. When pressed for further details, Allchin said he did not want to offer specifics because Microsoft was trying to work on its reputation regarding security. "The fact that I even mentioned the Message Queuing thing bothers me," he said during his testimony [5]. What made this testimony so credible and forthcoming is that Jim Allchin was a main component and major player for Microsoft. During his introduction to the court, he states, “My name is Jim Allchin. I am the Group Vice President for Platforms at Microsoft Corporation. I have overall responsibility for the technical architecture, engineering and product delivery for all of Microsoft’s Windows operating systems, for portions of Microsoft’s new .NET initiative in the area of Web services, for Microsoft’s family of server applications such as the SQL Server database and Exchange email and collaboration product, and for Microsoft’s new media technologies. I am also responsible for delivering the developer tools, frameworks and product support to fulfill the promise of Microsoft’s .NET vision of interconnected software providing services across the Internet. The approximately 10,000 employees in my group build software platforms that consumers and businesses use as integral aspects of their day-to-day activities” [5]. In the following sections, we will closer look at some of the security flaws of the components of Microsoft Windows Operating System technology.

## 1.2 Windows Kernel

Every operating system has a kernel. In general, the kernel is the lowest-level, most central part of a computer operating system and one of the first pieces of code to load when the machine starts up. The task of an operating system’s kernel is to take care of the most basic of tasks a computer operating system must perform [6]. The kernel is what enables the software of the machine to talk to the hardware and is responsible for basic operating system housekeeping tasks such as memory management, launching programs and processes, and managing the data on the disk. All applications and even the graphical interface of Windows run on a layer on top of the kernel [7]. The performance, reliability, and security of the entire computer depend on the integrity of the kernel. The kernel is the most carefully coded piece of the entire operating system [6]. Since all other programs depend upon it, a glitch in the kernel can make all other programs crash or perform unexpectedly.

For example, when browsing the Internet, the browser needs processor time to properly display the web pages, while also needing space on the hard drive to store commonly accessed information, such as login credentials or downloaded files. While it is the task of the operating system to properly spread the computer's resources across running applications, it is the kernel that performs the actual act of assigning. It is synonymous to a person preparing a meal in the kitchen. The various ingredients (the computer applications) need to be prepared using kitchen appliances (the system resources) in order to form the meal (the operating system) after which can be served to the people attending the dinner (the users). In this analogy, the person preparing the meal is the kernel because they decide when the ingredients are put into the kitchen appliances, while the meal is the operating system because it depends on the meal which ingredients and kitchen appliances are needed. This analogy also stresses the symbiotic relationship between kernel and operating system: they are useless without each other. Without a recipe, a person cannot prepare dinner; similarly, a recipe without a cook will not magically prepare itself.

### 1.2.1 Rootkits

Greg Hoglund and James Butler teamed up to co-author a book titled *Rootkits: Subverting The Windows Kernel.* Rootkits are not new, and no operating system is impregnable against this kind of attack, but they have emerged recently as one of the hot new attacks, particularly against computers running Microsoft Windows operating systems. Rootkits are a special set of hacker tools that are used after the attacker has broken into a computer system and gained root level access [8]. Usually hackers break into a system with exploits and install modified versions of common tools. These type of rootkits are called user-mode rootkits because they run in user mode [9].

Some more sophisticated rootkits have kernel-mode module components. These rootkits are more dangerous because they change the behavior of the kernel. They can hide objects from even kernel-level defense software.  For example, they can hide processes, files in the file system, registry keys, and values under Windows, and implement stealth capabilities for other malicious components [9]. The results discussed in Chapter 5 displays the vulnerabilities of the Windows Kernel to a rootkit attack.

## 1.3 The Windows Registry

When you install a program on a computer running windows, the program stores information it needs in a database called the registry [10]. This database was developed as a basis for all system-wide hardware and software parameters and custom user settings that exist in Windows. The registry is difficult to decipher and understand. However, it is one of the most important components of any modern operating system belonging to the Windows family. The registry stores all the settings of the operating system and the applications running in it [11]. The registry also stores information on all the hardware, including Plug and Play and OLE, networking parameters, hardware profiles, and user profiles. A single error in the system registry can influence the entire system configuration and prevent the operating system from booting [10].

The Microsoft Windows Operating System has its roots in MS-DOS [12]. MS-DOS was a command line interface whose configuration settings, by today's standards, barely exist. MS-DOS received its configuration settings from two small files, config.sys and autoexec.bat. The config.sys file primarily loaded device drivers, while autoexec.bat was for setting environment variables, running programs, and so on. The first Windows Graphical user interface was Microsoft Windows 3.0 [10].

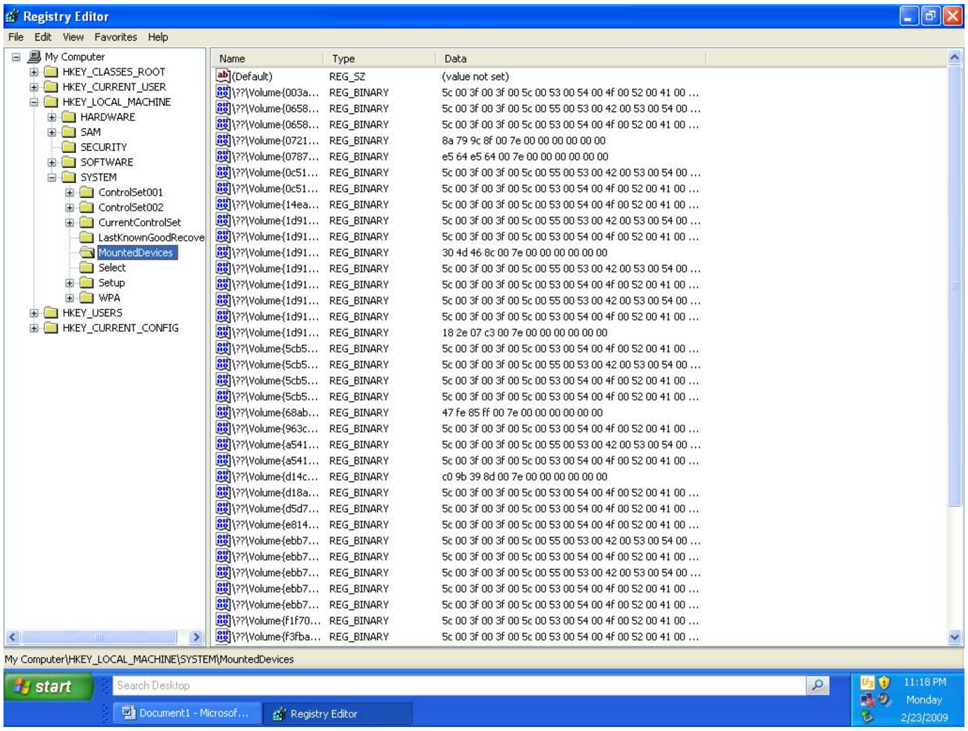
This version of Windows introduced INI files as containers for configuration files. These INI files were flat text files lacking any hierarchical structure. The configuration data was organized by sections, even so, their length and the amount of data in them made management difficult. It also was difficult to store binary data in text files. The standard Windows 3.0 installation had six INI (initialization) files: Control.ini, Program.ini, Protocol.ini, System.ini, Win.ini, and Winfile.ini. [13]. The Control.ini files contained the control panel settings. The Program.ini file contained initialization settings for Windows Program Manager. The Protocol.ini file stored initialization settings for Windows networks. The System.ini file served as the main storage for system information related to hardware, such as information about device drivers, shells to load, and so on. The Winfile.ini file contained Windows File Manager Settings, and the Win.ini file contained information mostly related to system behavior.

Windows 3.0 was upgraded to version 3.1 and with it came the rudiments of the Windows registry. The registry was a successor to INI files, which was the heart and soul Microsoft Windows 3.0 [13]. In Windows 3.1, the system registry was organized in a hierarchical file system and was used as a repository for system configuration settings.

Windows 95 and NT 3.5 expanded the registry to the structure and interface that exists in Windows XP/2003 [10]. Although the structure and interface are similar between the earlier version and today's version of the Registry, its size and complexity have grown tremendously.

At a physical level, the Windows registry is stored in files called hives [14]. The interface for the user and application takes on a logical scheme, or format. This logical structure closely resembles the directory structure used by Windows Explorer to store data in files and folders. Instead of using folders, the Registry uses keys. Instead of using files, the Registry uses values [14].

The interface by which the user primarily views, searches, or modifies the Registry is with the Registry editor tool. Figure 1 shows the Windows Registry logical view from the Registry Editor (Windows default register editor). Each folder in the left key pane is a registry key. The right panes show the key's value. The Subkey is used to show the relationship between a key and the keys nested below it. Branch refers to a key and all its subkeys. Windows uses symbolic link (i.e. similar to file system's shortcut) to link a key to a different path, which allows the same key and its values to appear at two different paths [12].



```
Name                  Type         Data
(Default)             REG_SZ       (value not set)
\??\Volume{003a...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{0658...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{0658...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{0721...    REG_BINARY   8a 79 9c 8f 00 7e 00 00 00 00 00 00
\??\Volume{0787...    REG_BINARY   e5 64 e5 64 00 7e 00 00 00 00 00 00
\??\Volume{0c51...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{0c51...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{14ea...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{1d91...    REG_BINARY   30 4d 46 8c 00 7e 00 00 00 00 00 00
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{1d91...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{1d91...    REG_BINARY   18 2e 07 c3 00 7e 00 00 00 00 00 00
\??\Volume{5cb5...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{5cb5...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{5cb5...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{5cb5...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{68ab...    REG_BINARY   47 fe 85 ff 00 7e 00 00 00 00 00 00
\??\Volume{963c...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{a541...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{a541...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{d14c...    REG_BINARY   c0 9b 39 8d 00 7e 00 00 00 00 00 00
\??\Volume{d18a...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{d5d7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{e814...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{ebb7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{ebb7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{ebb7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 55 00 53 00 42 00 53 00 54 00 ...
\??\Volume{ebb7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{ebb7...    REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{f1f70...   REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
\??\Volume{f3fba...   REG_BINARY   5c 00 3f 00 3f 00 5c 00 53 00 54 00 4f 00 52 00 41 00 ...
```

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices

**Figure 1: Windows Registry Logical View Key**

Once inside the registry there are five main folders each starting with HKEY. The abbreviation HKEY stands for *Handle to Registry Key*, which is exactly what it is, basically a reference (if you will) to a hidden registry key. These folders are virtual folders, that is; they don't really exist on your system, but are merely references to hidden, system data files. There are 5 root keys (i.e. starting point) in the Windows registry [11]. Table 1 shows the root keys and the abbreviation normally used.

**Table 1: Root Keys**

| Name | Abbreviation |
|---|---|
| HKEY_CLASSES_ROOT | HKCR |
| HKET_CURRENT_USER | HKCU |
| HKEY_LOCAL_MACHINE | HKLM |
| HKEY_USERS | HKU |
| HKEY_CURRENT_CONFIG | HKCC |

### Value

Each key has one or more values. There are 3 parts in value, which are Name, Type and Data, as shown in Table 2.

**Table 2: Value Parts**

| Value Parts | Description |
|---|---|
| Name | Every value has a unique name in that particular key |
| Type | Values type determines the type of data the value contains. The common value types in the registry for instance are: REG_BINARY type contains binary data; REG_DWORD type contains double-word (32-bit) data; REG_SZ type contains fix-length string data. |
| Data | Values data contains data, which usually relates to the values type. |

When an application read the value's data in REG_BINARY from the registry, the application decides on how to decode the value. An application can store data in binary (using REG_BINARY type) using their own data structure, hence only the application knows how to interpret it. For instance, interpreting REG_BINARY data as 8-bit ASCII or 16-bit Unicode could result in two different values. Regardless of value's type, the registry actually stores all values in binary format in the actual file. Since all values are stored alongside with their corresponding type, it allows the Registry Editor to interpret the value's data correctly [11].

### 1.3.1 Registry Root Key Organization

The HKLM and HKU root keys are the only root keys that Windows physically stores on files [11]. HKCU is a symbolic link to the subkey in HKU. HKCR and HKCC are symbolic links to subkeys in HKLM. Below are the brief descriptions of each of the 5 root keys:

***HKEY_USER***

The HKU root key contains per-user (user-specific) information. HKU contains at least these 3 subkeys:

- .DEFAULT
- SID, SID is the security identifier for the console user (user currently using the keyboard).
- SID_CLASSES contains per-user class registration and file association.

The HKU root key has another well-known SID in Windows XP.

- S-1-5-18 refers to the LocalSystem account.
- S-1-5-19 refers to the LocalService account. It is used to run local services that do not require the LocalSystem account.
- S-1-5-20 refers to the NetworkService account. It is used to run network services that do not require the LocalSystem account.

Any other subkeys in the HKU root key are associated to secondary users. Windows has a feature called Secondary Logon, which allows the user to run a program as a different user, usually with elevated privileges [11]. Thus, any user can logon to a limited account for daily routines and use elevated privileges for occasional administrative tasks. The secondary user SID (usually the administrative account SID) will only be present in the HKU subkeys if the user performs a secondary logon during the users session.

***HKEY_CURRENT_USER***

The HKCU root key contains the computer user's per-user settings. The HKCU root key is actually a symbolic link to HKU/SID, the current console user's SID [11]. This branch contains information on environmental variables, desktop settings, mapped network drive settings, and application settings [11]. Table 3 briefly describes some of the HKCU subkeys.

**Table 3: Partial HKCU Subkeys**

| Subkeys | Descriptions |
|---|---|
| Environment | Each subkey corresponds to an environmental variable the user has set |
| Identities | Each Identities subkey corresponds to an identity in Microsoft Outlook Express. Outlook Express allows multiples identities (users) to use a single mail client. However, since Windows supports multiple user profiles, users rarely have to share their mail client |
| Network | Each Network subkey corresponds to a mapped drive Windows connects during user system logon. The Subkey name is the drive letter to which the network drive is mapped. The subkey contains configuration to connect the network drive. |
| Software | Contains user-specific application settings. Programs store their settings in a standard way, HKCU\Software\Vendor\Program\Version\. Vendor is programs publisher; Program is the programs name; and Version is programs version. |
| Volatile Environment | Contains environmental variables that are defined when any user logs on to Windows |

***HKEY_LOCAL_MACHINE***

The HKLM root key contains per-computer (computer-specific) settings which apply to all users logging into that particular computer. Table 4 shows all the HKLM subkeys:

**Table 4: HKLM Subkeys**

| Subkeys | Descriptions |
|---|---|
| Hardware | Stores information regarding hardware Windows detects during startup. The subkeys are dynamically created during the system startup. They include information on device drivers and associated resources. |
| Sam | Security Accounts Manager (SAM) is a local security database which contains local users and groups information. The ACL prevents the Administrator from viewing this subkey |
| Security | Contains Windows local security database in the SAM subkey. The ACL prevents the Administrator from viewing this subkey |
| Software | Stores per-computer application settings. Programs store their settings in this standard form, HKLM\Software\Vendor\Program\Version. |
| System | Contains control set, which contains device driver and service configurations. HKLM\SYSTEM\CurrentControlSet is a symbolic link to ControlSetXXX, and the key HKLM\SYSTEM\Select indicates which ControlSetXXX is in use. |

***HKEY_CLASSES_ROOT***

HKCR contains two types of per-user settings, file associations, and class registration for Component Object Model (COM) object. File associations describe the file types and associated programs that open and edit them. The HKCR root key consumes most of the space in the registry. Windows merges two keys HKLM\SOFTWARE\Classes (contains default file associations and class registration) and HKCU\Software\Classes (contains per-user file associations and class registration) to obtain the HKCR. In fact, HKCU\Sofware\Classes is a link to HKU\SID_Classes. By merging the two keys, a program can register per computer and per-user file associations and program classes.

***HKEY_CURRENT_CONFIG***

The HKCC is a symbolic link to the current hardware profile configurations subkey, HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current. Current is a link to the key HKLM\SYSTEM\CurrentcontrolSet\Hardware Profiles\username.

### 1.3.2 Registry Hives

The Registry Editor only shows the logical structure of the registry, as shown in figure 1. Physically, the registry is not stored in a single file in the hard drive. Windows stores the registry in a few separated binary files called hives. For each hives file, Windows creates additional supporting files that contain a backup copy of the respective hives to restore the hives during a failed system boot. Only the HKLM and HKU root keys have corresponding hives (since the rest are symbolic links). However, none of 5 root keys are directly associated to a hive file [11].

The Windows registry contains lots of information that can be of potential value and helpful in aiding hackers an easy way of compromising a windows system and the network as a whole.

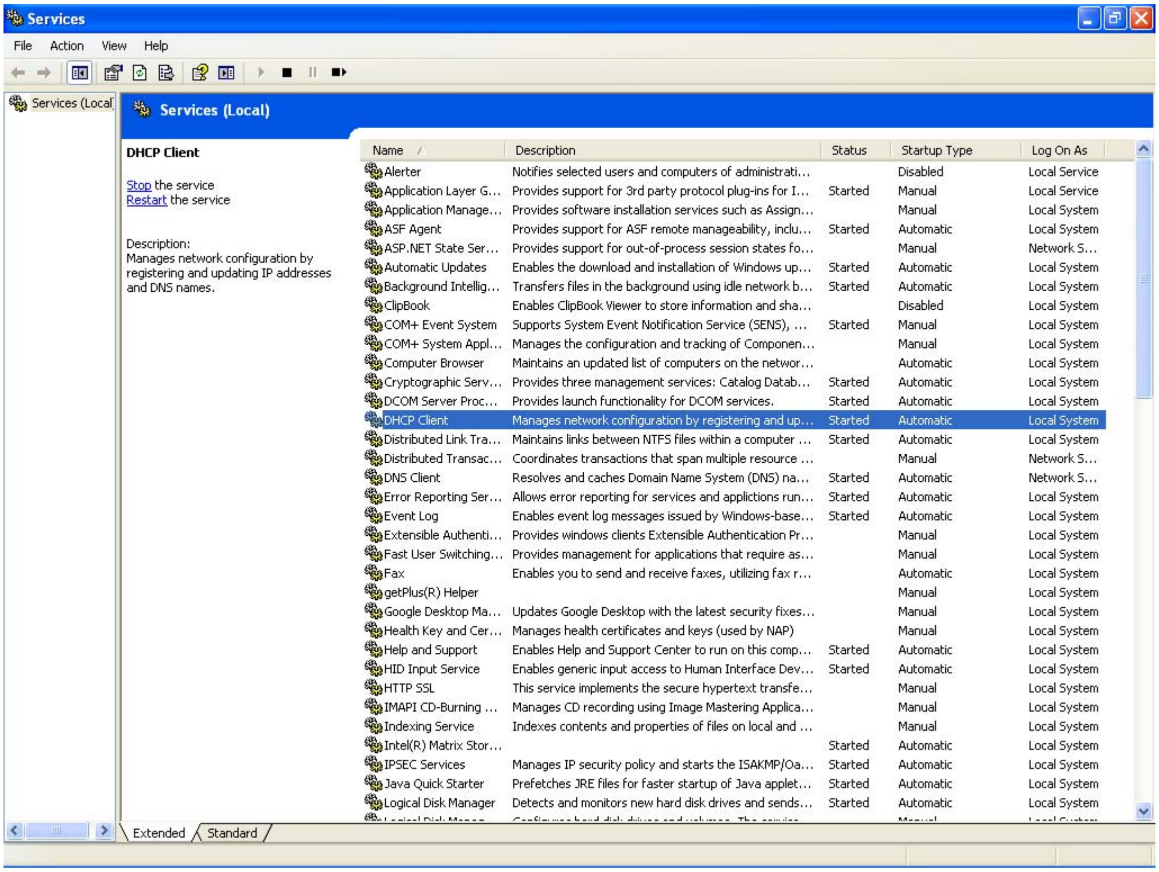
### 1.4 Windows Services

A service is an application that runs in the background, independent of any user session [13]. Previously called an NT service, the core function of a Windows service is to run an application in the background. Because services run unattended at startup, they are well suited for server-type applications such as a Web server. But this also has its drawbacks, because a user may not be aware that a service is running. Without any user interaction, one could be running a number of default services and never be aware of the potential security risks. This was made all-too-clear a while back as worms with name like "Code Red" and "Nimda" spread across the Internet, often exploiting users who were unknowingly running Web services on their workstations [13]. In turn, these infected workstations spread the worm to thousands of other systems across the Internet [13].

There are few things that make Windows Services different from a Windows application. A Windows service starts much before any user logs in to the system (if it has been setup to start at boot up process). A Windows service can also be setup in such a way that it requires a user to start it manually or let the service run automatically.

Windows services have three startup modes: automatic, manual, and disabled as shown in figure 2. But it turns out those three descriptions are not as straightforward as they at first seem. Automatic startup means that the service will be loaded at boot time. Manual, however, would be better described as on-demand startup. When a service is set for Manual startup, it remains dormant until a user starts it or an application requests it via the StartService API call. In other words, a service set for Manual startup will actually start automatically as it is needed. For example, if you run the Clipbook Viewer

(Clipbk.exe) to view the local clipboard, the Clipbook service also starts. The Clipbook service also allows the clipboard to be viewed by remote computers, a service you probably do not want running in the background. The Clipbook service depends on the Network DDE service, which in turn depends on the Network DDE DSDM service, all of which are started, if set for automatic or manual startup.



**DHCP Client**

Stop the service
Restart the service

Description:
Manages network configuration by registering and updating IP addresses and DNS names.

| Name | Description | Status | Startup Type | Log On As |
|---|---|---|---|---|
| Alerter | Notifies selected users and computers of administrati... | | Disabled | Local Service |
| Application Layer G... | Provides support for 3rd party protocol plug-ins for I... | Started | Manual | Local Service |
| Application Manage... | Provides software installation services such as Assign... | | Manual | Local System |
| ASF Agent | Provides support for ASF remote manageability, inclu... | Started | Automatic | Local System |
| ASP.NET State Ser... | Provides support for out-of-process session states fo... | | Manual | Network S... |
| Automatic Updates | Enables the download and installation of Windows up... | Started | Automatic | Local System |
| Background Intellig... | Transfers files in the background using idle network b... | Started | Automatic | Local System |
| ClipBook | Enables ClipBook Viewer to store information and sha... | | Disabled | Local System |
| COM+ Event System | Supports System Event Notification Service (SENS), ... | Started | Manual | Local System |
| COM+ System Appl... | Manages the configuration and tracking of Componen... | | Manual | Local System |
| Computer Browser | Maintains an updated list of computers on the networ... | | Automatic | Local System |
| Cryptographic Serv... | Provides three management services: Catalog Datab... | Started | Automatic | Local System |
| DCOM Server Proc... | Provides launch functionality for DCOM services. | Started | Automatic | Local System |
| DHCP Client | Manages network configuration by registering and up... | Started | Automatic | Local System |
| Distributed Link Tra... | Maintains links between NTFS files within a computer ... | Started | Automatic | Local System |
| Distributed Transac... | Coordinates transactions that span multiple resource ... | | Manual | Network S... |
| DNS Client | Resolves and caches Domain Name System (DNS) na... | Started | Automatic | Network S... |
| Error Reporting Ser... | Allows error reporting for services and applictions run... | Started | Automatic | Local System |
| Event Log | Enables event log messages issued by Windows-base... | Started | Automatic | Local System |
| Extensible Authenti... | Provides windows clients Extensible Authentication Pr... | | Manual | Local System |
| Fast User Switching... | Provides management for applications that require as... | | Manual | Local System |
| Fax | Enables you to send and receive faxes, utilizing fax r... | | Automatic | Local System |
| getPlus(R) Helper | | | Manual | Local System |
| Google Desktop Ma... | Updates Google Desktop with the latest security fixes... | | Manual | Local System |
| Health Key and Cer... | Manages health certificates and keys (used by NAP) | | Manual | Local System |
| Help and Support | Enables Help and Support Center to run on this comp... | Started | Automatic | Local System |
| HID Input Service | Enables generic input access to Human Interface Dev... | Started | Automatic | Local System |
| HTTP SSL | This service implements the secure hypertext transfe... | | Manual | Local System |
| IMAPI CD-Burning ... | Manages CD recording using Image Mastering Applica... | | Manual | Local System |
| Indexing Service | Indexes contents and properties of files on local and ... | | Manual | Local System |
| Intel(R) Matrix Stor... | | Started | Automatic | Local System |
| IPSEC Services | Manages IP security policy and starts the ISAKMP/Oa... | Started | Automatic | Local System |
| Java Quick Starter | Prefetches JRE files for faster startup of Java applet... | Started | Automatic | Local System |
| Logical Disk Manager | Detects and monitors new hard disk drives and sends... | Started | Automatic | Local System |

**Figure 2: Extended View Windows Services (Local)**

The only way to prevent this from happening is by setting the startup mode of each of these services to "Disabled". However, disabled is not really disabled. A user with proper permissions can start a disabled service by simply changing its startup mode to Manual or Automatic thereby exploiting that service and use it for malicious activities.

Windows services are exploited by manipulating the service to run a command or access the file system to read or write a protected file. Since most services are run in the security context of the SYSTEM account, they usually have privileged access to most system functions. This makes them particularly interesting to attackers. By manipulating a service, an attacker can escalate his own privileges to do just about anything he wants to do. For example, Microsoft Security Bulletin MS02-006 addresses a buffer overflow in the SNMP service that would allow an attacker to remotely execute commands with the permissions of the SYSTEM account. Other exploits are less serious but may still use flaws in a service to allow other unauthorized actions. For example, there have been flaws in the SMTP service that allows a spammer to disguise their identity by relaying e-mail through a mail server.

The trick for the attacker is getting access to the service [16]. For most Internet services, this is simply a matter of connecting to the assigned TCP port. For other services, one must have local console access to be able to do any serious exploiting [16].

### 1.5 Windows Networking Protocols

Often criticized for their lack of integration with other vendors' operating systems, Microsoft adopted the TCP/IP as the de facto standard in all operating systems from Microsoft Windows NT 3.51 and forward, encompassing Windows 2000 Server and client versions, Windows XP Professional Server, Windows XP Professional client and Windows XP Enterprise editions [17]. The impetus behind Microsoft adopting the TCP/IP standard in all Windows operating systems are defined by networking expert Louis Columbus "is to accomplish a variety of tasks [17]." First, TCP/IP provides a solid foundation on which to build a connectivity strategy and drive the development of mixed or heterogeneous networks that include many different operating systems having TCP/IP in common as their shared method for communicating. Second, TCP/IP plays the critical role of integration of Windows 2000 workstations and servers with the Internet. Third, the customization of the TCP/IP command set by Microsoft for the Windows 2000 Series of operating systems specifically creates a differentiator that positions Microsoft effectively versus UNIX and NetWare as a viable alternative [14].

### 1.5.1 Why TCP/IP Is the De Facto Standard for Networking

Microsoft's' en masse adoption of TCP/IP as their de factor networking protocol was necessary due to the following reasons. First, the need for device independence at the router, hub, and switch level of networks forced the need for a standard soon after networking began to grow in popularity in government and education sectors. Second, the requirement of having a standardized addressing method was critical so what has turned into the IP address of systems could be recognized through the many different networks. This standardization applies to the definition of IP addressing, IP Address classes, and subnet masking conventions. The outgrowth of these standards spawned the creation of the Open Systems Interconnect (OSI) Model, shown in Figure 3 to create a standardized approach to solving the many intricacies of connecting heterogeneous systems with each other. In the case of security, the critical role of the OSI Model is shown in the role of the network validating IP addresses before they are passed to higher layers in the model where applications and data reside [18]."

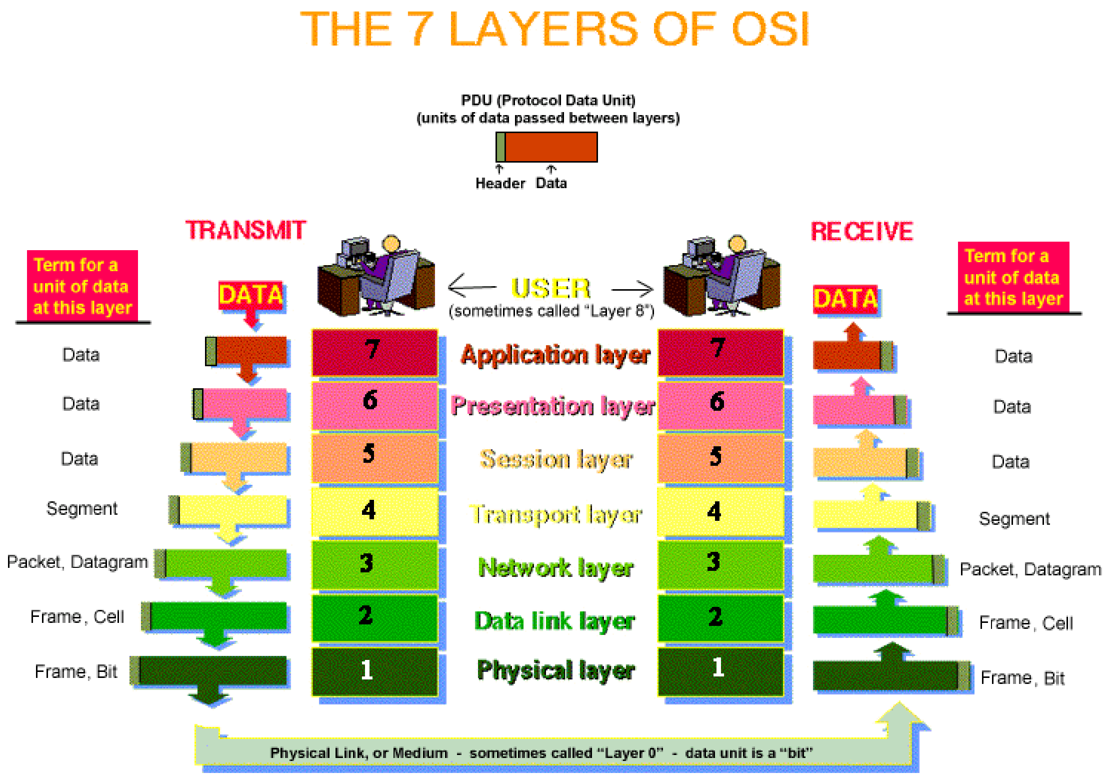


**Figure 3: The OSI Model [18]**

The layers of the OSI Model, includes the critical distinction of the physical and link layers, with one focusing on the transport of data from one system to another and the second set being the "logic" of communications between systems [18].

### 1.5.2 Integration Strategies using Microsoft Operating Systems

With the OSI Model defined and the TCP/IP command set created, networking between vastly different computer systems became possible. The decision for Microsoft to adopt the TCP/IP standard was critical not only for their growth, but for the growth of the entire networking industry [17]. With the breadth and depth of their customer base, and over 200 million desktop PCs running first MS-DOS and later Microsoft Windows, the need for more robust connectivity was apparent. Microsoft also had to respond to the requests from their enterprise, or large business, customers to integrate with Novell NetWare and many forms of the Unix operating systems.

Microsoft systems communicate using a standard implementation of TCP/IP that each installation of their operating systems could recognize and work with. Microsoft's initial release of their interpretation of the TCP/IP standard included a robust set of commands that made it possible to connect with many different types of operating systems. According to Columbus, Microsoft included "the essential TCP/IP commands including FTP (File Transfer Protocol), TFTP, Telnet, RCP, REXEC, RSH and LPR, in addition to DHCP and Domain Naming Services (DNS) [17].' Of all these, the Dynamic Host Configuration Protocol (DHCP) would prove to the most useful, especially with the growth of the Internet. DHCP activates and de-activates IP addresses on a timing or "lease" approach, and this has proven to be very useful for connecting for a specific period to targeted systems, and gets used extensively by Internet Service Providers. The net effect of having these TCP/IP commands in the baseline operating system was to make the Microsoft operating systems more open and capable of being integrated with than ever before.

### 1.5.3 Connecting with Novell

Novell's approach with NetWare had first been to use the MS-DOS and later Microsoft Windows operating systems as the foundation for their own peer-to-peer and later client/server networks on top of the Microsoft operating systems. Peer-to-peer refers to the ability one system to communicate directly with another, while client/server indicates there is the need for a server to coordinate network traffic and applications across a network [17]. Microsoft, with the introduction of Windows 2000 Server, realized that they could capture more revenue with a stronger integration strategy into the Novell NetWare environments their customers were grappling with. In response to this, Microsoft created Windows 2000 Services for NetWare. Realizing that the majority of their shared customers wanted to get access to file, print, and directory resources, Microsoft created specific options in Windows 2000 Servers for NetWare for extensive server-level support in those three dominant areas. Microsoft also created the NWLink application for making it possible for client-level PCs running NetWare to access files on Microsoft-based servers. Columbus devotes Chapter 10 in the referenced book to Novell NetWare integration strategies and states, "Microsoft developed a complimentary set of services to enable each platform to integrate with the other and migrate from NetWare-based services. Two services are shipped together in one package and include File and Print Services for NetWare (FPNW) and Directory Service Manager for NetWare (DSMN) [17]."

The intent by Microsoft for offering those tools was to eventually have Novell NetWare users migrate from that network operating system to Microsoft Windows 2000 Server and later operating systems from Microsoft, and the strategy worked. It was successful, because 30% of users eventually migrated to Microsoft platforms.

The heart of any integration strategy across diverse network operating systems starts with an appreciation of the need for standards to ensure consistency in everything from the role of hardware to connect the systems through the logic of how messages are communicated. The routers, hubs, and switches used in the creation of a network align in their role to the levels of the OSI Model, and provide the necessary routing of network packets.  These network packets create the foundation of how systems communicate, relying on the TCP/IP networking protocol to provide the foundation for interoperability. In the case of Microsoft adopting the TCP/IP standard and their role in networking, they did not set the market direction, they responded to it and further validated the TCP/IP standard as the basis of system integration and connectivity.

# Chapter 2: A History and Overview of Windows Policies and Products

In 1998 Bill Gates was quoted in the Seattle Weekly as saying, “Microsoft looks at new ideas, they don't evaluate whether the idea will move the industry forward, they ask, how will it help us sell more copies of Windows [19]?” In the following sections, Microsoft’s business practices, policies, and products will be analyzed.

## 2.1 Microsoft’s Business Model

“Microsoft generates revenue from licensing software. It operates on a factory model of software sales: keep availability of bits limited to increase value and produce more bits for more people to increase revenue. In a traditional factory environment, products are made of limited resources and are reproduced and sold. By simple laws of economics if supply of the product is somewhat limited and a demand exists, a profit can be made. Software, on the other hand, is a limitless resource. The sale value is created by the artificial limit placed on copying bits by corporations such as Microsoft. While the resource limit cannot be avoided in the physical world, this artificial limit placed on Microsoft software helps only Microsoft at the expense of the customer. The customer gets no better software by paying more to Microsoft [23].”

### 2.11 Acquiring Rival Companies

“One of Microsoft’s strategies in obtaining market dominance was through acquisition. From their very beginning in 1975 they have taken other people's ideas and repackaged them as their own, either by cloning the software, or purchasing and dismantling rival businesses, modifying and re-branding the software as though it was their own in the first place [23].” For example, Microsoft's first acquisition was

Forethought on June 29, 1987, which was founded in Sunnyvale, California in 1983 [24]. Before it's acquisition, Forethought originally developed a presentation program for the Macintosh computer. After Microsoft bought Forethought, it became Microsoft's Graphics Business Unit, which continued to further develop the software into what is now called Microsoft PowerPoint [57]. On December 31, 1997, Microsoft acquired Hotmail for $500 million, its largest acquisition at the time, and integrated Hotmail into its MSN group of services. Hotmail was founded by Jack Smith and Sabeer Bhatia and launched in July of 1996. Hotmail was one of the first free webmail services and was funded by the venture capital firm Draper Fisher Jurvetson. After it was acquired by Microsoft, it was re-branded as "MSN Hotmail". The current version, "Windows Live Hotmail", was officially announced in 2005 and released worldwide in 2007. Microsoft acquired Seattle-based Visio Corporation on January 7, 2000 for $1.375 billion. Visio, a software company, was founded in 1990 as Axon Corporation, and its primary product was diagramming application software, Visio, which was integrated into Microsoft's product line as Microsoft Visio after its acquisition. Figure 4 is comprised a list of some of the companies acquired by Microsoft. The consequence of the practices of Microsoft aforementioned creates a lack of true innovation. The evolution of Microsoft’s software is nothing short of glacial. What is not seen is anything radical or revolutionary, because the influence of Microsoft on society, and more generally the increase in commercialization, is responsible for the lack of any genuine innovation in many years. While competition in computer hardware drives devices faster, smaller, and cheaper the lack of comparable competition in software makes programs ever slower, bloated, and expensive. It is estimated that Microsoft has acquired six companies per year since 1987 after its initial acquisition of Forethought [44].

**Table 5: A Few Companies Acquired By Microsoft [44]**

| Date | Company | Business | Country | Value |
|---|---|---|---|---|
| March 31, 2009 | Consumers Software | Software | Canada | $20,500,000 |
| June 29, 1992 | Fox Software | Database Software | U.S. | $174,000,000 |
| February 3, 1997 | NetCarta | Internet Software | U.S. | $20,000,000 |
| April 30, 1997 | WebTV Networks | ISP | U.S. | $425,000,000 |
| December 31, 1997 | Hotmail | Internet Software | U.S. | $500,000,000 |
| January 7, 2000 | Visio Corporation | Drawing Software | U.S. | $1,375,000,000 |
| September 13, 2000 | MongoMusic | Online music SE | U.S. | $65,000,000 |
| March 17, 2001 | Vacationspot | ISP | U.S. | $70,850,000 |
| July 12, 2002 | Navision | Software program | Denmark | $1,300,000,000 |
| December 16, 2004 | GIANT | Anti-spyware | U.S. | Unknown |
| August 31, 2005 | Frontbridge Tech | E-mail protection | U.S. | Unknown |
| August 13, 2007 | AdECN | Security | U.S. | Unknown |
| August 13, 2007 | aQuantive | Digital Marketing | U.S. | $6,333,000,000 |
| March 19, 2008 | Komoku | Rootkit Security | U.S. | $5,000,000 |

### 2.12 The Microsoft Monopoly (CyberInsecurity)

The acquisition of businesses at the pace of six per year has enabled Microsoft to assume market dominance, especially with software. The goal is very simple; to be the exclusive computing platform of all electronic devices, including home appliances and PCs, gaming consoles, workstations and servers (with Microsoft Windows), and mobile devices (with Microsoft Windows CE). According to the report *CyberInsecrity*, Microsoft's efforts to design its software in evermore complex ways so as to illegally shut out efforts by others to interoperate or compete with their products has succeeded [27]. The report continues to say that because of this monopoly, these products are used by nearly everyone and it is riddled with flaws. The CCIA (Computer & Communication Industry Association) warned of the security dangers posed by software monopolies during the US antitrust proceeding against Microsoft in the mid and late 1990's, and later urged the European Union to take measures to avoid a software "monoculture" that each day becomes more susceptible to computer viruses, Trojan Horses and other digital pathogens [27]. The CCIA is an advocacy organization based in Washington, D.C that represents large players in the computer, Internet, information technology, and telecommunications industries. According to their site, the CCIA "promotes open markets, open systems, open networks, and full, fair, and open competition [27]." The CCIA and the authors of CyberInsecurity, Dan Geer, Rebecca Bace, Peter Gutmann, Perry Metzger, John S. Quarterman, Charles Pfleeger, and Bruce Schneier, believe that the presence of a single, dominant operating system in the hands of nearly all end users is inherently dangerous, and in their final analysis, Microsoft was considered a threat to national security. It should also be mentioned that as a result of this report, the primary author, Dan Geer was fired from his job as the CTO of a company called Stake because of this report and the fact that the company he worked for happened to be one of Microsoft's largest clients.

## 2.13 Microsoft Software Backwards Incompatibilty

“Every investment in new software requires consideration of older software and data formats which will remain in use. For this reason software users typically prefer new versions that are completely backwards compatible with older versions. By definition, backwards compatibility means new platforms will run software written for older versions of the platform and that applications will be able to read data formats from prior versions [23].” Microsoft has made a habit of breaking compatibility with their software. Many of their Windows products will not run software that worked on earlier versions of the operating system, often forcing users to upgrade additional software if they upgrade Windows. “For example, Microsoft Windows XP contains changes to the Win32 API which break many applications written for Windows 2000, NT, and other versions [23].” “Microsoft Exchange Server 2000 will not run on Microsoft Windows Server 2003. Internet Information Services 5.0 will not run on Windows Server 2003. SQL Server 2000 will only run on Windows Server 2003 with Service Pack 3 installed [23].”

In 2007, Microsoft lost a lawsuit on the issue of incompatibilities with its software in Des Moines, Iowa. The incompatibilities were between Microsoft's Windows 386, 3.*x* and Windows 95 products and DR-DOS. This was an operating system developed by Digital Research, and later acquired by Novell, which was 100 per cent compatible with Microsoft's MS-DOS. The citizens of Iowa, pursued a consumer class action lawsuit against Microsoft, in which Microsoft executives had to take the stand in Des Moines defending the company's conduct. In the end, Microsoft had to pay the citizens of Iowa $180 million to settle the class-action lawsuit that proved the company abused its monopoly position to overcharge PC shoppers. The deal covered consumers who purchased Microsoft's PC software between 1994 and 2006, and who, prosecutors said, spent $453 million more than they should have [28]. Under the agreement, Microsoft had to refund users $16 for each copy of Windows or MS-DOS, $25 for Excel, and $29 for Office. Users of Word, Works, and Home Essentials got $10 per copy of the software [28].

The lawsuit might have prompted Microsoft to make available for users the software compatibility pack for the newest version of Windows Office 2007 that is being used today. Without the software compatibility pack, it would be impossible to read a DOCX (Word 2007) file format if opened in Word 2003 or previous versions of Office. However, there are still some incompatibility problems with this software. For example, the Word 2007 equation editor is incompatible with that of Word 2003 and previous versions, and when converting DOCX files to DOC files, equations are rendered as graphics. Consequently, Word 2007 cannot be used for any publishing, file-sharing and collaborative endeavor in any mathematics-based fields, including science and technology, in which users may have earlier versions of Word [29]. For reasons unknown, Excel and PowerPoint 2007 retain the old equation format, meaning that users cannot move equations between Word and the other programs even though they are the same version. Many publishers do not accept submissions in Word 2007; for example, academic publishers have informed Microsoft that this severely impairs Word 2007's usability for scholarly publishing [29].

## 2.2 Microsoft’s Products

It is a common misconception that Microsoft is ahead because their products are superior. Microsoft's products are generally not superior. As an example, Windows is more bloated, much less stable, less secure, much more expensive, and lacking much of the capabilities of Linux and Mac OSX, two of its competing operating systems. The reason that Microsoft is ahead is because their marketing is superior and because they leverage their existing market share to keep consumers locked into Microsoft specific solutions. In terms of Security, Microsoft's products are notorious for their security holes. Security holes in Internet Explorer and Microsoft Operating Systems have been widely publicized for years and unfortunately are now accepted as a common occurrence whenever they are announced. The public has become largely desensitized to new security holes whenever they are announced.

### 2.21 Microsoft's Advertising Strategies

Whenever marketing a product to the masses, especially when in competition with another organization selling a similar product, the organization must advertise their product as being the better one. One of the brilliant marketing strategies that Microsoft imposes on consumers is to advertise new products against old products in the same product line. With each new release of an application, Microsoft usually promotes features of stability and security, especially in their Windows product line. These are of key concerns to most business and home consumers. In promoting their product stability, Microsoft is countering two other product groups: competitors and previous versions of their own products. Microsoft is continually criticized for having relatively insecure, unstable products. Therefore with each new release they have to advertise increased stability and security against their own older products. Microsoft also tends to be misleading in its advertisement to the consumers. For example, according to an article written by Stephen Wittford in 2003, The Advertising Standards Authority (ASA) of SA ordered an ad by Microsoft to be pulled because it claimed that the ad was both unsubstantiated and misleading. The ad depicted a dodo, a woolly mammoth, saber tooth tiger and a hacker. The ad claimed that not everyone benefits from Microsoft software and that with it; a customer's data couldn't be safer even if it was kept in a safe. It was published in the November issues of ITWeb Brainstorm and Time Magazine [30]. A freelance journalist named Richard Clarke debunked the ads claim, saying that Microsoft software was littered with vulnerabilities. Microsoft was asked by the ASA to provide information, substantiated by an independent, credible expert, on the degree of security of its software in accordance with Code of Advertising Practices. Microsoft was also asked to defend the ad against Clarke's claim that the ad was misleading. Microsoft submitted documentation to substantiate its claims about the security of the software and said the advert was not designed to mislead the consumer, but was merely a tongue in cheek dramatization that the software would threaten the survival of hackers. After reviewing both parties' submissions, the ASA ruled that Microsoft's claims about the security of its software were unsubstantiated, as it had not been evaluated by an independent entity. The ASA ruling said that because the claim was unsubstantiated, it was therefore misleading and ordered the ad to be withdrawn [30].

## 2.22 Internet Explorer

Internet Explorer is Microsoft’s free World Wide Web browser included in every version of Microsoft Windows. Being included in the majority of personal computers when they're sold, most people who are new to the Internet are introduced to the Internet with Internet Explorer as their first web browser.

All versions of Internet Explorer are vulnerable to one or more security issues. Most of these issues are the result of design flaws, while some are bugs and others the result of third parties [31].

Internet Explorer (IE) has at least one major security design flaw. Beyond mere inclusion with Microsoft Windows, Internet Explorer is integrated into the operating system [31]. Microsoft claims without IE's core components Windows cannot fully function, which was the argument Microsoft used during its antitrust case. IE is integrated into the file system browser and other applications. Therefore no capability is given to completely uninstall it, even for servers. The low level integration also results in web browser security issues creating operating system vulnerabilities, sometimes even if the browser is never launched directly by the user. Many security bugs in IE could be exploited to give an attacker complete control over Windows. Microsoft’s security bulletin highlights many of the vulnerabilities that they had to patch for IE among other Operating System patches. However, security patches for Internet Explorer also sometimes cause peripheral problems in the operating system. Figure 5 is an overview of IE’s architecture.

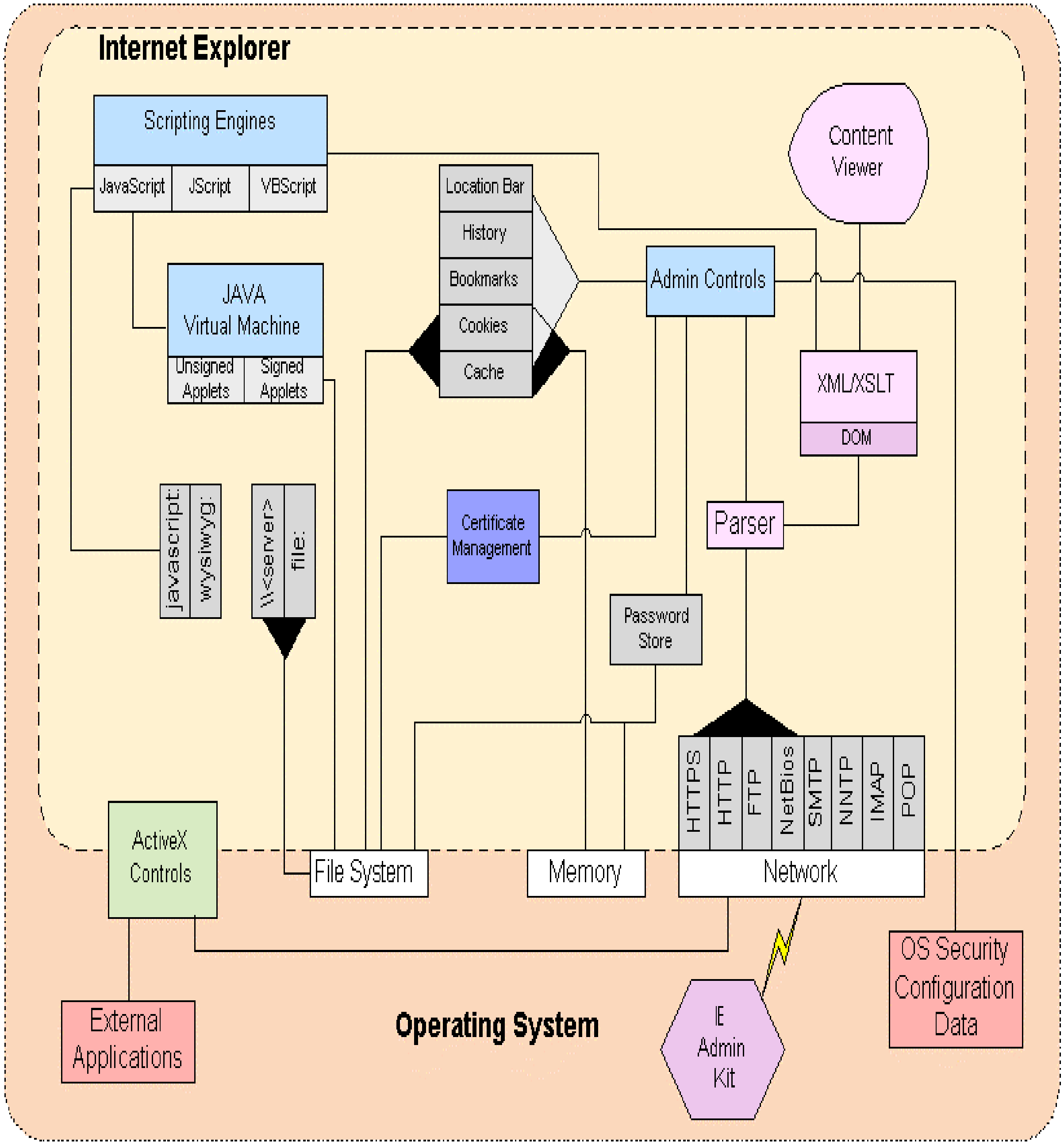


**Figure 4: An Overview of the Internet Explorer Architecture [31]**

## Chapter 3: Monitoring Network Security

Network security is a fundamental enabling technology of the Internet [32]. The limits of security are the limits of the Internet, and no business or person is without these security needs. Since the Internet is a network of networks, network security is extremely crucial and important. Network security is a chess match between the defender and the attacker, and the attackers have all the advantages. First, the defender has to defend against every possible attack, while the attacker only has to find one weakness. Second, the immense complexity of modern networks makes them impossible to properly secure without the proper tools and knowledge. And third, skilled attackers can encapsulate their attacks in software, allowing people with no skill to use them [40].

Network monitoring implies a series of sensors in and around the network [49]. Every firewall produces a continuous stream of audit messages [50]. So does every router and server. IDSs (Intrusion Detection Systems) send messages when they notice some anomalous behavior. Every other security product generates alarms in some way. But these sensors by themselves do not offer security. It should be assumed that the attacker is in full possession of the specifications for these sensors, is well aware of their deficiencies, and has tailored their attack accordingly. They may even have passwords that let them masquerade as a legitimate user [52].

The first step is intelligent alert. Network attacks can be subtle, and much depends on context. Software can filter the tens of megabytes of audit information a medium-sized network can generate in a day, but software is too easy for an attacker to fool [39]. Intelligent alert requires people to:

- Analyze what the software finds suspicious;

- Delve deeper into suspicious events, determining what is really going on;
- Separate false alarms from real attacks;
- Understand context.

By itself, an alert is only marginally useful. More important is to know how to respond. This is the second step of good network monitoring. Network devices produce megabytes of audit information daily. Automatic search tools sift through those megabytes, looking for signs of attacks. Examining those attacks, understanding what they mean and determining how to respond dictates the overall security of the network as a whole.

In this thesis, Network Honey was the target network used to investigate the activities of the attackers, by monitoring network security. In the next section the tools used to investigate the activities of attackers in the Network Honey Design will be explained.

## 3.1 Network Honey

The goal of Network Honey was to test Microsoft Window's networking environment against malicious threats, externally, to determine its viability. The centerpiece, or foundation of Network Honey is Active Directory. Active Directory, a technology originally created by Microsoft in 1996 and first used with Windows 2000 is Microsoft's network operating system (NOS) directory built on top of Windows 2000 and Windows Server 2003 and used on Microsoft Windows based computers and servers to store information and data about networks and domains [33]. This information is comprised of data about users and resources, and it allows access and manipulation of those resources. Active Directory is a way to manage all elements of the network, including computers, groups, users, domains, security policies, and any type of user-defined objects. “Active Directory is built around Domain Name System (DNS) and Lightweight Directory Access Protocol (LDAP) [36].” It is built around DNS, because it is the standard on the Internet, and LDAP because most vendors support it. Active Directory clients use DNS and LDAP to locate and access any type of resource on the network [37].

### 3.1.1 Network Honey Tools

In order to efficiently and effectively monitor network security and take countermeasures once the network or systems on the network are compromised requires the proper tools to be in place to bring that task to fruition. Network Honey is comprised of Firewalls, Antivirus, Syslog Server (Daemon), Vulnerability scanners, and IDS.

#### 3.1.1a Intrusion Detection Systems

Intrusion detection systems first (IDS) first appeared in the early 1980s, however, the products that are widely used today came onto the market in the mid to late 1990s in conjunction with the explosion of the Internet [49]. IDS are designed to monitor activities for the purpose of finding security violations. An IDS is not designed to stop intrusions but to alert when an intrusion is attempted so that counter measures can be taken to thwart the attempt or to improve clean up efforts if an attack has been successful.

IDS fall into two categories: network based intrusion detection (NIDS), and host based intrusion detection (HIDS). It was determined that the use of both NIDS and HIDS in Network Honey better served the purpose for analyzing traffic. Network based intrusion detection works by analyzing network traffic. This method was chosen for Network Honey for a few reasons. First, because NIDS is passive in that it listens to activity on the network and analyzes with little to no performance or compatibility issues that could skew results. Second, it has a wide range of detection when monitoring malicious activities. NIDS analyzes all activity regardless if the attack was successful or not. Finally and most of all, because it is the case that the host on any network that deploys HIDS can be compromised, it provides extra security in terms of validating the security logs, because NIDS sensors run on a host separate from the target and is more impervious to tampering. As previously stated, HIDS runs in a monitored host, which allows for detailed logs, the ability to detect unknown attacks, and because it also provides substantially fewer false positives then NIDS. HIDS logging information was used during the reconnaissance process, whenever it was determined that more information or investigation was needed.

Snort was chosen for the implementation of the NIDS. Snort was chosen because it has similar performance and functionality when compared to commercial IDS solutions, and no other open source IDS comes close to Snort in functionality. Because of Snort’s popularity there was a great deal of support available for Snort to help aid in the initial configuration. Another equally important advantage of using Snort is that there are a large number add on products that can be used to expand its functionality. It should also be mentioned that there is not only a great deal of support for Snort in the open source world. A number of commercial packages have implemented interfaces that allow users to integrate signature files from Snort into these commercial products. This makes a strong statement about the reliability and quality of the signature files available for the Snort package.

Snort is a signature based IDS [53]. This means that Snort relies on a continually updated set of rules to detect new attacks. The rules come can be implemented from a number of sources, and the administrator of a Snort IDS also has the option to write their own rules if there is an attack that is not yet identified by available Snort rules. The Sourcefire Vulnerability Research Team (VRT) was the source chosen to provide real time updates for rules.

MySQL was chosen for the SQL database. Snort does not need SQL to run, but the alert management product for this implementation, ACID, needs alerts to be stored in a SQL database. Along with storing data in a SQL database, Snort also drops copies of all alerts to a human readable text file. This was useful for debugging and investigating alerts.

The final major item that was needed to have a complete solution was the alert manager previously mentioned before. Snort can handle alerts in a number of ways, including alerting to the console, sending e-mail, sending SMS messages, writing to a log file and logging to a database. Logging to a database allows a third party tool, such as ACID, to read the logs and generate reports. There are a number of analysis consoles available for Snort, however, ACID was chosen because it has the features that were needed, and has proven itself to be stable. The only disadvantage of ACID is that it is not currently being actively developed.

OSSEC was chosen for the HIDS as a solution. “OSSEC is an Open Source Host-based IDS that provides real time monitoring, detection, log analysis rootkit detection, and prevention of security breaches, delivering automated policy enforcement and incident response for servers, applications, and data [58].” This was a great complement to the firewalls that were deployed on Network Honey because it enabled the ability to develop proactive policies to stop hackers or authorized users with malicious intent from misusing systems. The configuration and setup of Snort with ACID and MySQL was done with the guidance of two valuable resources. First, on the Snort website, Kasey Efaw wrote a setup guide on how to configure Snort on Windows XP boxes, and Jeff Richards wrote a guide on setting up Snort with MySQL and ACID back in 2000. With a few modifications and updates, it worked flawlessly.

### 3.1.1b Firewalls

A firewall can be defined as “a system that is designed to prevent unauthorized access to private computers or networks [42].” It is a hardware or software device which filters network traffic based on a set of rules. The firewall examines each packet received and determines whether there is any characteristic of this data matching the rule set by which it is disallowed [50]. If there is a match (or lack thereof), a packet can be barred from passing on through the network. In short, users can protect their machines from unwanted (or malicious) network traffic. Firewalls control what gets in and comes out of the network.

The first thing a potential hacker wants is information on the target machine or network. Preventing them from getting this information is a huge step in securing an infrastructure [47]. A firewall does exactly this using the methods described above, disallowing the traffic that represents the prying eyes of a hacker.

To determine if data or access requests are allowed to pass the firewall (from or to the inside Network) the firewall has to be able to authenticate messages and the traffic has to be directed through the firewall. The first question is where to situate the firewall. Firewalls have traditionally been used to protect the perimeter of a network; that is, the connection between the company network and the Internet. In Network Honey, the deployment of a perimeter firewall, along with host based firewalls, was chosen.

Network Honey deploys the traditional firewall method, but it was essential to add the host based as well, because as company intranets grow larger, threats appear within firewall boundaries as well. The point of Network Honey was to simulate that type of environment to provide all available security vectors, and the choice of adding host based firewalls with the perimeter firewalls brought that to fruition. What a host-based firewall does is adds an extra layer of security around the systems, protecting the potential targets from hostile attacks and misconfigurations [56]. Instead of only having a single firewall at the perimeter, referred to as choke-point firewall, there can now be one on each server or end device [43]. If the system is penetrated, it won't be used as a stepping-stone for further attacks, and the presence of the attacker can be exposed [54].

The perimeter firewall was enabled on the Netgear router in Network Honey, in which security logs and alerts were automatically sent to the mail server (Microsoft Exchange Server) on the network. The host based firewall is a two way firewall from the Vendor ZoneAlarm. The results of these logs will be displayed in Chapter 5.

### 3.1.1c Antivirus

Antivirus software is computer software used to identify and remove computer viruses, as well as many other types of harmful computer software, collectively referred to as malware [46]. While the first antivirus software was designed exclusively to combat viruses, most modern antivirus software can protect against a wide range of malware, including worms, rootkits, and trojan horses [51].

Using antivirus software is a necessity. Not only must antivirus software be installed on all servers and workstations but virus definition files (DATs) must be constantly updated. When the environment (enterprise networks) is large, most companies purchase antivirus software that has a management component that allows automatic DAT updates and virus scans over the network as well as one that provides central administration features. This usually comes in the package of a server. For home users, the antivirus software is installed directly on the computer. It is important when shopping for antivirus software to ensure that it does not conflict with the firewall software already installed in the respective system. In Network Honey, Symantec Endpoint was the antivirus software chosen.

#### 3.1.1d Syslog Server Daemon

Another way to determine if a network intrusion has occurred or to determine the events which an intruder uses to gain unauthorized access to a system is through the use of a Syslog Server. In many organizations this event information is collected and compiled using the Syslog protocol, via the logger program, the Syslog system calls, and the Syslog daemon process. Syslog was designed to provide the ability to report system events. Kiwi Syslog Server was the product used to provide the services of a Syslog server on Network Honey. Kiwi Syslog Server receives syslog messages from network devices, and displays them in real-time. Actions are performed on received messages and messages are filtered by host name, host IP address, priority, message text or time of day. This proved to be a useful tool when collecting data for further analysis.

# Chapter4: Network Penetration-Network X

Penetration testing is security testing in which an evaluator attempts to circumvent the security features of a system based on their understanding of the system design and implementation [48]. The purpose of penetration testing is to identify methods of gaining access to a system by using common tools and techniques used by attackers [38]. Penetration testing can be an invaluable technique to any organization's information security program. At a minimum, it may slow the organization's networks response time due to network scanning and vulnerability scanning. Furthermore, the possibility exists that systems may be damaged in the course of penetration testing and may be rendered inoperable, even though the organization benefits in knowing that an intruder could have rendered the system inoperable. Since penetration testing is designed to simulate an attack and use tools and techniques that may be restricted by law, federal regulations, and organizational policy, it is imperative to get formal permission for conducting penetration testing prior to starting [48].

Penetration testing is considered to be ethical hacking. As stated in the previous paragraph, without the permission of the target organization, any attempts to penetrate an organizations network can result in incarceration. The author of this thesis was given permission to perform the penetration tests and as simple protocol in the penetration world, the name of the organization is never revealed, hence Company X.

With the permission of Company X to penetrate Network X, comes a set of agreed upon rules of engagement. To conduct these tests, the author was given the following:

- Specific IP addresses/ranges to be tested

- Any restricted hosts (i.e., hosts, systems, subnets, not to be tested)
- A list of acceptable testing techniques (e.g. social engineering, DoS, etc.) and tools (password crackers, network sniffers or scanners, etc.)
- Times when testing was to be conducted (e.g., during business hours, after business hours, etc.)
- Identification of a finite period for testing
- IP addresses of the machines from which penetration testing will be conducted so that administrators can differentiate the legitimate penetration testing attacks from actual malicious attacks
- Point of Contact information

## 4.1 Assumptions and Requirements

Penetration testing can be overt or covert. These two types of penetration testing are commonly referred to as limited knowledge and zero knowledge. Limited knowledge involves performing a penetration test with the knowledge and consent of the organization's IT staff, as well as basic information that a hacker would have come up with anyway. Zero knowledge involves performing a penetration test without the knowledge of the organization's IT staff but with full knowledge and permission of the upper management [53].

The agreement with Company X was to perform limited knowledge over zero knowledge penetration testing, because of time constraints and because it was the least expensive of the two. As previously stated, the author was not provided with any real information about the target environment other than targeted IP address/ranges, three restricted hosts (Human resources server, president and vice president computers), all other information was required covertly before the attack was initiated.

The time agreed upon for the initiation of the tests was during the three least busy days of the week and anytime after hours. All techniques, were authorized to be used in the penetration tests.

## 4.2 Description of the Methodology

“The methodology of a penetration test is separated into four phases [38].” The first phase in conducting a penetration test is planning. As stated in the previous section, before the testing begins, goals, timetables, and parameters must be set. In the planning phase, rules are identified, management approval is finalized, and the testing goals are set. The planning phase sets the groundwork for a successful penetration test. No actual testing occurred in the planning phase.

The second phase is the discovery phase. This phase consists of continuous information gathering. This is where the author becomes an illegal hacker. The discovery phase starts the actual testing. Network scanning (port scanning), briefly mentioned at the beginning of this chapter as an agreed upon tactic was used to identify potential targets. In addition to port scanning, other techniques were commonly used to gather information on the targeted network and are as follows:

- Domain Name System (DNS) interrogation
- Search of the Network X's web server(s) for information
- Search of the organization’s Lightweight Directory Access Protocol server(s) (LDAP) for information

- Telnet and FTP hijacking
- Banner Grabbing
- Trace Routes

The second part of the discovery phase is vulnerability analysis. During this phase, services, applications, and operating systems of scanned hosts are compared against vulnerability databases (for vulnerability scanners this process is automatic). The author chose to use public databases as well to identify vulnerabilities manually. The manual process helps to identify new or obscure vulnerabilities.

The third phase of a penetration test is the attack phase. Executing an attack is the heart of the penetration test. This is where previously identified potential vulnerabilities are verified by attempting to exploit them. If an attack is successful, the vulnerability is verified and safeguards are identified to mitigate the associated security exposure. Frequently, exploits that are executed during attack execution can grant either the minimum or the maximum level of access [41]. During the attack phase, the author gained maximum access to some hosts and whenever minimum access was gained, it resulted in learning more about the Network X and its potential vulnerabilities. In either case, additional analysis and testing was required to determine the true level of risk for the network.

The fourth phase of a penetration test is the reporting phase. Although it is considered the fourth phase, actually the reporting phase occurs simultaneously with the other three phases of the penetration test. In the planning phase, rules of engagement, test plans and written permission are developed. In the discovery and attack phase, written logs are usually kept and periodic reports are made to system administrators and/or management, as appropriate. Generally, at the end of the test an overall testing report is developed to describe the identified vulnerabilities, provide a risk rating, and to give guidance on the mitigation of the discovered weaknesses. In the next chapter, the results of those tests, reports, and recommendations will be further explained.

### 4.3 Developing Attacks for The Penetration Test

There was a mutual agreement with the company to not share any detailed information about their network other than the techniques used to gather information on the targeted network, the attacks, and the results in summary. In this section, the author is going to discuss the attack methods, research and development process, and the kinds of attacks used in the penetration test.

#### 4.31 Attack Method

Several stages of work were involved for each attack method included in the attack phase of the penetration test. Each attack required development, analysis, and documentation. The following list outlines the steps that were taken in developing Network X attacks for the evaluation:

- Researched or invented the attack
- Modified the attack to work when blocked or denied by present security measures in Network X
- Attempted to make the attack stealth
- Defined a procedure to verify attack success
- Defined a procedure to cleanup after the attack
- Documented the attack

### 4.3.2 Attack Research and Development

Many of the attacks were obtained from public sources on the Internet. Web sites maintained by organizations, such as NTBugtraq, CERT, ISS, Rootshell, Hackers-Black-Book, and Insecure.org, post announcements concerning recent vulnerabilities and attacks against the Microsoft Windows operating systems, and tools and resources (software) available to enhance or aid in attack modifications [45]. They also archive information about older attacks. Sometimes, they provide source code that exploits known vulnerabilities and also instructions on how to execute attacks. However, even with instructions and source code, it frequently took a significant amount of work to get an attack to function properly for the attack. Each attack for the evaluation was researched and was either downloaded from the Internet, modified and or created based on known vulnerabilities in Microsoft Windows Operating Systems.

### 4.3.3 Types of Attacks Used

There are generally annual reports generated to analyze, or evaluate threats and or potential hazards that contribute to much of the financial losses due to cybercrime. "IronPort Systems, a leading Internet gateway security company, reports that more than 50 percent of corporate desktops are infected worldwide." This does not include small business, local, state, and federal governments, the military, schools and universities, or the average home user. The afflicted are infected by various forms of malware and are as follows:

- Rootkits
- Trojan Horses
- Worms

- Virus
- Keyloggers

Each example of malware described above represent the most common form of malware hackers use to gain access to systems. These were the attacks used in this thesis. The following paragraphs will explain each malware used in the penetration test.

“A rootkit is a piece of software that attaches itself to the core operating system in order to bypass system security restrictions [8].” Every operating system relies on application program interfaces (APIs) to function. A call to open a file is an API call. A rootkit allows these APIs to be manipulated. Thus when the operating system requests a particular file, the rootkit could return any other data object. This level of control is almost impossible to counteract. Rootkits can disable any desktop-based security software. They can leave backdoors so that once a system is compromised, the hacker can remain stealth. They are extremely difficult to detect, and they are often designed to preserve and reinstall themselves. For the penetration test the author downloaded most of the rootkits from various sites in the Hackers Black Book. Another great resource that aided in the deployment of the rootkits was the book *Rootkits: Subverting The Windows Kernel* by Greg Hoglund and Jamie Butler.

A Trojan Horse is a form of malware that infects a machine by posing as another harmless piece of software [44]. As an example, a Trojan could be delivered by a website that offers enticing content (such as a music video) which requires a special video codec plug-in to Windows Media Player. When the user downloads the codec plug-in they also get a Trojan which silently installs itself on the targeted PC. To avoid detection, a Trojan will often not contain any harmful code. Instead, it will install itself and then load malicious code from a remote Web server, sometimes using network ports other than Port 80 (the standard HTTP port). Trojans are different from viruses or worms in that they do not propagate on their own, they require "social engineering" to trick the user into running them. However, the rapidly expanding universe of Web content and applications continues to create new opportunities for clever Trojan horse programs. For the penetration test, the author downloaded the code to create a few Trojans horses from Hack Forums.

A worm is a form of malware that propagates itself. Several worms (such as "Slammer") have made front page news in recent years by flooding networks with traffic and causing widespread outages. A worm usually takes advantage of a security flaw to install itself on a host PC. Once installed, it then scans the network—looking for other machines that have the same security flaw. Using this method, a single worm can create large-scale propagation in a matter of hours. The jury is still out as to how damaging the Conficker worm recently deployed will be in the next couple of months of this year. Because worms can be very damaging to a network, the author first downloaded and tested the worm in a test bed network before deploying it.

A virus is a hostile piece of code that replicates by inserting copies of itself into other code or documents. According to ICSA Labs (a security industry consortium, formerly known as the International Computer Security Association), more than 90 percent of all viruses spread via email [44]. Email-borne viruses have an attachment that may pose as a legitimate file, but actually contains a harmful executable. One of the more clever viruses propagated in the form of a password protected .zip file, making it very difficult for traditional anti-virus software to detect. The email was designed in a manner that enticed the end-user to enter a password and thereby infect their machine. Once infected, a machine can be used for any number of purposes including acting as an SMTP email server (which can be used to send out more copies of the virus—and then later to send out spam). These infected computers are often organized into groups, called botnets, and are a key tool in the delivery of malware. One of the greatest resources used to aid in the deployment of the viruses used in the network penetration test was a book from Peter Szor called *The Art of Computer Virus Research and Defense*, along with downloads from sites from The Hackers Black Book.

A keylogger monitors the key strokes and system events of an infected PC. Keyloggers can be combined with sophisticated logic to perform tasks such as looking for the address of an online bank, recording the username and password, and then transmitting this information back to a rogue server —which in turn can transfer funds from the affected user [56]. Keyloggers can also be used to harvest sensitive corporate information. Keyloggers that were used for the penetration test were downloaded from torrent sites.

# Chapter 5 Evaluation of Windows Network Security

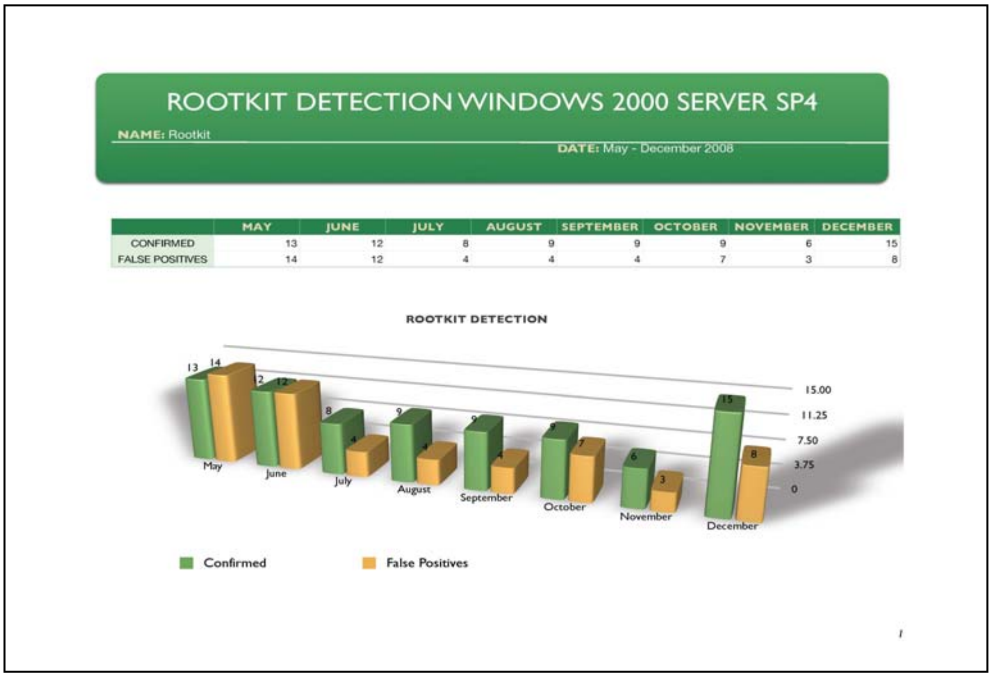
In this chapter, the results of the network monitoring of Network Honey and the results of the penetration tests of Network X will be further examined. Each section will cover a particular Microsoft OS, with the first branch of that section focusing on the Network Monitoring results and the second branch focuses on the network penetration results. All tests were performed over a six month period, and the malware that was either monitored in Network Honey or used for attack mechanisms against Network X were as follows: Rootkits, Trojans, Worms, Viruses, and Keyloggers. In Network Honey results, the data will be divided into malware that was confirmed, and false positives compiled for each month over a six-month period. In Network X, the penetration test data will be divided into successful attacks and failed attacks compiled each month over a six-month period. In Section 1, the results will be displayed for Microsoft Windows Server 2000 with service pack 4. Section 2 will cover the results for Microsoft Windows 2003 Server with service pack 2. Section 3 will cover the results for Microsoft Windows XP with service pack 3. Section 4 will cover the results for Microsoft Windows Vista with service pack 1.

## 5.1 Windows Server 2000 Service Pack 4

In the following sections, the results for Windows Server 2000 with service pack 4 in Network Honey and Network X will be displayed and explained.

### 5.1.1 Discussion of Results Network Honey

As discussed in Chapter 3, there were a variety of tools that were deployed to monitor the techniques of hackers and countermeasures taken to see how effective Microsoft Windows Server 2000 SP4 would hold up against constant attacks over a six month period. Microsoft Windows Server 2000 SP4 remains in production in many corporations, small businesses, local, state, and federal governments, and universities and schools throughout the U.S. As previously mentioned in the first chapter many of these entities are defined under critical infrastructures. As a consequence of this, it was important that this server be deployed and tested in Network Honey. Figure 5 shows the number of confirmed rootkits detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 75 rootkits detected and a total of 56 false positives on Windows Server 2000 SP4. The month with the highest level of rootkits detected was in December with 15, and the lowest level of rootkits detected was November with 6. The month with the highest number of false positives was May with 14 and the month with the lowest number of false positives was November with 3.



ROOTKIT DETECTION WINDOWS 2000 SERVER SP4

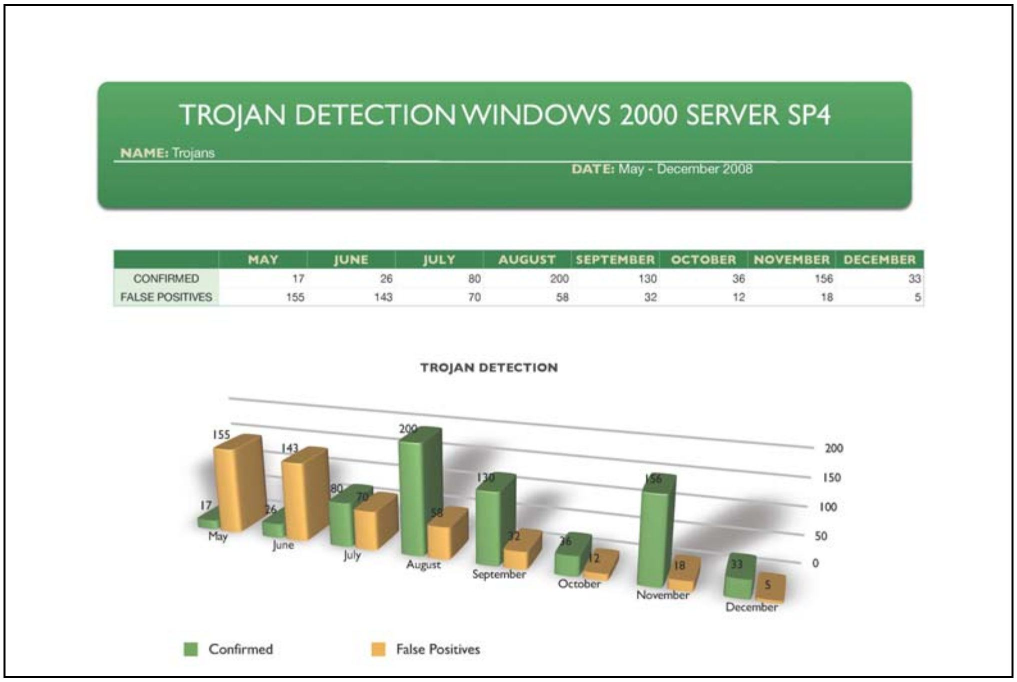
NAME: Rootkit

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 13 | 12 | 8 | 9 | 9 | 9 | 6 | 15 |
| FALSE POSITIVES | 14 | 12 | 4 | 4 | 4 | 7 | 3 | 8 |

**Figure 5: Rootkit Detection Results Windows Server 2000 SP4**

Figure 6 shows the number of confirmed trojans detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 678 trojans detected and a total of 493 false positives on Windows Server 2000 SP4. The month with the highest level of trojans detected was in August with 200, and the lowest level of trojans detected was May with 17. The month with the highest number of false positives was May with 155 and the month with the lowest number of false positives was December with 5.



TROJAN DETECTION WINDOWS 2000 SERVER SP4

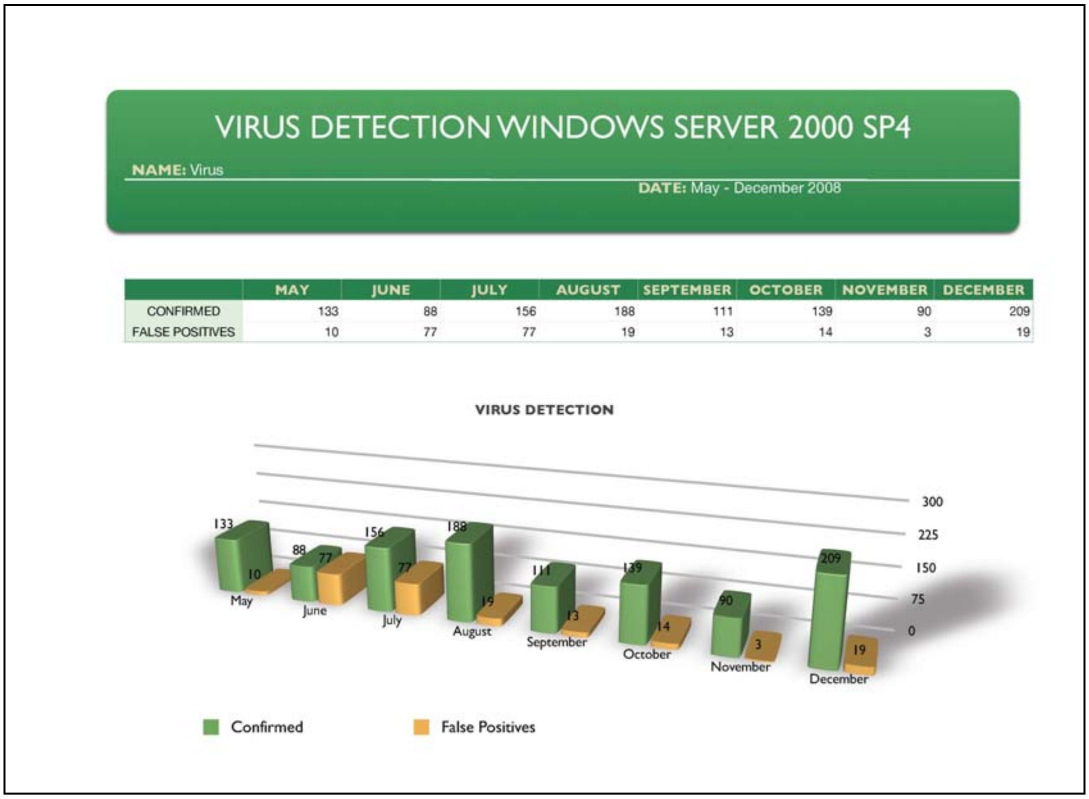
NAME: Trojans

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 17 | 26 | 80 | 200 | 130 | 36 | 156 | 33 |
| FALSE POSITIVES | 155 | 143 | 70 | 58 | 32 | 12 | 18 | 5 |

**Figure 6: Trojan Detection Results Windows Server 2000 SP4**

Figure 7 shows the number of confirmed viruses detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 605 viruses detected and a total of 110 false positives on Windows Server 2000 SP4. The month with the highest level of viruses detected was in May with 110, and the lowest level of viruses detected was July and August with 45. The month with the highest number of false positives was December with 22 and the month with the lowest number of false positives was November with 8.



VIRUS DETECTION WINDOWS SERVER 2000 SP4

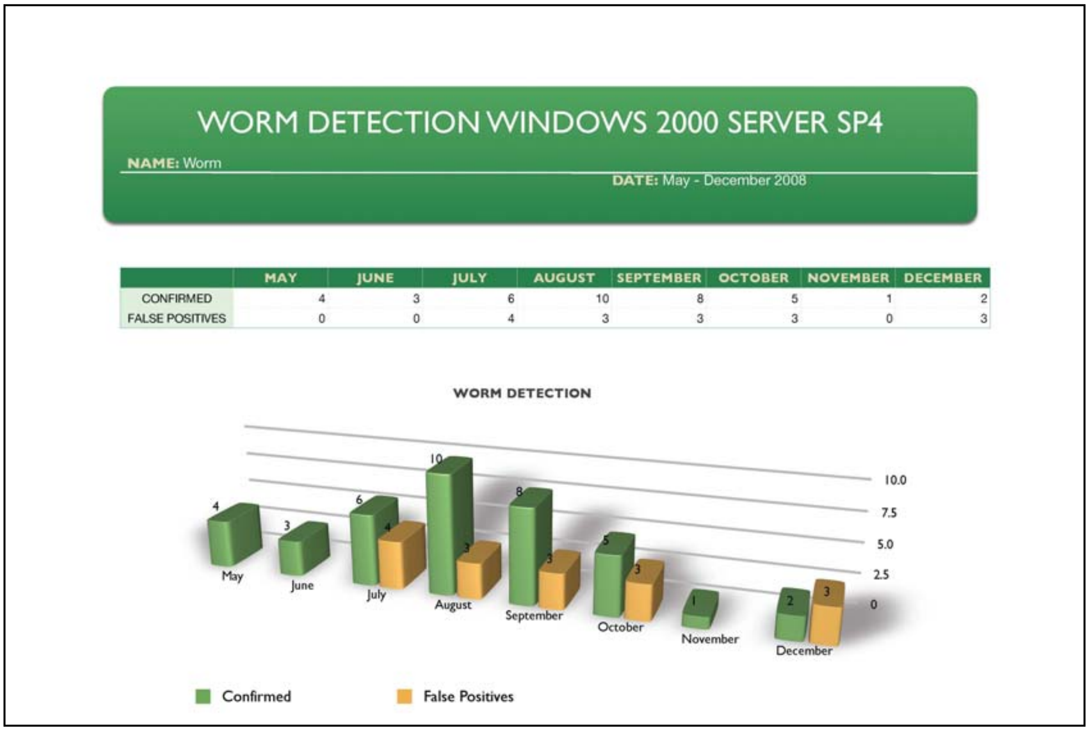
**NAME:** Virus

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 133 | 88 | 156 | 188 | 111 | 139 | 90 | 209 |
| FALSE POSITIVES | 10 | 77 | 77 | 19 | 13 | 14 | 3 | 19 |

**Figure 7: Virus Detection Results Windows Server 2000 SP4**

Figure 8 shows the number of confirmed worms detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 39 worms detected and a total of 16 false positives on Windows Server 2000 SP4. The month with the highest level of worms detected was in August with 10, and the lowest level of worms detected was November with 1. The month with the highest number of false positives was July with 4 and the month with the lowest number of false positives was May, June, and November with 0.



WORM DETECTION WINDOWS 2000 SERVER SP4

NAME: Worm

DATE: May - December 2008

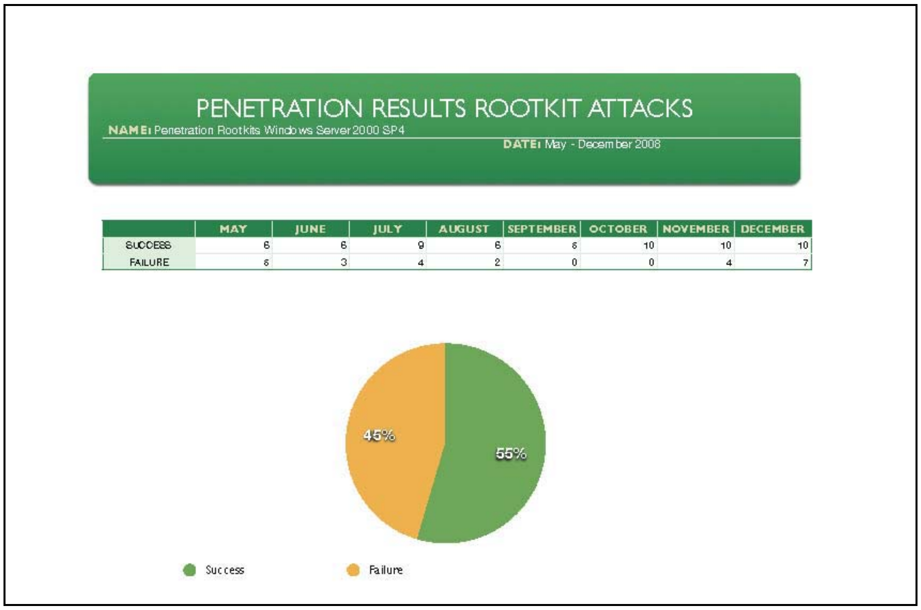
| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 4 | 3 | 6 | 10 | 8 | 5 | 1 | 2 |
| FALSE POSITIVES | 0 | 0 | 4 | 3 | 3 | 3 | 0 | 3 |

**Figure 8: Worm Detection Results Windows Server 2000 SP4**

### 5.1.2 Discussion of results Network X

As discussed in Chapter 4, with an agreement with Company X, there were a series of penetration tests for a six month period to test the vulnerability of the company's network infrastructure. Because the network is a homogenous Microsoft Windows network, it provided the perfect opportunity to discover how vulnerable a Microsoft Network could be to a multiple attacks over a six month time period. As previously mentioned in the first chapter there are many entities that are defined under the Patriot Act as a critical infrastructure known to have Microsoft Windows networks. As a consequence of this, it was important to see how easy if at all it was to penetrate the security of Windows Networks. Figure 9 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008.

There was a 55% success rate when conducting rootkit attacks on Microsoft Windows Server 2000 SP4.


PENETRATION RESULTS ROOTKIT ATTACKS

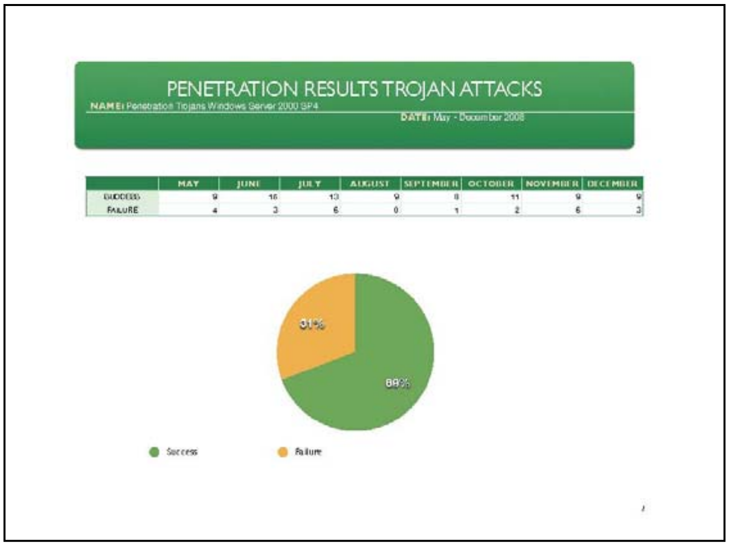
NAME: Penetration Rootkits Windows Server 2000 SP4

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 6 | 6 | 9 | 6 | 6 | 10 | 10 | 10 |
| FAILURE | 6 | 3 | 4 | 2 | 0 | 0 | 4 | 7 |

**Figure 9: Penetration Results Rootkits Attacks Windows Server 2000 SP4**

Figure 10 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 69% success rate when conducting Trojan attacks on Microsoft Windows Server 2000 SP4.


PENETRATION RESULTS TROJAN ATTACKS

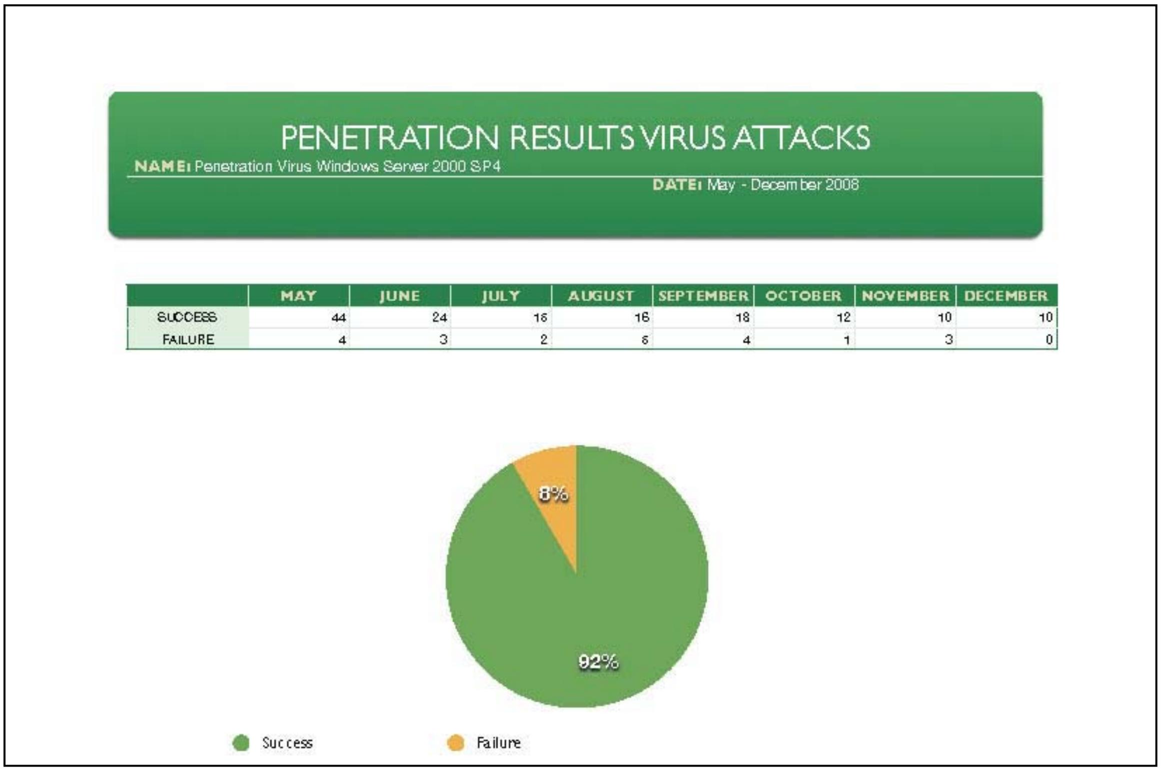
NAME: Penetration Trojans Windows Server 2000 SP4

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 9 | 16 | 13 | 9 | 8 | 11 | 9 | 9 |
| FAILURE | 4 | 3 | 6 | 0 | 1 | 2 | 6 | 3 |

**Figure 10: Penetration Results Trojan Attacks Windows Server 2000 SP4**

Figure 11 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 92% success rate when conducting virus attacks on Microsoft Windows Server 2000 SP4.



PENETRATION RESULTS VIRUS ATTACKS

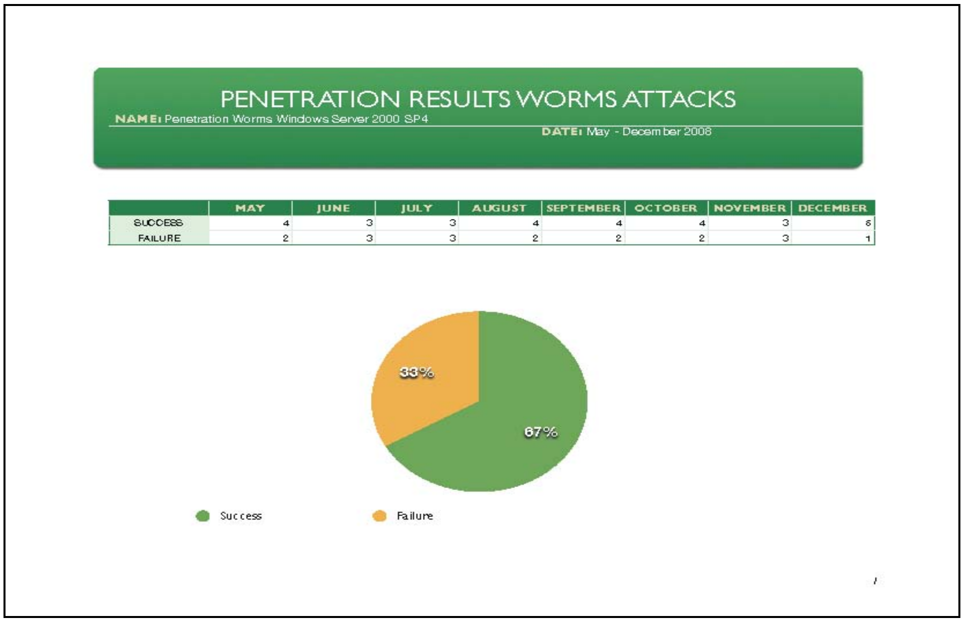
NAME: Penetration Virus Windows Server 2000 SP4

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 44 | 24 | 16 | 16 | 18 | 12 | 10 | 10 |
| FAILURE | 4 | 3 | 2 | 6 | 4 | 1 | 3 | 0 |

**Figure 11: Penetration Results Virus Attacks Windows Server 2000 SP4**

Figure 12 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 67% success rate when conducting worm attacks on Microsoft Windows Server 2000 SP4.



PENETRATION RESULTS WORMS ATTACKS

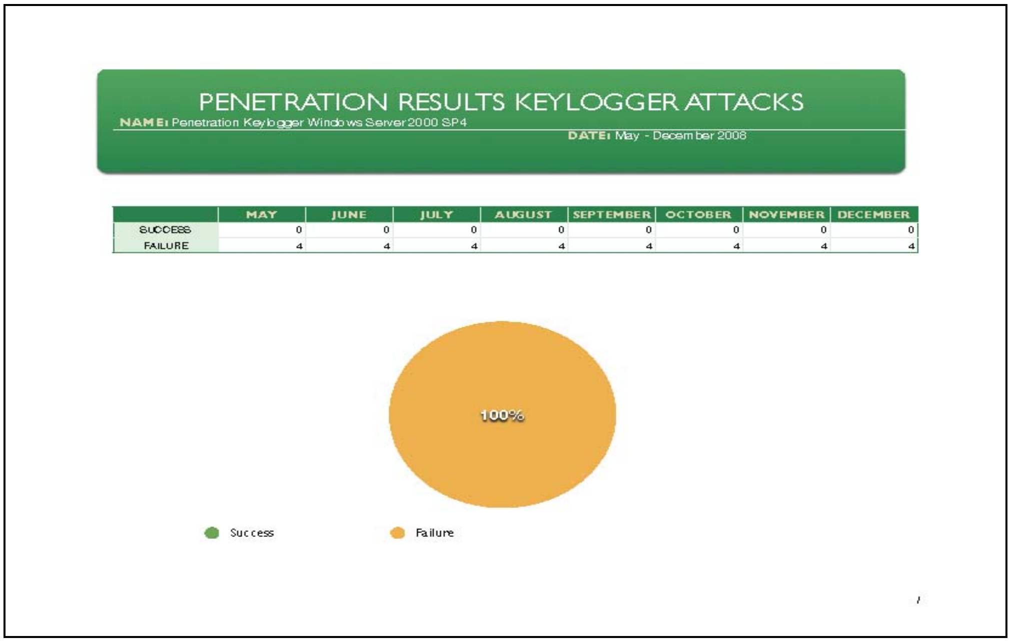
**NAME:** Penetration Worms Windows Server 2000 SP4

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 4 | 3 | 3 | 4 | 4 | 4 | 3 | 6 |
| FAILURE | 2 | 3 | 3 | 2 | 2 | 2 | 3 | 1 |

**Figure 12: Penetration Results Worm Attacks Windows Server 2000 SP4**

Figure 13 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 0% success rate when conducting keylogger attacks on Microsoft Windows Server 2000 SP4.



PENETRATION RESULTS KEYLOGGER ATTACKS

**NAME:** Penetration Keylogger Windows Server 2000 SP4

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FAILURE | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

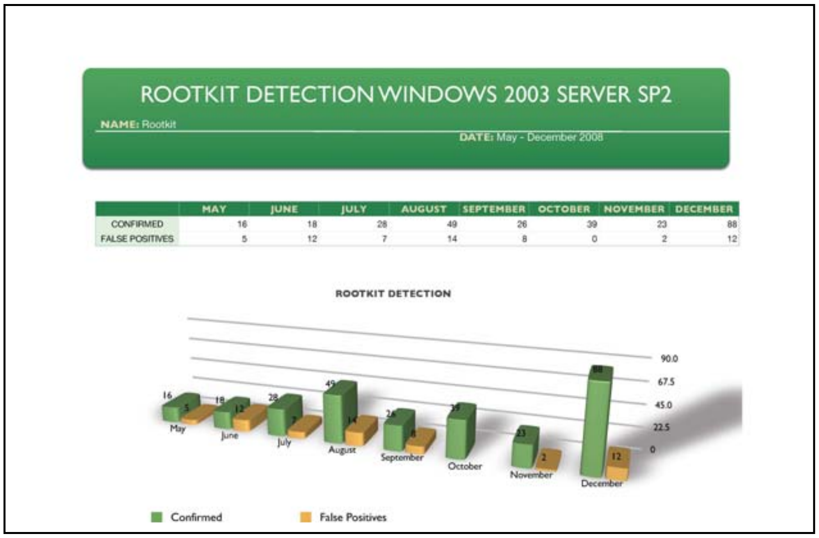
**Figure 13: Penetration Results Keylogger Attacks Windows Server 2000 SP4**

## 5.2 Windows Server 2003 Service Pack 2

In the following sections, the results for Windows Server 2003 with service pack 2 in Network Honey and Network X will be displayed and explained.

### 5.2.1 Discussion of Results Network Honey

Although Microsoft has recently introduced Server 2008 commercially, Microsoft Server 2003 continues to be the most popular and most deployed of all Microsoft Network Operating Systems (NOS). Because of this fact, it was essential to see how effective Microsoft Windows Server 2003 with service pack 2 would hold up against constant attacks over a six-month period. Figure 14 shows the number of confirmed rootkits detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 287 confirmed rootkits detected and a total of 60 false positives on Windows Server 2003 SP2. The month with the highest level of rootkits detected was in December with 88, and the lowest level of rootkits detected was May with 16. The month with the highest number of false positives was August with 14 and the month with the lowest number of false positives was October with 0.


ROOTKIT DETECTION WINDOWS 2003 SERVER SP2

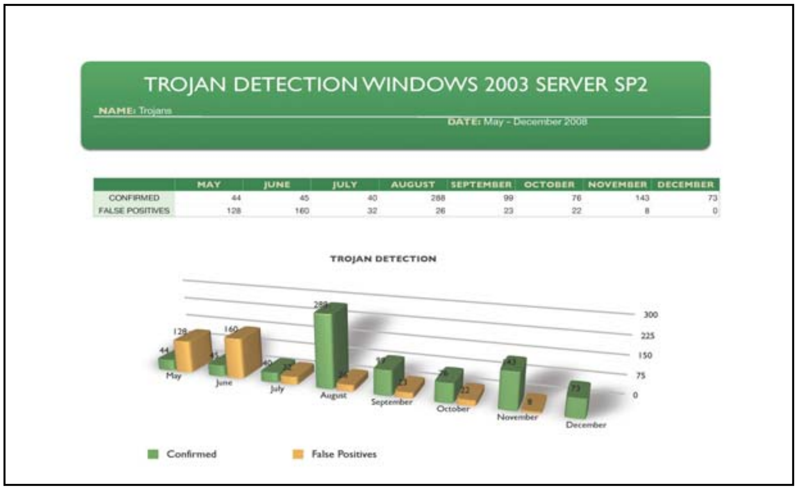
NAME: Rootkit

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 16 | 18 | 28 | 49 | 26 | 39 | 23 | 88 |
| FALSE POSITIVES | 5 | 12 | 7 | 14 | 8 | 0 | 2 | 12 |

**Figure 14: Rootkit Detection Results Windows Server 2003 SP2**

Figure 15 shows the number of confirmed trojans detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 788 confirmed trojans detected and a total of 399 false positives on Windows Server 2003 SP2. The month with the highest level of trojans detected was in August with 268, and the lowest level of trojans detected was July with 40. The month with the highest number of false positives was June with 160 and the month with the lowest number of false positives was December with 0.


TROJAN DETECTION WINDOWS 2003 SERVER SP2

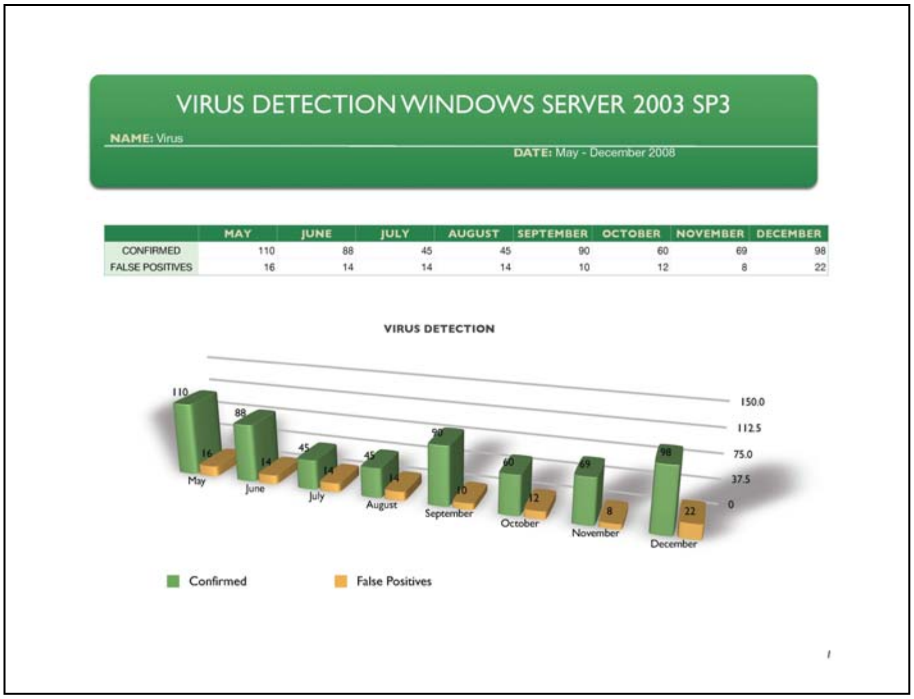
NAME: Trojans

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 44 | 45 | 40 | 288 | 99 | 76 | 143 | 73 |
| FALSE POSITIVES | 128 | 160 | 32 | 26 | 23 | 22 | 8 | 0 |

**Figure 15: Trojan Detection Results Windows Server 2003 SP2**

Figure 16 shows the number of confirmed viruses detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 1004 confirmed viruses detected and a total of 102 false positives on Windows Server 2003 SP2. The month with the highest level of viruses detected was in May with 110, and the lowest level of viruses detected was July and August with 45. The month with the highest number of false positives was May with 16 and the month with the lowest number of false positives was November with 8.



VIRUS DETECTION WINDOWS SERVER 2003 SP3

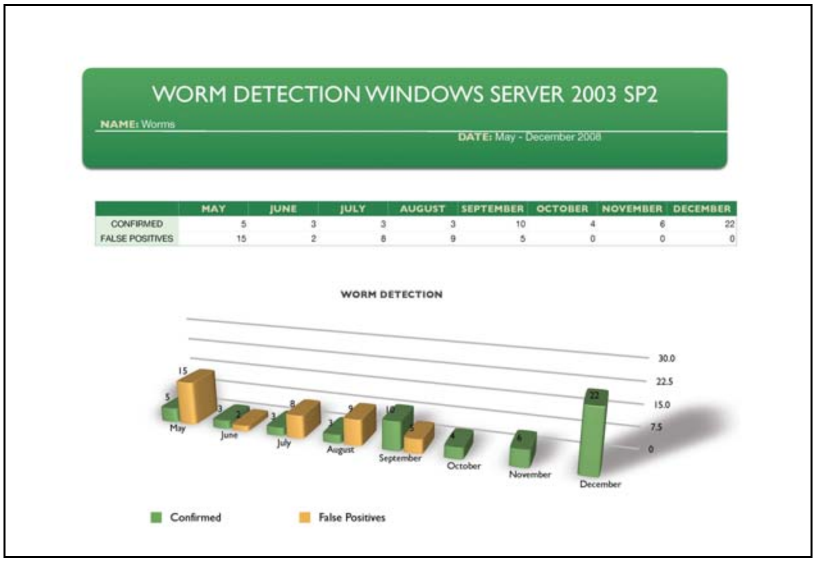
NAME: Virus

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 110 | 88 | 45 | 45 | 90 | 60 | 69 | 98 |
| FALSE POSITIVES | 16 | 14 | 14 | 14 | 10 | 12 | 8 | 22 |

**Figure 16: Virus Detection Results Windows Server 2003 SP2**

Figure 17 shows the number of confirmed worms detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 56 confirmed worms detected and a total of 39 false positives on Windows Server 2003 SP2. The month with the highest level of worms detected was in September with 10, and the lowest level of worms detected was June, July, and August with 3. The month with the highest number of false positives was May with 15 and the month with the lowest number of false positives were October, November, and December with 0



WORM DETECTION WINDOWS SERVER 2003 SP2

NAME: Worms

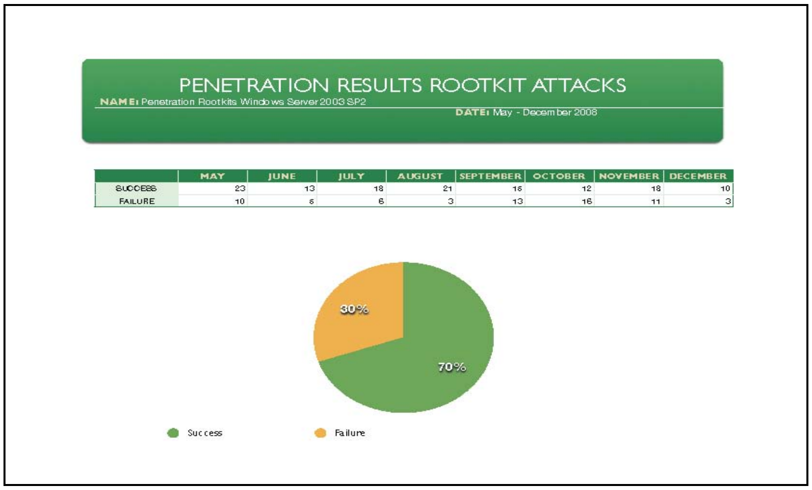
DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 5 | 3 | 3 | 3 | 10 | 4 | 6 | 22 |
| FALSE POSITIVES | 15 | 2 | 8 | 9 | 5 | 0 | 0 | 0 |

**Figure 17: Worm Detection Results Windows Server 2003 SP2**

### 5.2.2 Discussion of results Network X

Figure 18 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. The author had a 70% success rate when conducting rootkit attacks on Microsoft Windows Server 2003 SP2.



PENETRATION RESULTS ROOTKIT ATTACKS

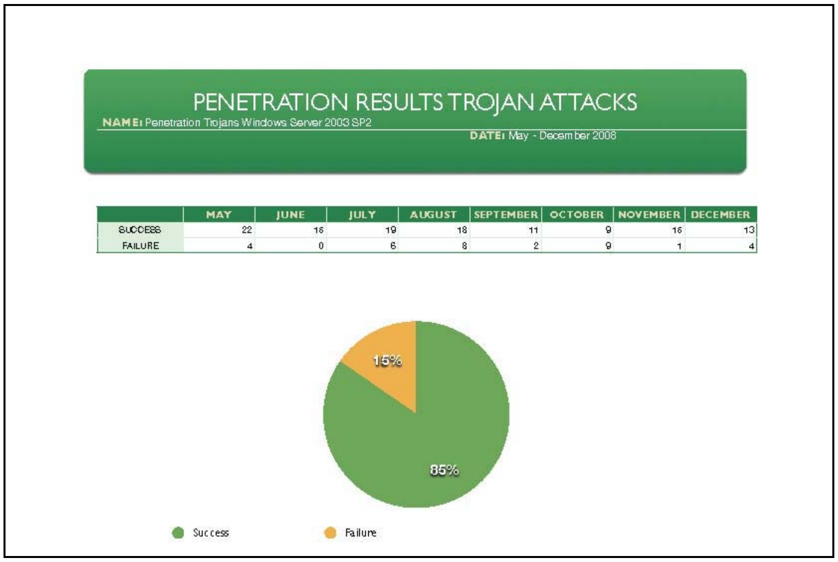
NAME: Penetration Rootkits Windows Server 2003 SP2

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 23 | 13 | 18 | 21 | 16 | 12 | 18 | 10 |
| FAILURE | 10 | 6 | 6 | 3 | 13 | 16 | 11 | 3 |

**Figure 18: Rootkit Penetration Results Windows Server 2003 SP2**

Figure 19 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was an 85% success rate when conducting trojan attacks on Microsoft Windows Server 2003 SP2.



PENETRATION RESULTS TROJAN ATTACKS

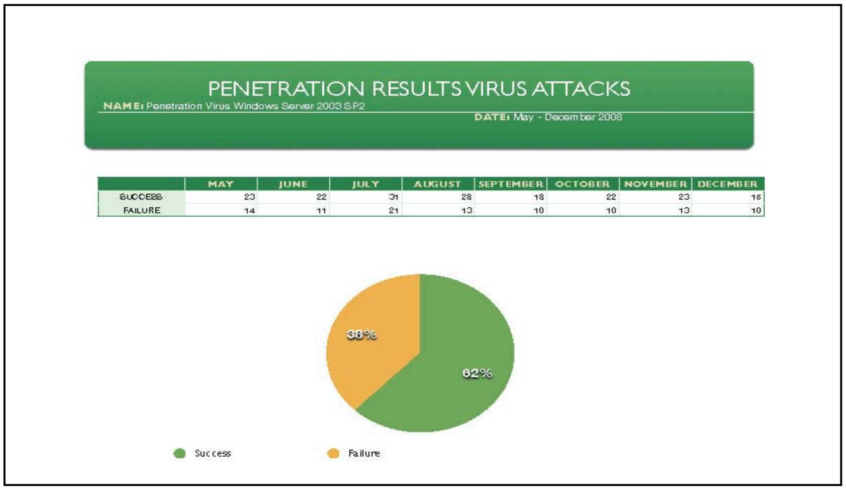
NAME: Penetration Trojans Windows Server 2003 SP2

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 22 | 16 | 19 | 18 | 11 | 9 | 16 | 13 |
| FAILURE | 4 | 0 | 6 | 8 | 2 | 9 | 1 | 4 |

**Figure 19: Trojan Penetration Results Windows Server 2003 SP2**

Figure 20 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. The author had a 62% success rate when conducting virus attacks on Microsoft Windows Server 2003 SP2.



PENETRATION RESULTS VIRUS ATTACKS

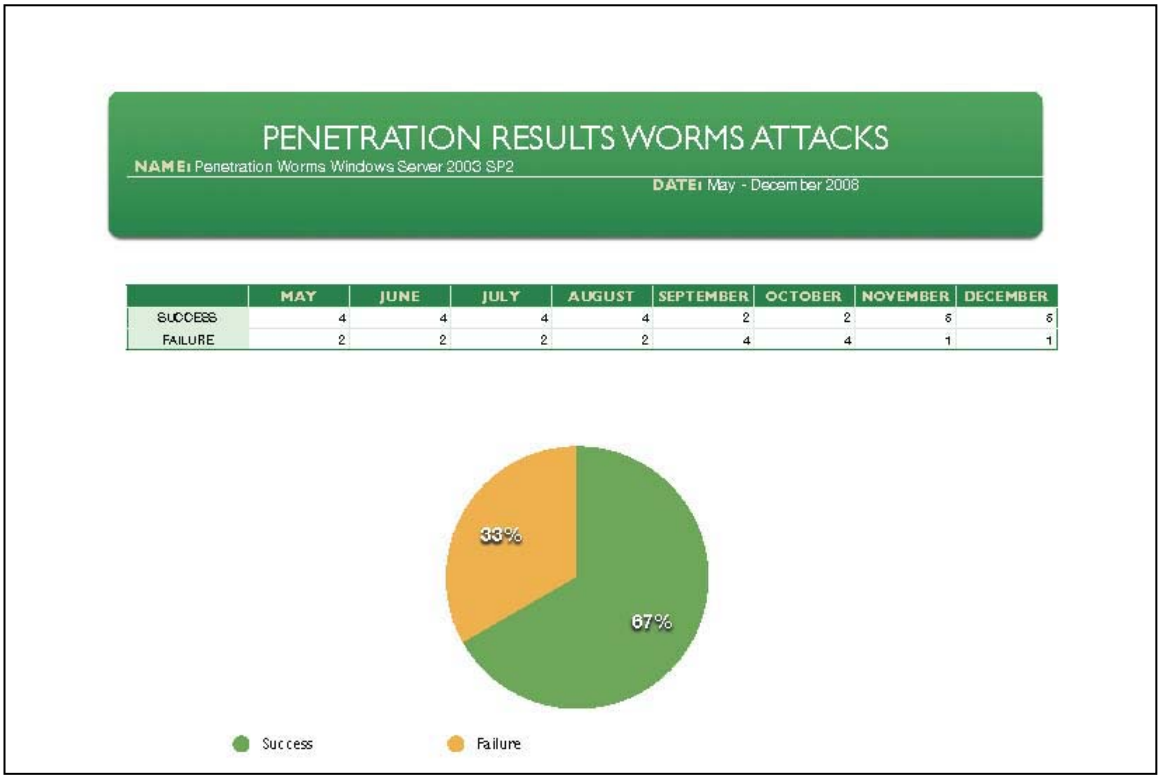
NAME: Penetration Virus Windows Server 2003 SP2

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 23 | 22 | 31 | 28 | 18 | 22 | 23 | 16 |
| FAILURE | 14 | 11 | 21 | 13 | 10 | 10 | 13 | 10 |

**Figure 20: Virus Penetration Results Windows Server 2003 SP2**

Figure 21 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 67% success rate when conducting worm attacks on Microsoft Windows Server 2003 SP2.



PENETRATION RESULTS WORMS ATTACKS

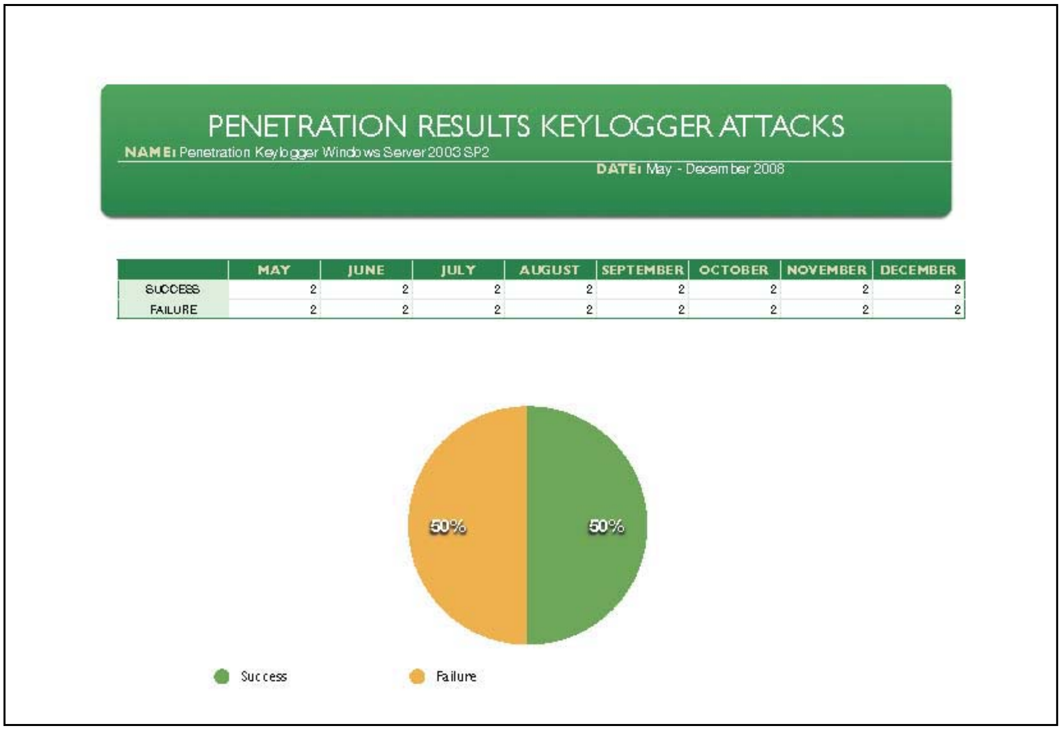
NAME: Penetration Worms Windows Server 2003 SP2

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 4 | 4 | 4 | 4 | 2 | 2 | 6 | 6 |
| FAILURE | 2 | 2 | 2 | 2 | 4 | 4 | 1 | 1 |

**Figure 21: Worm Penetration Results Windows Server 2003 SP2**

Figure 22 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 50% success rate when conducting keylogger attacks on Microsoft Windows Server 2003 SP2.



PENETRATION RESULTS KEYLOGGER ATTACKS

**NAME:** Penetration Keylogger Windows Server 2003 SP2

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| FAILURE | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

**Figure 22: Keylogger Penetration Results Windows Server 2003 SP2**

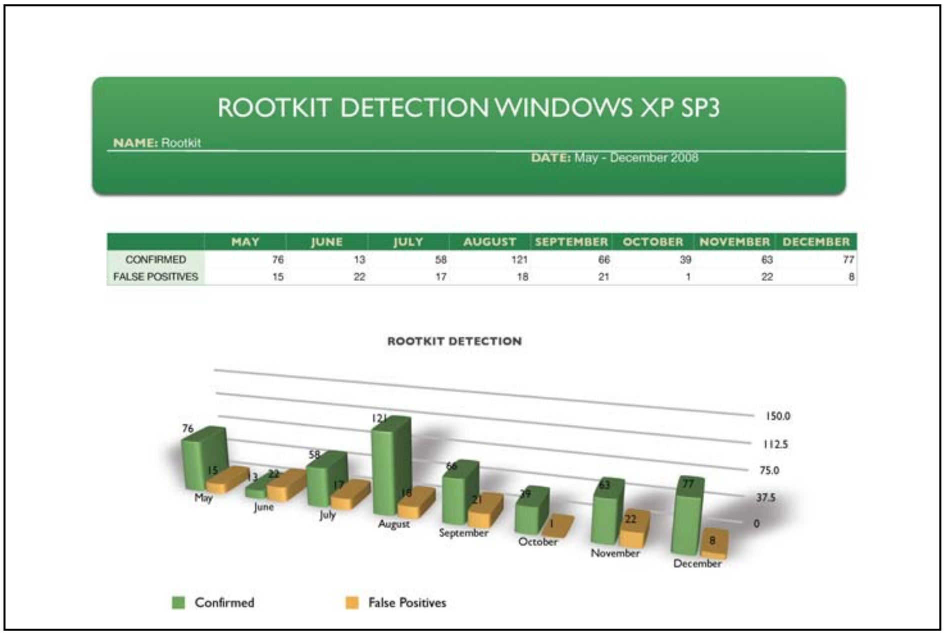
### 5.3 Windows XP Service Pack 3

In the following sections, the results for Windows Server 2003 with service pack 2 in Network Honey and Network X will be displayed and explained.

#### 5.3.1 Discussion of Results Network Honey

Windows XP is the most popular and most widely used Microsoft Desktop operating system. Because of this fact, it was essential to see how effective Microsoft Windows XP with service pack 3 would hold up against constant attacks over a six-month period. Figure 23 shows the number of confirmed rootkits detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 513 confirmed rootkits detected and a total of 124 false positives on Windows XP SP3. The month with the highest level of rootkits detected was in August with 121, and the lowest level of rootkits detected was June with

13. The month with the highest number of false positives was June and November with 22 and the month with the lowest number of false positives was October with 1.



ROOTKIT DETECTION WINDOWS XP SP3

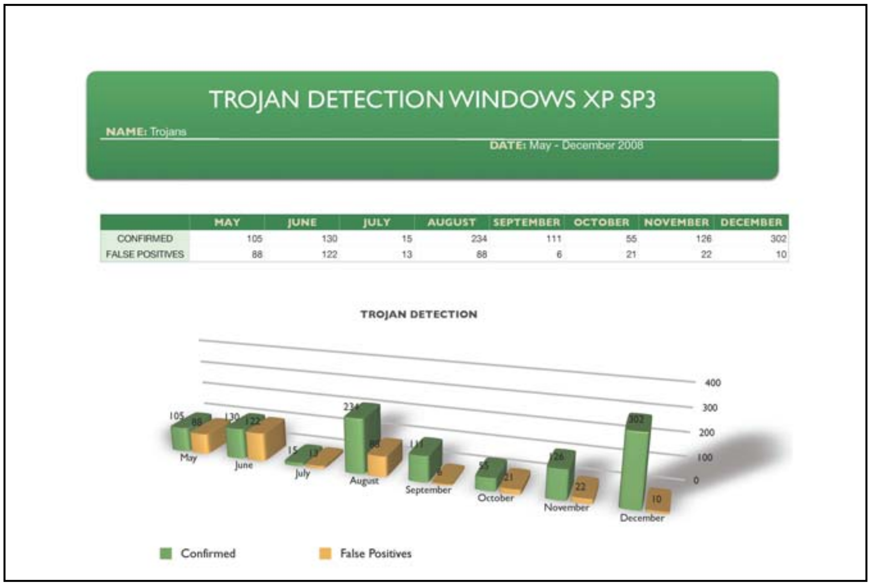
NAME: Rootkit

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 76 | 13 | 58 | 121 | 66 | 39 | 63 | 77 |
| FALSE POSITIVES | 15 | 22 | 17 | 18 | 21 | 1 | 22 | 8 |

**Figure 23: Rootkit Detection Results Windows XP SP3**

Figure 24 shows the number of confirmed trojans detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 1078 confirmed trojans detected and a total of 348 false positives on Windows XP SP3. The month with the highest level of trojans detected was in December with 302, and the lowest level of trojans detected was July with 15. The month with the highest number of false positives was June with 122 and the month with the lowest number of false positives was September with 6.



TROJAN DETECTION WINDOWS XP SP3

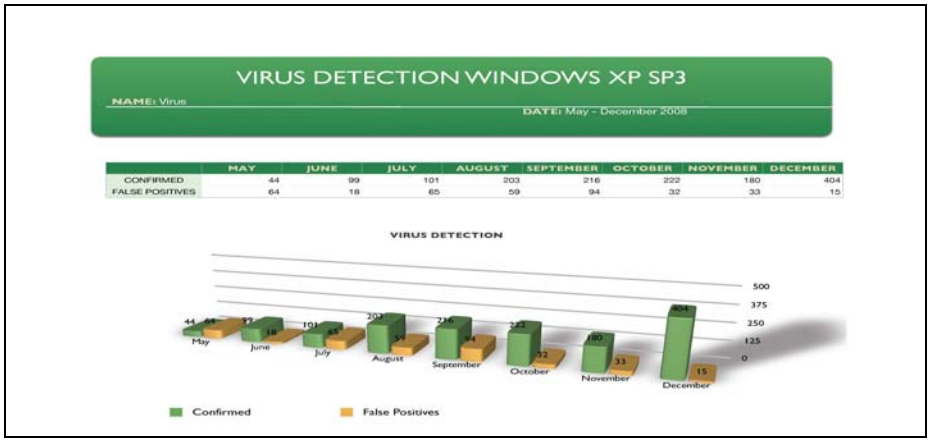
NAME: Trojans

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 105 | 130 | 15 | 234 | 111 | 55 | 126 | 302 |
| FALSE POSITIVES | 88 | 122 | 13 | 88 | 6 | 21 | 22 | 10 |

**Figure 24: Trojan Detection Results Windows XP SP3**

Figure 25 shows the number of confirmed viruses detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 1469 confirmed viruses detected and a total of 380 false positives on Windows XP SP3. The month with the highest level of viruses detected was in December with 404, and the lowest level of viruses detected was May with 44. The month with the highest number of false positives was July with 65 and the month with the lowest number of false positives was December with 15.



VIRUS DETECTION WINDOWS XP SP3

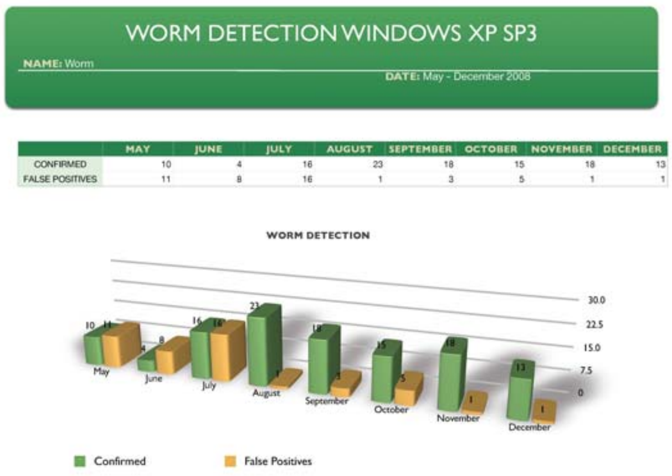
NAME: Virus

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 44 | 99 | 101 | 203 | 216 | 222 | 180 | 404 |
| FALSE POSITIVES | 64 | 18 | 65 | 59 | 94 | 32 | 33 | 15 |

**Figure 25: Virus Detection Results Windows XP SP3**

Figure 26 shows the number of confirmed worms detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 118 confirmed worms detected and a total of 46 false positives on Windows XP SP3. The month with the highest level of worms detected was in August with 23, and the lowest level of worms detected was June, July, and August with 3. The month with the highest number of false positives was May with 15 and the month with the lowest number of false positives were August, November, and December with 1.



WORM DETECTION WINDOWS XP SP3

NAME: Worm

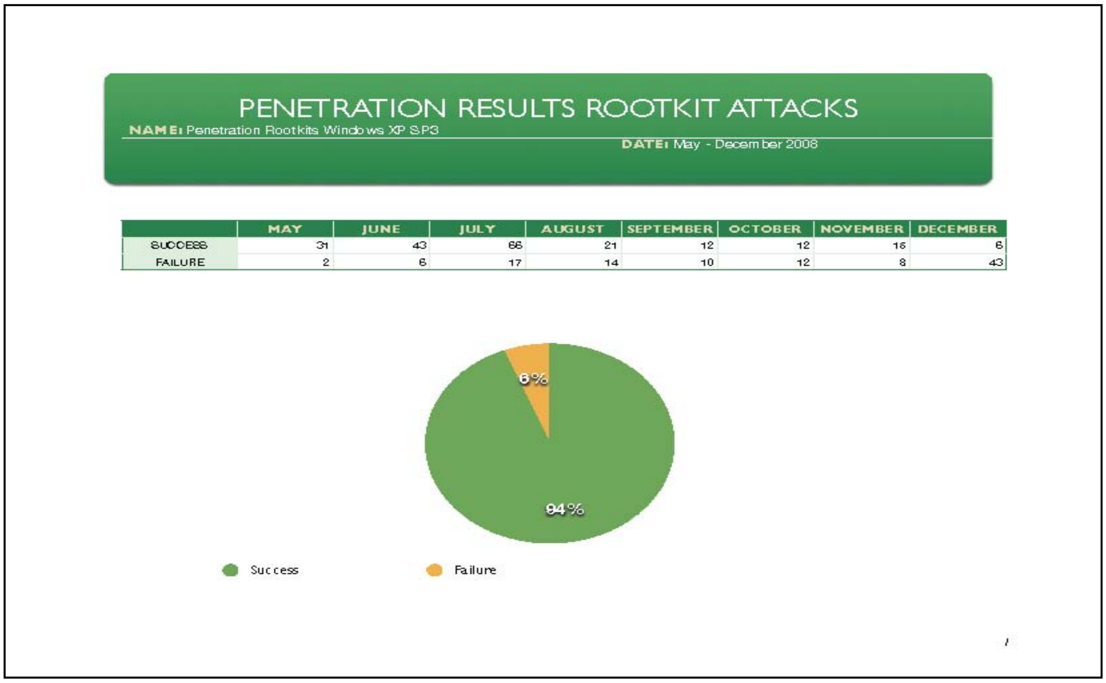
DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 10 | 4 | 16 | 23 | 18 | 15 | 18 | 13 |
| FALSE POSITIVES | 11 | 8 | 16 | 1 | 3 | 5 | 1 | 1 |

**Figure 26: Worm Detection Results Windows XP SP3**

### 5.3.2 Discussion of results Network X

Figure 27 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 94% success rate when conducting Rootkit attacks on Microsoft Windows XP SP3.



PENETRATION RESULTS ROOTKIT ATTACKS

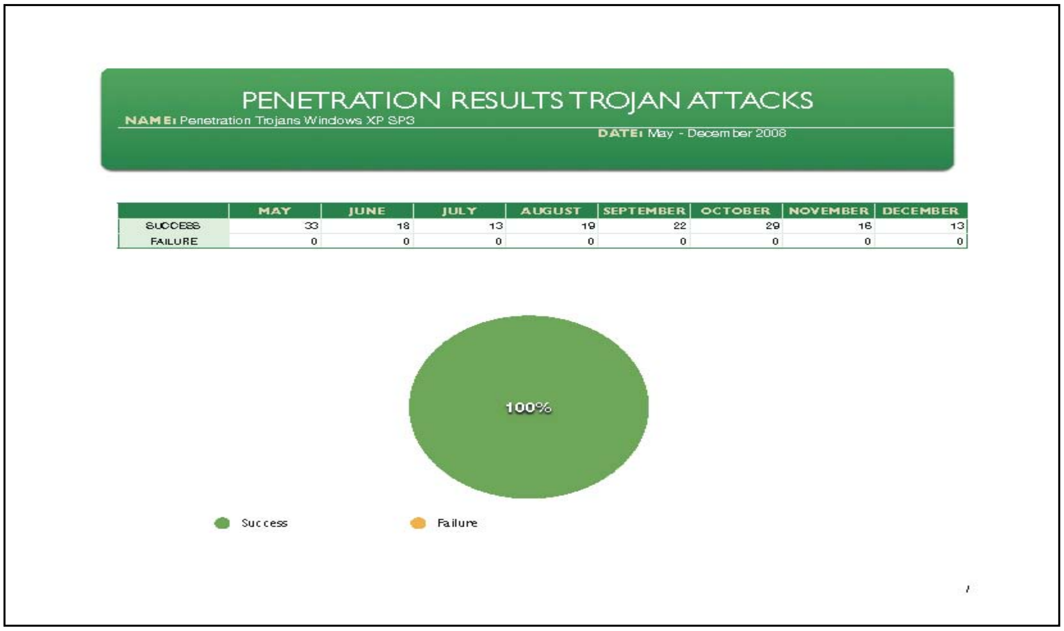
**NAME:** Penetration Rootkits Windows XP SP3

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 31 | 43 | 66 | 21 | 12 | 12 | 16 | 6 |
| FAILURE | 2 | 6 | 17 | 14 | 10 | 12 | 8 | 43 |

**Figure 27: Rootkit Penetration Results Windows XP SP3**

Figure 28 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 100% success rate when conducting Trojan attacks on Microsoft Windows XP SP3.



PENETRATION RESULTS TROJAN ATTACKS

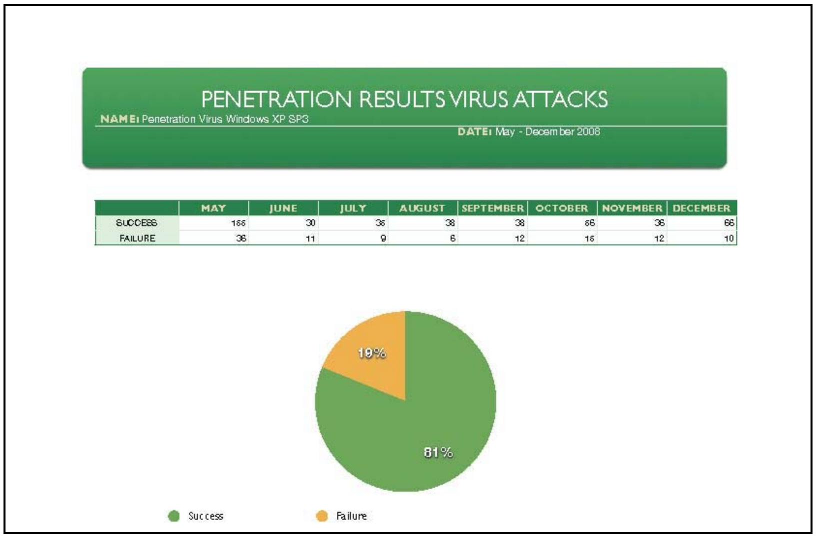
**NAME:** Penetration Trojans Windows XP SP3

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 33 | 18 | 13 | 19 | 22 | 29 | 16 | 13 |
| FAILURE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 28: Trojan Penetration Results Windows XP SP3**

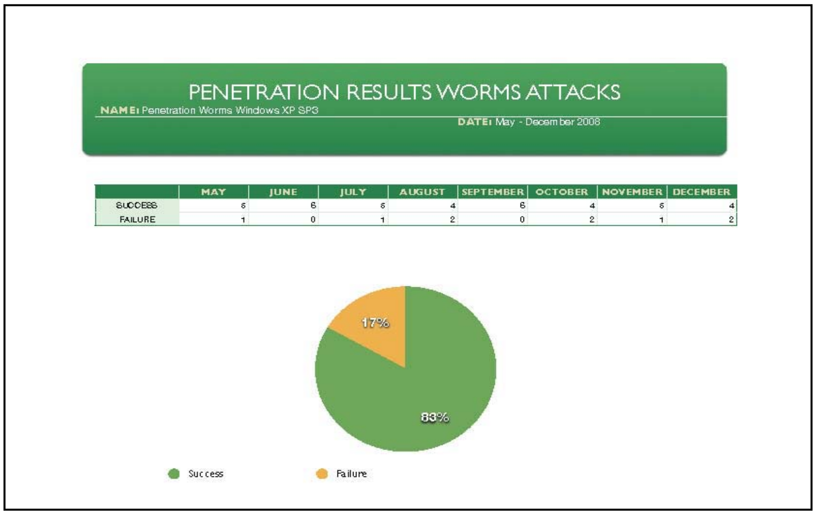
Figure 29 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was an 81% success rate when conducting virus attacks on Microsoft Windows XP SP3.



| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 166 | 30 | 36 | 38 | 38 | 66 | 36 | 66 |
| FAILURE | 36 | 11 | 9 | 6 | 12 | 16 | 12 | 10 |

**Figure 29: Virus Penetration Results Windows XP SP3**

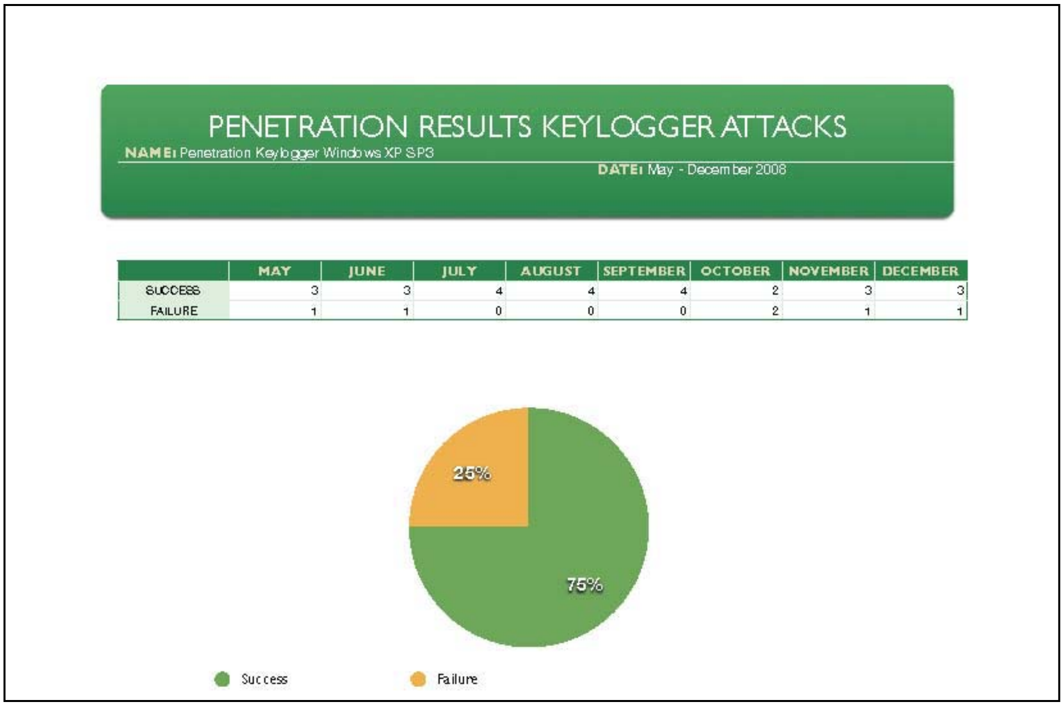
Figure 30 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was an 83% success rate when conducting worm attacks on Microsoft Windows XP SP3.



| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 6 | 6 | 6 | 4 | 6 | 4 | 6 | 4 |
| FAILURE | 1 | 0 | 1 | 2 | 0 | 2 | 1 | 2 |

**Figure 30: Worm Penetration Results Windows XP SP3**

Figure 31 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 75% success rate when conducting keylogger attacks on Microsoft Windows XP SP3.



PENETRATION RESULTS KEYLOGGER ATTACKS

NAME: Penetration Keylogger Windows XP SP3

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 3 | 3 | 4 | 4 | 4 | 2 | 3 | 3 |
| FAILURE | 1 | 1 | 0 | 0 | 0 | 2 | 1 | 1 |

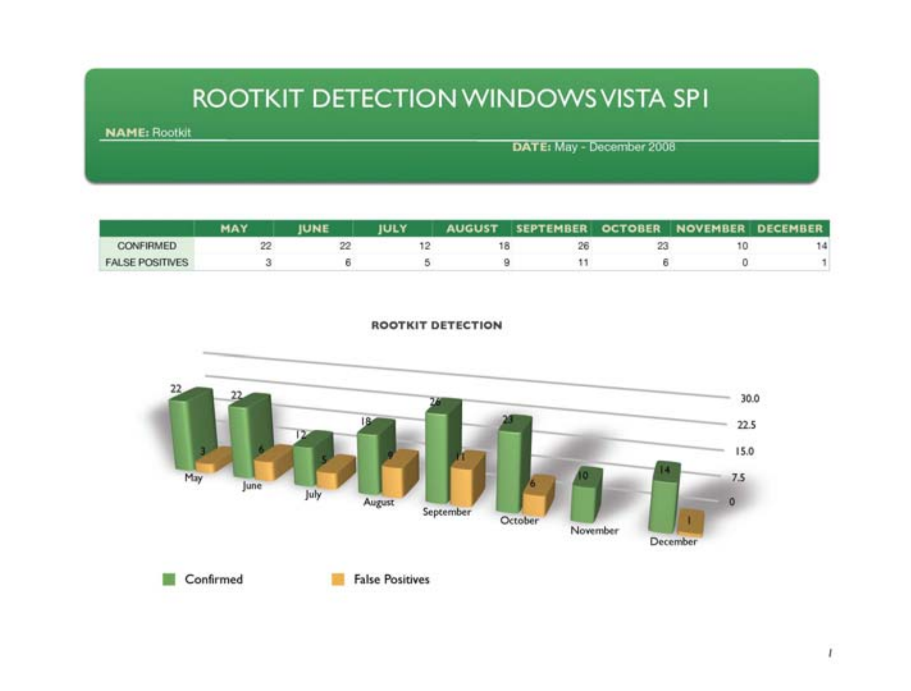
**Figure 31: Keylogger Penetration Results Windows XP SP3**

## 5.4 Windows Vista Service Pack 1

In the following sections, the results for Windows Vista with service pack 1 in Network Honey and Network X will be displayed and explained.

### 5.4.1 Discussion of Results Network Honey

Windows Vista is widely considered to be an absolute failure in the IT community and to the average consumer; however, because of Microsoft’s deal with OEMs, every new PC purchase in the future will either have Vista, or a derivative of Vista (Windows 7). Because of this fact, it was essential to see how effective Microsoft Vista with service pack 1 would hold up against constant attacks over a six-month period. Figure 32 shows the number of confirmed rootkits detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 145 confirmed rootkits detected and a total of 43 false positives on Windows Vista SP1. The month with the highest level of rootkits detected was in September with 26, and the lowest level of rootkits detected was November with 10. The month with the highest number of false positives was September with 11 and the month with the lowest number of false positives was November with 0.



ROOTKIT DETECTION WINDOWS VISTA SP1

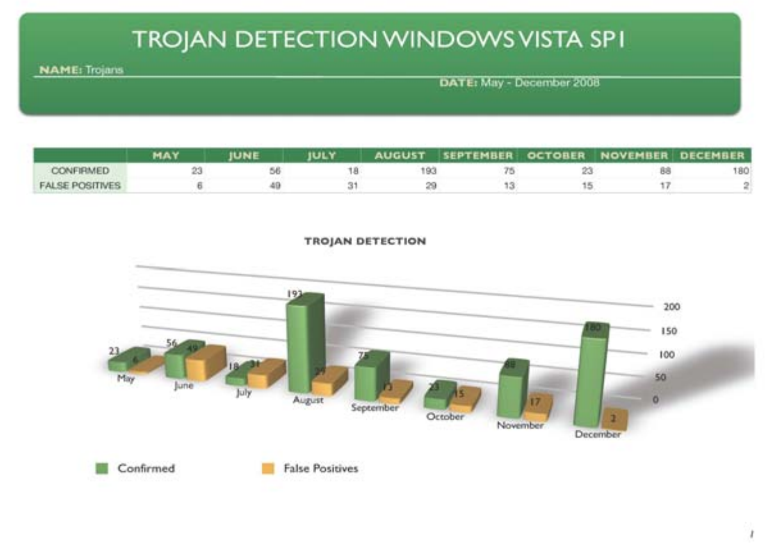
**NAME:** Rootkit

**DATE:** May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 22 | 22 | 12 | 18 | 26 | 23 | 10 | 14 |
| FALSE POSITIVES | 3 | 6 | 5 | 9 | 11 | 6 | 0 | 1 |

**Figure 32: Rootkit Detection Results Windows Vista SP1**

Figure 33 shows the number of confirmed trojans detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 656 confirmed trojans detected and a total of 163 false positives on Windows Vista SP1. The month with the highest level of trojans detected was in August with 193, and the lowest level of trojans detected was July with 18. The month with the highest number of false positives was June with 49 and the month with the lowest number of false positives was December with 2.



TROJAN DETECTION WINDOWS VISTA SP1

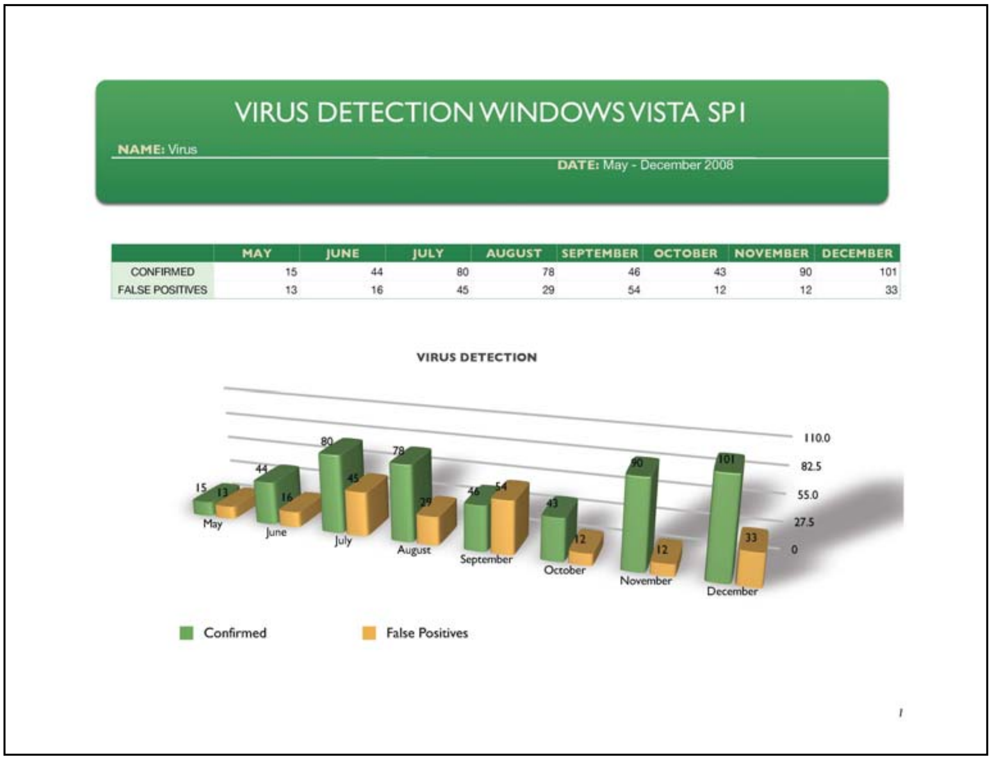
NAME: Trojans

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 23 | 56 | 18 | 193 | 75 | 23 | 88 | 180 |
| FALSE POSITIVES | 6 | 49 | 31 | 29 | 13 | 15 | 17 | 2 |

**Figure 33: Trojan Detection Results Windows Vista SP1**

Figure 34 shows the number of confirmed virus detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 497 confirmed viruses detected and a total of 214 false positives on Windows Vista SP1. The month with the highest level of viruses detected was in December with 101, and the lowest level of viruses detected was May with 15. The month with the highest number of false positives was September with 54 and the month with the lowest number of false positives were October and November with 12.



VIRUS DETECTION WINDOWS VISTA SP1

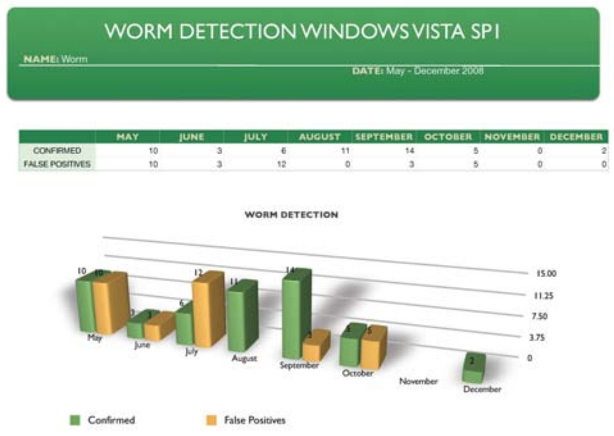
NAME: Virus

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 15 | 44 | 80 | 78 | 46 | 43 | 90 | 101 |
| FALSE POSITIVES | 13 | 16 | 45 | 29 | 54 | 12 | 12 | 33 |

**Figure 34: Virus Detection Results Windows Vista SP1**

Figure 35 shows the number of confirmed worms detected and the amount of false positives detected from May of 2008 until December of 2008. Throughout the six-month period, there was a total of 51 confirmed worms detected and a total of 33 false positives on Windows Vista SP1. The month with the highest level of worms detected was in August with 23, and the lowest level of worms detected was September with 14. The month with the highest number of false positives was July with 12 and the month with the lowest number of false positives were August, November, and December with 0.



WORM DETECTION WINDOWS VISTA SP1

NAME: Worm

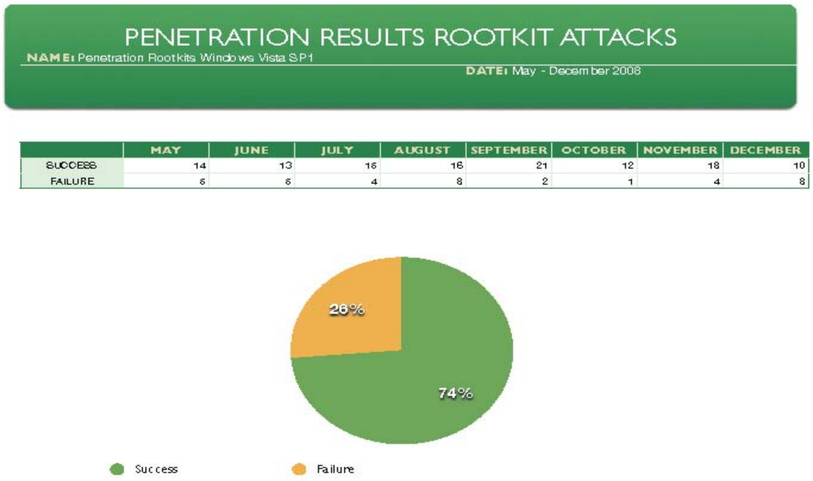
DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| CONFIRMED | 10 | 3 | 6 | 11 | 14 | 5 | 0 | 2 |
| FALSE POSITIVES | 10 | 3 | 12 | 0 | 3 | 5 | 0 | 0 |

**Figure 35: Worm Detection Results Windows Vista SP1**

### 5.4.2 Discussion of Results Network X

Figure 36 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 74% success rate when conducting Rootkit attacks on Microsoft Windows Vista SP1.



PENETRATION RESULTS ROOTKIT ATTACKS

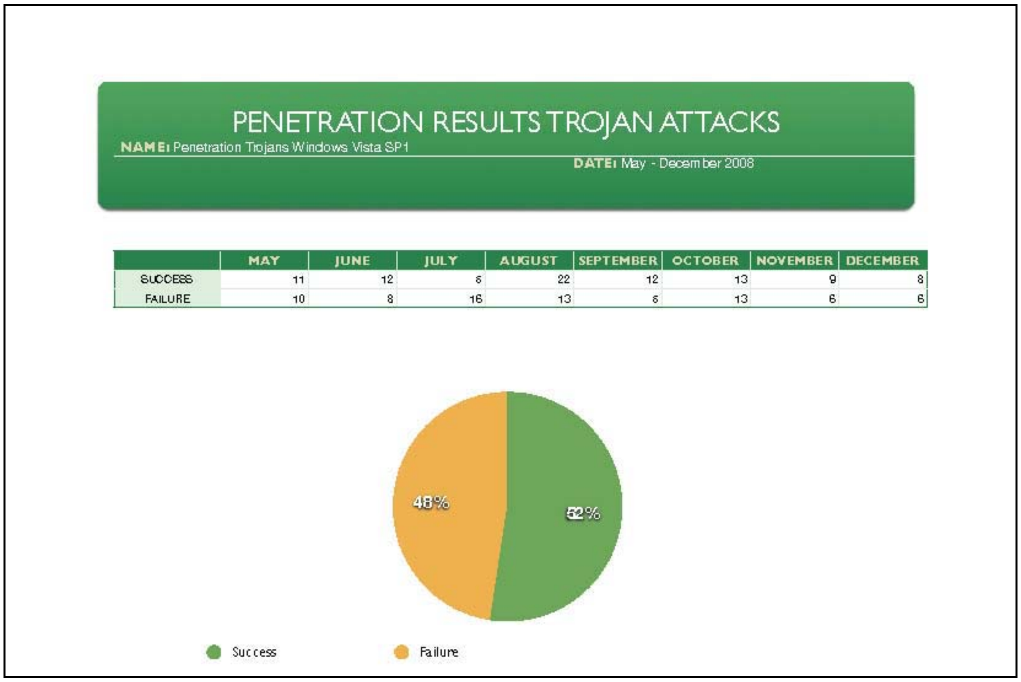
NAME: Penetration Rootkits Windows Vista SP1

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 14 | 13 | 15 | 16 | 21 | 12 | 18 | 10 |
| FAILURE | 5 | 5 | 4 | 8 | 2 | 1 | 4 | 8 |

**Figure 36: Penetration Test Results Rootkit Attacks Windows Vista SP1**

Figure 37 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 52% success rate when conducting Trojan attacks on Microsoft Windows Vista SP1.



PENETRATION RESULTS TROJAN ATTACKS

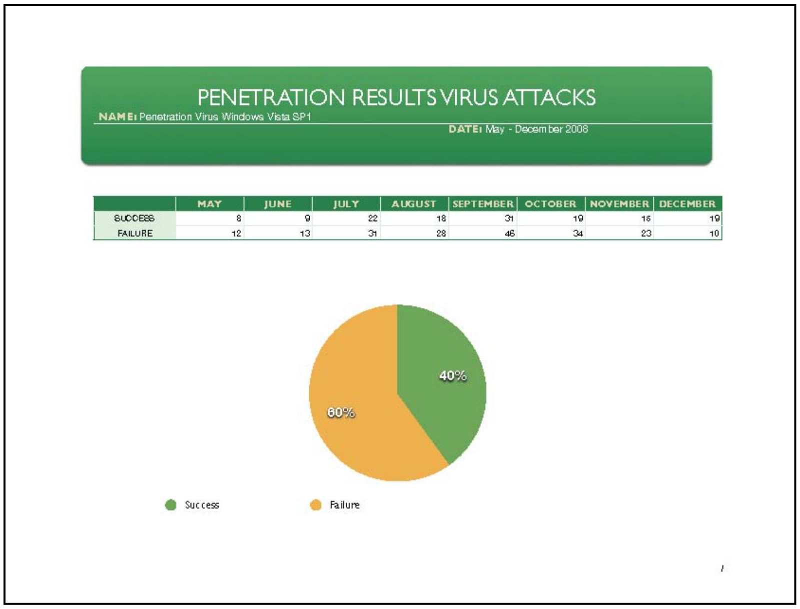
NAME: Penetration Trojans Windows Vista SP1

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 11 | 12 | 6 | 22 | 12 | 13 | 9 | 8 |
| FAILURE | 10 | 8 | 16 | 13 | 6 | 13 | 6 | 6 |

**Figure 37: Penetration Test Results Trojan Attacks Windows Vista SP1**

Figure 38 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 40% success rate when conducting virus attacks on Microsoft Windows Vista SP1.



PENETRATION RESULTS VIRUS ATTACKS

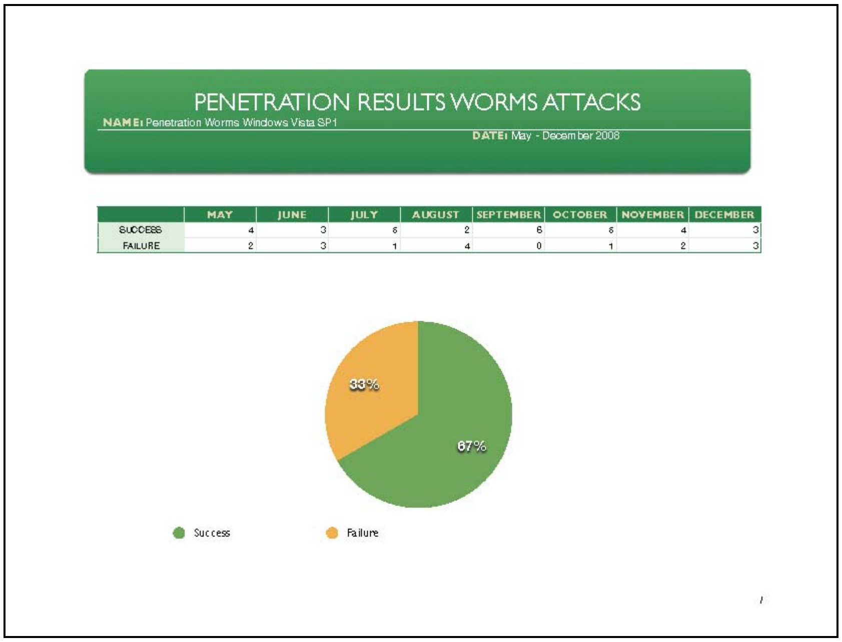
NAME: Penetration Virus Windows Vista SP1

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 8 | 9 | 22 | 18 | 31 | 19 | 16 | 19 |
| FAILURE | 12 | 13 | 31 | 28 | 46 | 34 | 23 | 10 |

**Figure 38: Penetration Test Results Virus Attacks Windows Vista SP1**

Figure 39 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 67% success rate when conducting worm attacks on Microsoft Windows Vista SP1.



PENETRATION RESULTS WORMS ATTACKS

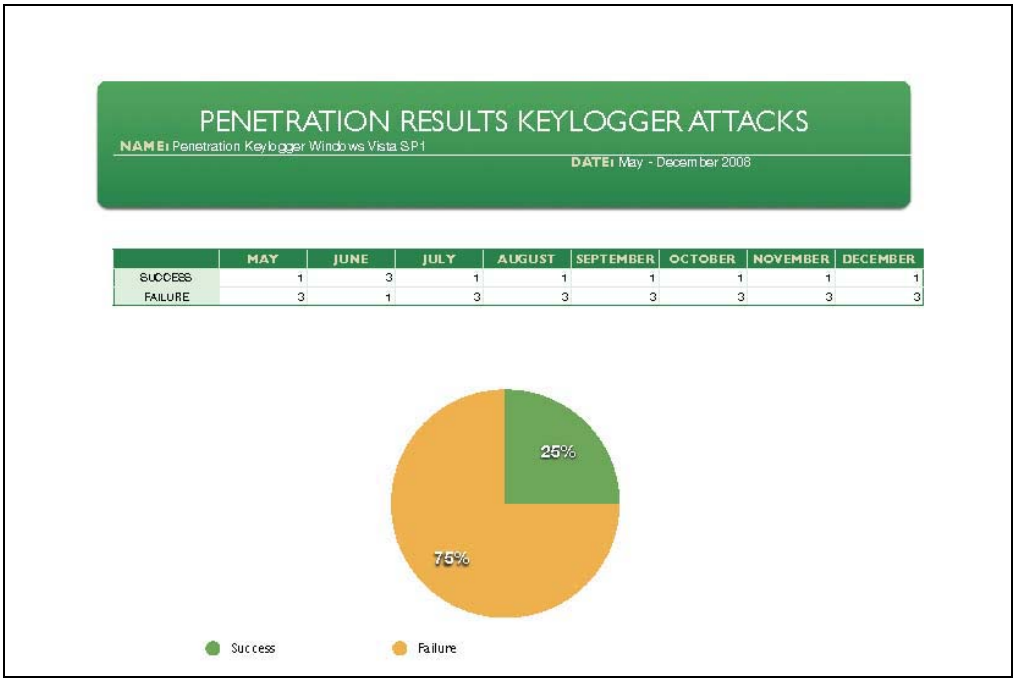
NAME: Penetration Worms Windows Vista SP1

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 4 | 3 | 5 | 2 | 6 | 5 | 4 | 3 |
| FAILURE | 2 | 3 | 1 | 4 | 0 | 1 | 2 | 3 |

**Figure 39: Penetration Test Results Worm Attacks Windows Vista SP1**

Figure 40 shows the overall percentage of success and failures when conducting penetration tests from May of 2008 until December of 2008. There was a 25% success rate when conducting keylogger attacks on Microsoft Windows Vista SP1.



PENETRATION RESULTS KEYLOGGER ATTACKS

NAME: Penetration Keylogger Windows Vista SP1

DATE: May - December 2008

| | MAY | JUNE | JULY | AUGUST | SEPTEMBER | OCTOBER | NOVEMBER | DECEMBER |
|---|---|---|---|---|---|---|---|---|
| SUCCESS | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| FAILURE | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |

**Figure 40: Penetration Test Results Keylogger Attacks Windows Vista SP1**

## Chapter 6: Summary and Conclusions

Over the last four to five years, the U.S. government has seen an increase of successful attacks against major critical infrastructures as defined by the Patriot Act. It just so happens that the majority of these entities if not all have a Microsoft Windows network infrastructure. According to a recent article in the Wall Street Journal, the U.S. electrical grid was penetrated by Cyberspies [59]. They left behind software programs that could be used to disrupt the system. Authorities that investigated the intrusions found software tools left behind that could be used to destroy infrastructure components. The article goes on to say that in 2008, there were a lot of intrusions in the networks that control the electrical grid, and that intelligence officials fear that nuclear power plants, financial networks, water, sewage, and other infrastructures can be controlled via the Internet [59]. Let’s examine the U.S. electrical grid. The U.S. electrical grid comprises three separate electric networks, covering the East, the West and Texas. Each includes many thousands of miles of transmission lines, power plants and substations. The flow of power is controlled by local utilities or regional transmission organizations. If the grid was successfully hijacked, it could cause irreparable damage to the United States [59].

The experiment done in this thesis serves as a microcosm to a macro-problem. Microsoft Windows networks are too vulnerable to serve as the backbone for any institution or organization’s networking infrastructure, especially entities considered to be government critical infrastructures. That being said, in the author’s opinion, Microsoft is a threat to national security. The results from Network Honey stipulate that even with fully patched heavily guarded (third party applications, firewall, antivirus etc.) Microsoft Windows hosts on the network, the ease at which these systems were compromised and the success of attacks means the security apparatus needs to be changed. The results from the network penetration tests basically substantiate this claim. The author, started out with limited skills, and with the help of hacker forums and sites, combined with a few books, was able to successfully penetrate an enterprise network.

It has often been said that the concept of cyberwarfare is synonymous with the idea of a unicorn; something that has been read in books and seen at movie theaters, but would never come to fruition. Some experts say that the idea a hacker or group of hackers could actually take over a country and hold it hostage is null and void. There is only one word for those skeptics and non-believers, and that word is Estonia.

In May of 2007 in Estonia attackers used a giant botnet — perhaps as many as one million computers in places as far away as the United States and Vietnam — to amplify the impact of their assault [60]. Attackers went for bank sites, newspapers, foreign ministry sites, and government-connected sites. Many of the bots had their targets hard-coded into their source. There were dozens of attacks, some lasting as long as 10 hours each and slamming Estonia's servers with 90 megabits of data a second. Estonia eventually survived it, but they suffered considerable financial losses [60].

The attack on Estonia should serve as a reason to worry if the U.S. government, banks, corporations, and other business entities continue to use Microsoft products. There already have been reports in the last two months of breaches in the Pentagon (Windows environment) and the Air Forces' air-traffic-control system that resulted in data loss. The Pentagon was compromised, and its $300 billion Joint Strike Fighter Project—the Defense Department's costliest weapons program ever was broken into. According to an article in the Wall Street Journal, the intruders were able to copy and siphon off several terabytes of data related to design and electronic systems, potentially making it easier to defend against the craft [61]. If the government does not make any radical changes in the near future to overhaul their network infrastructure and rid itself of Microsoft Windows, there could be dire consequences.

### 6.1: Future Research

Time constraints restricted the possibility of putting Linux and Unix systems under the same tests in order to do a comparative analysis. The hypothesis is that if the government overhauls its current Windows infrastructure with that of either Linux or Unix systems, it would create a more stable more secure computing environment.

## References

[1] James J. F. Forest, “Homeland Security: Protecting America's Targets: Volume 3 Critical Infrastructure”, Connecticut, Praeger Security International, 2006.

[2] Ramona R. Rantala, 2005. “Bureau of Justice Statistics Special Report U.S. Department of Justice Office of Justice Programs Cybercrime against Businesses, 2005 (Revised September 2008, NCJ 221943)” http://www.ojp.usdoj.gov/bjs/pub/pdf/cb05.pdf (accessed January 3, 2009).

[3] L. Janczewski and A. Colarik, “Cyber Warfare and Cyber Terrorism”, Pennsylvania, Information Science Reference, 2008.

[4] S. Jarkoff, 2009. “DoD Has No Desire to Mitigate Windows Dependency” http://techmiso.com/271/dod-has-no-desire-to-mitigate-windows-dependency (accessed January 18, 2009).

[5] “Written Direct Testimony of Jim Allchin In the United States District court for the district of Columbia” http://www.microsoft.com/presspass/legal/allchin.mspx (accessed January 30, 2009).

[6] Brian L. Stuart, “Principles of operating systems : Design & Applications”, Massachusetts, Course Technology, 2008.

[7] Laurie S. Keller, “Operating systems: Communicating With and Controlling the Computer”, New Jersey, Prentice Hall, 1988.

[8] G. Hoglund and J. Butler, “Rootkits: subverting the Windows kernel”, New Jersey, Addison Wesley, 2005.

[9] J. Kong, “Designing BSD rootkits : An introduction to Kernel Hacking”, San Francisco, No Starch Press, 2007.

[10] M. Russinovich and D. Solomon, “Microsoft Windows internals : Microsoft Windows server 2003, Windows XP, and Windows 2000”, Washington, Microsoft Press, 2005.

[11] J. Honeycutt, “Microsoft Windows Registry Guide”, Washington, Microsoft Press, 2005.

[12] H. Phillips and E. Skagerberg, “New perspectives on Microsoft Windows 2000 MS-DOS command line comprehensive enhanced”, Massachusetts, Course Technology, 2001.

[13] S. Anson and S. Bunting, “Mastering Windows Network Forensics and Investigation”, Indianapolis, Wiley, 2007.

[14] L. Dostálek and A. Kabelová, “Understanding TCP/IP : a clear and comprehensive guide to TCP/IP protocols”, Birmingham, England,  Packt Pub, 2006.

[15] B. Forouzan and S. Chung, “TCP/IP protocol suite 3rd edition”, Massachusetts, McGraw-Hill Higher Education, 2006.

[16] K. Jones, R. Bejtlich, C. Rose, “Real digital forensics: computer security and incident response”, New Jersey, Addison-Wesley, 2006.

[17] L. Columbus, “The Microsoft Windows XP professional handbook”, Massachusetts, Charles River Media, 2002.

[18] A. Deveriya, “Network administrators survival guide”, London, Pearson Education, 2005.

[19] W. Shipley, 1998. “Becoming a programmer for Windows is like becoming a dentist for Tyrannosaurus Rex” http://www.stepwise.com/Articles/Editorial/wjs_Windows.html (accessed December 20, 2008).

[20] R. Stross, “The Microsoft way : the real story of how the company outsmarts its competition”, Massachusetts, Addison-Wesley, 1996.

[21] R. Selby, “Microsoft secrets: how the world's most powerful software company creates technology, shapes market”, New York, Free Press, 1995.

[22] J. Edstrom and M. Eller, “Barbarians led by Bill Gates: Microsoft from the inside, how the world's richest corporation wields its power”, New York, Holt, 1998.

[23] M. Schwartz, 2004-2007. “Microsoft Versus” http://www.msversus.org (accessed January 10, 2009).

[24] Wikipedia, 2009 “List of mergers and acquisitions by Microsoft” http://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_Microsoft (accessed January 30, 2009).

[25] Wikipedia, 2009 “Forethought (company)” http://en.wikipedia.org/wiki/Forethought_(company) (accessed Janaury 30, 2009).

[26] Wikipedia, 2009 “Visio Corporation” http://en.wikipedia.org/wiki/Visio_Corporation (accessed Janaury 30, 2009).

[27] D. Greer et. al. “CyberInsecurity: The Cost of Monopoly How the Dominance of Microsoft’s Products Poses a Risk to Security” http://www.ccianet.org/papers/cyberinsecurity.pdf (accessed February 4, 2009).

[28] J. Hruska, 2007. “Microsoft to pay $179 million in Iowa antitrust settlement” http://arstechnica.com/business/news/2007/09/judge-approves-microsoft-iowa-settlement-fees.ars (accessed November 10, 2008).

[29] Inera Inc, 2007. “Word 2007 Scholarly Publishing Update” http://www.inera.com/word2007update.shtml (accessed November 29, 2008).

[30] S. Whitford, 2003. “Microsoft ad pulled by ASA” http://www.itweb.co.za/sections/business/2003/0303201315.asp?S=Software&A=SFT&O=google (accessed September 23, 2008).

[31] S. Cashman, “Microsoft Internet Explorer 6: Introductory Concepts and Techniques, Windows XP Edition”, Massachusetts, Course Technology, 2004.

[32] C. McNab, “Network security assessment”, California, O'Reilly Media, Inc., 2004.

[33] J. Richards, R. Allen, and A. Lowe-Norris, “Active Directory”, California, O'Reilly, 2006.

[34] J. Day, “Patterns in network architecture: a return to fundamentals”, New Jersey, Pearson Education, 2008.

[35] B. Smith and B. Komar, “Microsoft Windows security resource kit”, Washington, Microsoft Press, 2003.

[36] B. Price, J. Price, and S. Fenstermacher, “Mastering Active Directory for Windows Server 2003 R2”, Indiana, Wiley, 2006.

[37] R. Allen, “Active directory cookbook”, Massachusetts, O'Reilly, 2009.

[38] A. Basta and W. Halton, “Computer security and penetration testing”, Massachusetts, Thompson, 2008.

[39] A. Fadia, M. Zachar, “Network intrusion alert: an ethical hacking guide to intrusion detection”, Massachusetts, Course Technology, 2007.

[40] M. Jakobsson, Z. Ramzan, “Crimeware: understanding new attacks and defenses”, New Jersey, Addison-Wesley, 2008.

[41] C. Anley et. al., “The shellcoder's handbook: discovering and exploiting security holes” Idiana, Wiley Pub., 2007.

[42] W. Noon and I. Dubrawsky, “Firewall fundamentals”, Cisco Press, 2006.

[43] G. Donahue, “Network warrior”, California, O'Reilly Media, 2007.

[44]  P. Szor, “The art of computer virus research and defense”, New Jersey, Addison-Wesley, 2005.

[45] J. Biggs, “Black hat: misfits, criminals, and scammers in the Internet age”, New York, Apress, 2008.

[46] E. Skoudis with L. Zeltser, “Malware: fighting malicious code” New Jersey, Prentice Hall, 2004.

[47] J. Guisnel, “Cyberwars: espionage on the Internet”, New York, Plenum Trade, 1997.

[48] James S. Tiller, “The ethical hack: a framework for business value penetration testing”, Forida, Auerbach Publications, 2005.

[49] R. Bejtlich, “The Tao of network security monitoring: beyond intrusion detection”, Boston, Addison-Wesley, 2005.

[50] R. Tibbs and E.Oakes, “Firewalls and VPNs: principles and practices”, New Jersey, Pearson Prentice Hall, 2006.

[51] J. Aycock, “Computer viruses and malware” New York,  London,  Springer, 2006.

[52] R. Buschkes, P. Laskov, “Detection of intrusions and malware & vulnerability assessment : third international conference, DIMVA 2006”, Berlin, Germany, July 13-14, 2006: proceedings, New York,  Springer, 2006.

[53] E. Seagren, “Secure your network for free: using NMAP, Wireshark, Snort, Nessus, and MRTG”, Massachusetts, Elsevier Science, 2006.

[54] J. Aquilina, E. Casey, and C. Malin, “Malware forensics: investigating and analyzing malicious code” Massachusetts, Syngress Pub., 2008.

[55] K. Cox and C. Gerg, “Managing security with snort and IDS tools” California, Oreilly, 2004.

[56] J. McNamara, “Secrets of computer espionage: tactics and countermeasures”, Indiana, Wiley, 2003.

[57] S. Sinchak, “Hacking Windows Vista”, Indiana, Wiley, 2007.

[58] D. Sid, 2008, 2009. “OSSEC” http://www.ossec.net/main/about (accessed March 8, 2008).

[59] S. Gorman, 2009. “Electricity Grid in U.S. Penetrated By Spies” http://online.wsj.com/article_email/SB123914805204099085-lMyQjAxMDI5MzA5ODEwNDg4Wj.html (accessed April 15, 2009).

[60] 2007. “The cyber raiders hitting Estonia” http://news.bbc.co.uk/2/hi/europe/6665195.stm (accessed January 8, 2009).

[61] S. Gorman, A. Cole and Y. Dreazen, 2009. “Computer Spies Breach Fighter-Jet Project” http://online.wsj.com/article/SB124027491029837401.html (accessed April 26, 2009).