2005-09

Taxonomy of spyware and empirical study of network drive-by-downloads

Barwinski, Mark Andrei

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TAXONOMY OF SPYWARE AND EMPIRICAL STUDY OF NETWORK DRIVE-BY-DOWNLOADS

by

Mark Andrei Barwinski

September 2005

Thesis Advisor:	Cynthia E. Irvine
Co-Advisor:	Tim E. Levin

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Taxonomy Of Spyware And Empirical Study Of Network Drive-By-Downloads			**5. FUNDING NUMBERS**
6. AUTHOR(S) Mark Andrei Barwinski			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Spyware has rapidly become a major security concern in government and corporate networks as well as for home computers. Spyware is able to circumvent common security practices, funneling information to remote parties and consuming system resources with impunity. This malicious software has infiltrated common search engines and Internet sectors generally considered "safe." Making use of browser vulnerabilities, spyware infection is wide-spread.

This thesis considers common infection vectors and reviews current definitions in arriving at an improved definition of spyware. It identifies four common activities present in all spyware which lead to multiple behavioral capabilities. An empirical analysis of network drive-by-downloads shows the presence of spyware in bank, online travel, and real estate-related Internet sectors. The impact of system security patch maintenance on spyware susceptibility, and browser differences in the context of drive-by-downloads is also presented.

14. SUBJECT TERMS Spyware, Malware, Infection, Internet, Information Assurance.			**15. NUMBER OF PAGES** 182
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

TAXONOMY OF SPYWARE AND EMPIRICAL STUDY OF NETWORK DRIVE-BY-DOWNLOADS

Mark Andrei Barwinski
Civilian, Naval Postgraduate School
B.S., California State University Long Beach, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2005

Author: Mark Andrei Barwinski

Approved by: Dr. Cynthia E. Irvine
Thesis Co-Advisor

Tim E. Levin
Thesis Co-Advisor

Dr. Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Spyware has rapidly become a major security concern in government and corporate networks as well as for home computers. Spyware is able to circumvent common security practices, funneling information to remote parties and consuming system resources with impunity. This malicious software has infiltrated common search engines and Internet sectors generally considered “safe.” Making use of browser vulnerabilities, spyware infection is wide-spread.

This thesis considers common infection vectors and reviews current definitions in arriving at an improved definition of spyware. It identifies four common activities present in all spyware which lead to multiple behavioral capabilities. An empirical analysis of network drive-by-downloads shows the presence of spyware in bank, online travel, and real estate-related Internet sectors. The impact of system security patch maintenance on spyware susceptibility, and browser differences in the context of drive-by-downloads is also presented.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

GLOSSARY

API	Application Programming Interface.
ARPANET	Advanced Research Projects Agency Network.
ASCII	American Standard Code for Information Interchange.
BotNets	An army of compromised Internet-connected computers controlled by a remote attacker.
Cookie	Text files issued by a web server and stored in a client, used by web browsers to maintain session state information and identify users for web customization [3].
CPU	Central Processing Unit.
Daemons	A program that executes without human intervention to accomplish a task. Associated with Unix systems, equivalent to services in Windows systems.
DOS	A network attack designed to disrupt availability.
EULA	End User License Agreement.
FRAME	An HTML tag used for dividing a web page into separate and distinct sections.
Hacker	People proficient in computer programming and hardware inner workings who exploit systems through skill and tactics.
HTTP	HyperText Transfer Protocol. Used to create documents on the World Wide Web.
IFRAME	HTML tag which allows for inline frames, frames within a block of text.
IP	Internet Protocol.
Key logger	A computer program or hardware device used to capture keystrokes and initially used in diagnostics.
Libpcap	A system-independent interface used in network packet capture. Libpcap file formats are used by a variety of network traffic capture software.
NAME	HTML tag parameter used for identification purposes.
NAT	Network Address Translation.
PID	Process identifier.

Rootkit	A type of Trojan horse which hides itself, other files, and network connections. Runs at the kernel level of the machine and is able to intercept all API calls [35].
Services	In Windows, a program, routine, or process that performs a specific system task without user intervention.
Spam Remailer	Computer programs designed to forward SPAM by using compromised computer's bandwidth resources.
SPAM	Unsolicited email considered bulk mailing from a senders point of view and junk email from a receiver's point of view.
SRC	HTML tag parameter providing information on source of an image or other.
Surfing	The act of following hyperlinks or browsing Internet web pages.
TCP	Transmission Control Protocol.
Tracks	Spyware program category considered by some and associated with the collection of information maintained by various applications. For example, collecting recent opened documents or web browser history information.
Trojan horse	Malicious programs disguised as legitimate or benign software.
UDP	User Datagram Protocol.
UNICODE	An international standard of codes used to represent letters, numbers, control characters, and others in computers.
URL	Uniform Resource Locator.
Virus	Self-replicating program that propagates by inserting identical or mutated copies of itself into other executable code or documents.
Warez	Pirated commercial software made available to the public and for which copy protection or registration requirements have been disabled.
Web bug	Images placed inside web pages and html email by companies and organizations to track information about the viewer. Information collected may include IP address, hostname, operating system, web browser type, and date image was viewed [25, 50].
Worm	A self-contained self-replicating computer program similar to a virus which unlike a virus, does not need to be part of another program to propagate.
Zero-day-exploit	Computer program or technique used to exploit unpublished or little-known vulnerability in a computer system.

ACKNOWLEDGMENTS

To my wife and children, who sacrificed and endured so much through this journey. Thank you for your understanding and unwavering support. Calm seas ahead, skipper! Ready to tack?

To my brother and father, thank you for making those faithful decisions in 1971 and 1986. None of this would be possible without your foresight and encouragement.

I also would like to thank Dr. Cynthia Irvine and Prof. Tim Levin for their guidance, support, and patience, making this thesis experience an enjoyable and interesting one. I have learned much about computer science and myself in the process.

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0114018. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Finally, special thanks to Bob for your continued support and understanding.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Cyber attacks in the Internet have been increasing in both frequency and complexity, and a strong emphasis on wealth appropriation through illegal means has taken over the once ideals-driven or attention-driven hacker activities of the 1980's and 1990's. The spyware industry has flourished on both sides of the legal and ethical fence, and has thrived where legal issues are gray. It has become one of the greatest threats to cyberspace at a time when our society relies heavily on such network of networks. Continued vulnerability discoveries and limited user awareness, coupled with an urge to interconnect vulnerable systems to vulnerable networks have placed our personal privacy, financial infrastructure, and national secrets at risk.

> For the United States, the information technology revolution quietly changed the way business and government operate. Without a great deal of thought about security, the Nation shifted the control of essential processes in manufacturing, utilities, banking, and communications to networked computers [53].

The National Strategy To Secure Cyberspace released by the White House in February 2003 provides guidelines and encourages cooperation between government and private industry with the hope of curbing a troubling trend. As the report states, "Cyberspace has become the nervous system – the control system of our country."

It is this control system, which came under coordinated repeated attack in 2004, with the deployment of what became known as the Download.ject attack. Several web sites hosted by Microsoft's Internet Information Services Web server (IIS) were compromised. Although the exact number of businesses was not disclosed, it was reported that numerous large Internet institutions, including banks, insurance companies, auction outlets, and other main stream web sites were compromised [27]. The attack consisted of the addition of code which capitalized on Internet Explorer client vulnerabilities to install key logging and Trojan horse software in visitor's computers by the mere action of visiting the compromised web site. Similar attacks affecting Apache web servers were also noted during 2004 [4]. Sensitive information such as Social

Security Numbers, credit card numbers, user names and corresponding passwords, and encrypted communications between client browsers and financial institution web sites were targeted by these various attacks [4, 8, 52, 69].

Often, it has been said that high risk behavior on the Internet leads to infection by spyware, viruses, Trojan horses, key loggers and the like. The use of peer-to-peer file sharing networks, the downloading of freeware and shareware, and the visiting of hacker- or warez-related web sites, as well as adult entertainment and gambling-related web sites might be considered "high risk behavior" as it is generally considered to lead to increased risk of infection. However, there is ample evidence that the compromise of main-stream web sites can also lead to infection of their visitors while they conduct activities generally considered "safe" or "low risk." Little has been mentioned about relatively trusted and popular destinations of the Internet until the attacks of 2004.

The objective of this research is to better understand spyware by conducting a review of the various types of spyware, mode of infection, and associated spyware activities. Additionally, an evaluation of relatively "safe" main stream sectors of the Internet is conducted with four different test bed platforms consisting of default unpatched installations of Windows XP with Internet Explorer or Firefox Internet browsers, and fully patched Windows XP SP2 platforms with Internet Explorer or Firefox Internet browsers. By evaluating eight different "safe" Internet sectors in addition to three "high risk" Internet sectors, the intent is to provide a better understanding of the risks of one browser over another (a default Windows XP installation over a fully patched installation) and a ranking of several Internet sectors, in the context of spyware drive-by-download infection.

Chapter II provides background information on the various network technologies giving rise to spyware software development. The hardware and languages used in the execution of the experiments conducted as part of this work are also described.

Chapter III provides an analysis of current spyware definitions. It explains reasons for the difficulty in arriving at a consensus working definition of spyware among

security professionals and provides a new definition based on common activities noted among spyware programs. The chapter also discusses four main infection vectors associated with spyware.

Chapter IV provides a description of the basis of the experiment conducted as part of this work, including tool development, methodology, and implementation of experimental test beds. This chapter discusses the development of VBScripts to drive browser applications to URLs in various Internet sectors, the manner in which the URLs were collected and the tools used in the detection of spyware infection.

Chapter V provides a description of the actual experiment, a Internet web surfing simulation, including a step-by-step description of the experimental process. This chapter also describes experiment execution anomalies, including browser crashes, virtual machine crashes, and script crashes. Finally, the chapter provides a summary table of all the spyware detected in the various test beds on an Internet sector-per-sector basis.

Chapter VI provides an in-depth analysis of the experimental data. A detailed description of an attack experienced during the execution of the banking sector experiment is provided. Additional detailed discussions on infections are provided for the insurance, real estate, and online travel sectors of the Internet. A list of spyware-related binary files downloaded during the course of the experiment is also provided. Comparisons are made among the four platforms used in the experiment and the level of infection observed in each of them. Finally, the chapter provides a list of identified malicious web sites responsible for spyware infection.

Chapter VII describes three related spyware studies. The first was performed at the University of Washington, and consisted of network traffic analysis of the university network over a one-week period in 2003 in which spyware was found to have penetrated over 69% of the organizations within the university environment. The second pertains to ongoing research performed by Microsoft in the area of spyware infection detection and malicious web site identification through the use of web “monkeys.” The third study was conducted at the University of Nebraska-Lincoln in which, over a period of one month,

600 web sites from among four different Internet sectors were manually inspected for malicious spyware infection. Similarities and differences between each of these studies and work reported here are discussed.

Finally, Chapter VIII provides conclusions about this research and the data obtained from the experiments. Additionally, a discussion of future work is also provided.

II. BACKGROUND

This chapter provides general background information on various factors that have played a central role in the development or spread of spyware. Additionally, some of the technologies used in experiments described in Chapter IV are also described in this chapter.

A. INTERNET

The Advanced Research Projects Agency of the U.S. Department of Defense (ARPA) supported the development of a packet switched communications network during the early 1960's. Work by researchers at MIT, the Defense Advanced Research Projects Agency (DARPA), the RAND Group, and the National Physical Laboratories (NPL) lead to the development of the underlying technologies needed to realize Licklider's vision of the "Galactic Network" [30] and Roberts's plan for the ARPANET [39]. Interest grew in the development of a national communication infrastructure intended to be distributed and resilient to disruption [7]. The Network Measurement Center at the University of California Los Angeles and the Stanford Research Institute (SRI) became the first and second nodes of the ARPANET in 1969, respectively. By 1971, approximately two dozen research and government sites were interconnected. Soon additional nodes followed, realizing the network of networks and experiencing exponential growth over the next forty years. By 1990, the ARPANET was officially decommissioned by the Department of Defense, and the National Science Foundation took possession of the management of what became NSFNet, a network with over 100,000 computers.

The expansion of hypertext concepts by Nelson and Engelbart in the 1960's, followed by the first email application in 1972 [54], and the World Wide Web by Berners-Lee in 1990 lead to the current global network consisting of hundreds of millions of hosts.

B. COMPUTER NETWORKS

The simpler computer networks, as envisioned in the 1960's and 1970's [30, 46], evolved into complex inter connected systems. These networks soon became

indispensable and extremely valuable assets, attracting individuals interested in exploiting or disrupting the technology in a variety of ways. The following is a brief description of the network technologies relevant to spyware and the experiment described in Chapter IV.

1. Firewalls

The Morris Worm[1] in 1988 [47] led to the realization that the Internet was no longer a closed community where all participants knew and trusted each other. Although firewalls were in use by the late 1980's in segmenting local area networks, by the early 1990's, the first security firewalls were beginning to filter packets based on rule sets. Firewalls were initially intended to have the following properties: function as a single point between two or more networks, control and authenticate traffic through the firewall device, and log network traffic [37].

Most firewalls are configured in a manner to keep malicious traffic out of a local area network, but not to keep the local area network traffic from reaching outside nodes. Generally this is a desirable property intended to maintain a degree of protection while maximizing access to outside resources. Recently firewalls have added egress content filtering functionality and filter on specific URLs or keywords. Such features prevent certain types of web sites from being visited or clear text content from being released outside the local area network.

Yet firewall technology has been ineffective in preventing infection by spyware for reasons to be discussed in Chapter III.

2. Network Address Translation (NAT)

In 1994, the first RFC document describing Network Address Translation (NAT) was published [13]. As early as 1992, it was becoming apparent that there would be a shortage of Internet Protocol (IP) addresses [23]. Although IP Version 4 was capable of uniquely identifying 4.4 billion devices, the impending explosion in deployment of communication devices requiring IP addresses warranted a solution. NAT was originally

[1] Robert T. Morris was a student at Cornell University. He wrote what was considered the first worm, a self replicating program which used various vulnerabilities to copy itself to other computers, begin execution of the program and search for additional computers to infect. At the time, approximately 88,000 computers were connected to the ARPANET and a significant portion of the network was affected.

envisioned as a way to "provide temporary relief while other, more complex and far-reaching solutions are worked out." Over a decade later, NAT has a near-ubiquitous deployment throughout the Internet.

NAT allows IP addresses to be shared across numerous devices while remaining mostly transparent to end-to-end protocol interaction.

Sharing of these IP addresses is performed by translating IP address headers, in particular IP addresses and ports. During transmission of packets from the inside of a NAT device towards the outside, the NAT device rewrites the source address in the packet header with a different value, and adjusts the IP and TCP header checksums accordingly to reflect the changes made to address fields. When a packet travels in the reverse direction, meaning originating outside the NAT device and destined towards a device located on the inside, the destination address reconstituted and the corresponding checksums are recalculated. While NAT has various behavior modes, when configured in the symmetric mode, the translation of header information on outbound and inbound packets has a net effect of filtering network traffic. When translating outgoing packet addresses, NAT replaces the local private source address with the public source address and proceeds to create a temporary alias between this public address and the private device address. A local session state is established in the NAT device for mapping between private and public addresses. Incoming packets from public addresses are inspected for valid destination addresses. If a corresponding private address and public address mapping is found, the packet header is modified and forwarded to the appropriate device. If no such mapping is found, the packet is discarded.

This symmetric NAT configuration filters out all inbound traffic not associated with an initial outbound connection. The router used in the experiment described in Chapter IV was configured in this manner, intended to limit infection vectors to only those associated with web traffic as a result of initially visiting a specific URL. Figure 1 depicts such a scenario where network traffic is initiated at Host A and destined for Host B. The NAT device notes the initial traffic and binds communication between itself and

Host B. Subsequent network traffic is allowed through based on this established binding while traffic originating from a different communication port or host is denied access to Host A.

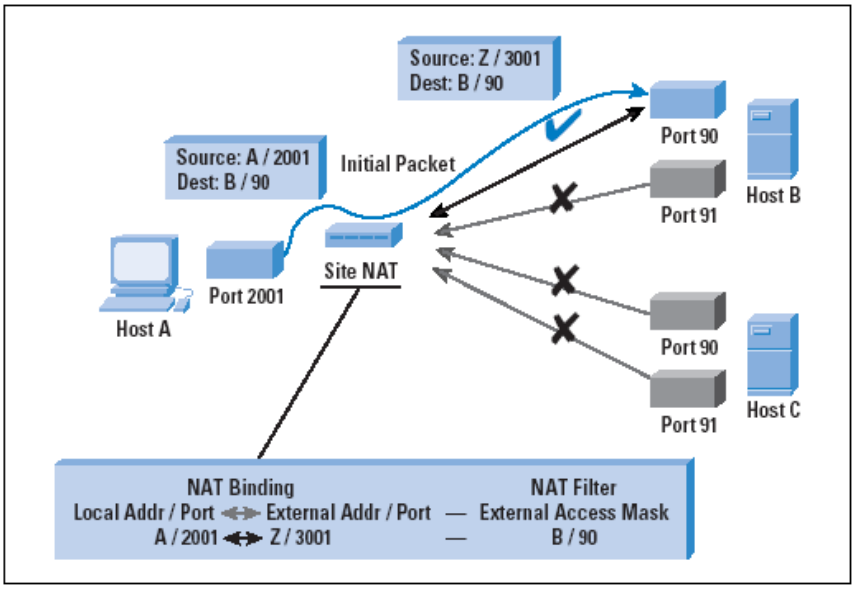


Figure 1 Symmetric NAT. From [23].

C. VIRTUAL MACHINES

VMWare Workstation is commercially available desktop software for running multiple x86-based operating systems within a single physical personal computer. The operating system on which VMWare Workstation runs is referred to as the Host Operating System while the individual images operating within VMWare are referred to as the Guest Operating Systems. The guest operating systems can be configured as a virtual local area network or bridged to interact with the outside world. VMWare achieves this by creating fully isolated virtual machines. Each virtual machine encapsulates the guest operating system and all its applications. A virtualization layer maps actual physical hardware resources to the virtual machine's resources.

VMWare software was used in the experiment described in Chapter IV.

D. INTERNET BROWSERS

With the development of HTML and the network communications infrastructure provided by the Internet, the stage was set for the creation of Internet browsers. The roots of hypertext can be found in work performed by Vannevar Bush in 1945 [24].

Nelson and Engelbart further advanced the concept in the 1960's and finally, Tim Berners-Lee used this groundwork to develop a standard language for screen content layout known as Hypertext Markup Language (HTML), and a program for displaying of HTML called a "browser."[2] The first widely used web browser was developed by Marc Andreesen and Eric Bina, with the release on NCSA Mosaic for X-Windows on Unix computers in 1993. Later that year a Macintosh version was developed and by 1994, Spyglass Inc. was awarded commercial rights and subsequently licensed the technology to several companies, Microsoft among them. In October of 1994, Netscape released the first beta version of its browser, known as Mozilla 0.96b. It soon became the most popular browser on the Internet. By August of 1995, Microsoft released Internet Explorer for Windows 95, which would overtake Netscape's lead as the most popular browser by 1999.

New features were continually added to the browser applications. Support for JavaScript and Java applets was provided by Netscape in 1995. By 1996, Internet Explorer provided the first truly useful implementation of Cascading Style Sheets (CSS) and by 1997, the Document Object Model (DOM) was supported by Internet Explorer, thus allowing Javascript to implement dynamic content.

Although market share numbers vary dramatically depending on the source, it is clear that Internet Explorer is the most popular browser with approximately 88% market share as of April of 2005. On the other hand, it is estimated Mozilla's Firefox browser holds approximately 6.75% of the market. Worthy of mention however is that since becoming available on November 9, 2004, Firefox reported 50 million downloads by April 29, 2005 [16]. For these reasons, Internet Explorer and Firefox browsers were selected for the experiment described in Chapter IV They represent both the most popular and the fastest growing browsers in use, respectively. Figure 2 depicts a market share comparison among Internet Explorer, Firefox, and other non-Firefox browsers between June of 2004 and May of 2005.

[2] WorldWideWeb browser was officially demonstrated by Tim Berners-Lee to CERN employees in March 1991.

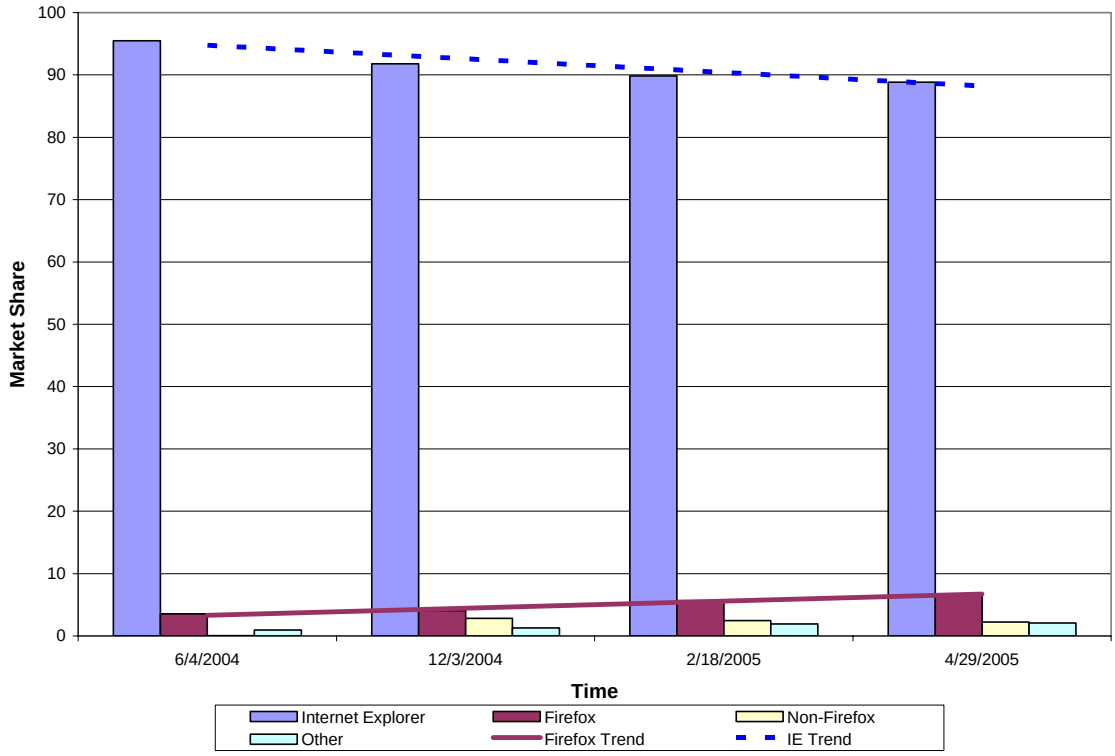


Figure 2 U.S. Browser Market Share. After [67]

E. INTERNET BROWSER EXPLOITS

The Computer Emergency Response Team (CERT) at the Carnegie Mellon Software Engineering Institute has been tracking the number of reported software vulnerabilities and computer incidents since 1988. Figure 3 depicts trends in both vulnerabilities and incidents between 1988 and 2004. Although no Internet browser-specific statistics are reported here, the figure illustrates the state of the software industry. Shown in dark colored bars, is the number of vulnerabilities reported each year (x-axis), reported vulnerabilities in thousands along the left side (y-axis). The number of incidents is shown over the same x-axis with a y-axis along the right side showing reported incidents in the thousands. This figure shows a dramatic rise in both, vulnerabilities and consequently, reported incidents since 1999. Internet browsers are expected to be part of this trend.

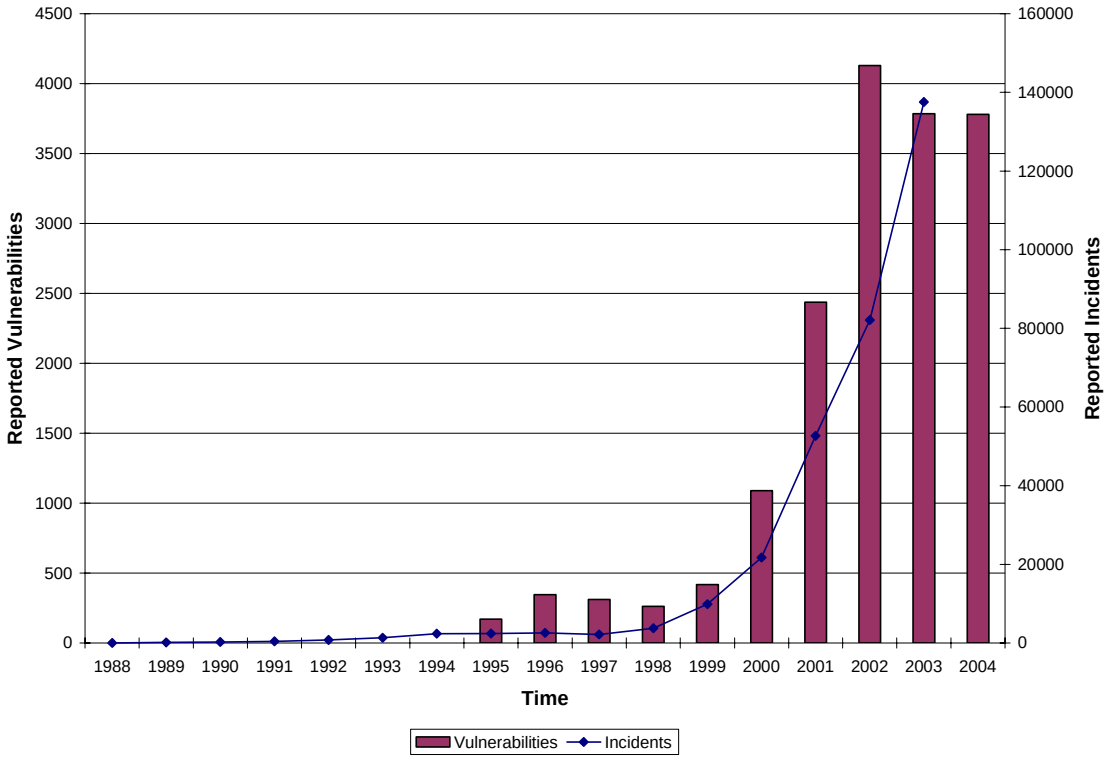


Figure 3 Rise of Reported Vulnerabilities and Incidents. After [5]

The sections that follow will present a limited description of only a few of the numerous vulnerabilities discovered in Microsoft Internet Explorer. These vulnerabilities are discussed here due to their relevance as a spyware infection vector during the course of Internet surfing activities. While the vulnerabilities discussed apply to Microsoft products such as Internet Explorer, Microsoft Outlook and Outlook Express, similar vulnerabilities have been discovered in alternative browsers such as Netscape and Firefox. These vulnerabilities may give administrator or root privileges to an attacker.

1. IFRAME and FRAME Tag Vulnerabilities

IFRAME and FRAME tag vulnerabilities pertain to a family of vulnerabilities associated with the mishandling of HTML tags. Many of the vulnerabilities associated with these tags are the result of improper boundary checking or error handling leading to buffer overflow conditions. If a malicious web site developer generates overly long SRC and NAME attributes within the HTML and combines this with a specially crafted script

designed to prepare a heap block, Microsoft Internet Explorer can be triggered to dereference memory addresses which may fall within such previously prepared heap block. This may cause the browser application to execute an attacker's shell code with the same privileges as the current user [58]. Variant attacks may cause the browser to crash or misrepresent the source of a web site as reported under the status bar of the browser display [57].

2. Cross Domain Scripting Vulnerabilities

Cross-domain vulnerabilities pertain to the manner in which Microsoft Internet Explorer determines the security zone of a particular browser frame [59]. Under certain circumstances, Internet Explorer does not correctly validate the security context of a web server redirected frame. This condition can be exploited to allow a malicious script to be evaluated in a less restrictive security domain than it would otherwise be. If such a malicious script is evaluated in the Local Machine Zone, it would execute arbitrary code with the same privileges as the user currently logged into the system [14, 19, 45].

3. Drive-by-download

Drive-by-downloads is the term generally used for the category of computer programs which are able to install themselves without user authorization or awareness during the course of visiting a web site. The installation of these programs is associated with the exploitation of IFRAME, FRAME, and cross-domain vulnerabilities in such a manner so as to trigger the execution of a malicious program at a victim computer. This malicious program may be an installer program designed to deliver a spyware payload and facilitate further access to the compromised system. IFRAME, FRAME, and cross-domain vulnerabilities are only a few of numerous ways a browser may be forced to execute arbitrary code by an attacker, and are mentioned in this section for illustrative purposes only.

F. WEB SEALS

Web seals of approval or trustmarks, as they can sometimes be referred to, are a self-regulatory accreditation scheme initially established to promote good online practices. These seals of approval were also intended to promote consumer confidence in e-commerce transactions and privacy on the Internet. In September of 1999, the issue of Web privacy was raised at the 21st Conference of International Data Protection

Commissioners [66]. As a result of this, a joint project was undertaken by the Office of the Information and Privacy Commissioner in Ontario, Canada, and the Office of the Federal Privacy Commissioner of Australia. This joint effort lead to the preparation of a report entitled "Web Seals: A Review of Online Privacy Programs," presented at the 22nd International Conference on Privacy and Personal Data Protection on September of 2000. An additional study was conducted by the Standing Committee of Officials of Consumer Affairs E-commerce Working Party, presenting their findings report in January of 2005 [65]. These reports evaluated such leading online privacy seals as BBBOnLine, developed by the United States Council of Better Business Bureaus; TRUSTe, originally founded by the Electronic Frontier Foundation and the CommerceNet Consortium and funded by such companies as AOL, Intel, Microsoft, to name a few; and WebTrust, originally developed by the American Institute of Certified Public Accountants (AICPA) and the Canadian Institute of Chartered Accountants (CICA). Both reports concluded that although security and privacy are one of the top concerns in consumer survey responses, consumers remain generally unaware of web seal programs. Furthermore, it is concluded that the success of the web seals remains unclear.

Therefore, when designing the experiment described in Chapter IV, web seals were not considered as a significantly gauge of legitimacy or trust placed on a specific web site.

G. SCRIPTING LANGUAGES

A variety of technologies have facilitated functionality by allowing programs to interact with one another making their methods available and thus providing services to applications. Other scripting technologies offer interaction with users over the Internet, providing dynamic content or feature-rich experiences. The following is a brief description of some of the scripting languages and mobile code mechanisms that can facilitate attacks. Due to the integration of some or all of these technologies in commonly used browsers (e.g., Microsoft Internet Explorer, Firefox), and the wide spread presence of these products, a large, readily available, pool of prospective victim systems is vulnerable to attack.

1. ActiveX

The term ActiveX pertains to a variety of technologies under what has become an umbrella Microsoft branding name [6]. ActiveX makes use of Microsoft's Component Object Model (COM) – a packaging technology providing a group of conventions and supporting libraries which facilitate interaction among a variety of application components in a consistent and object-oriented manner. These COM objects may be written in C++, Java, Visual Basic, and other languages, and are implemented as Dynamic Link Libraries (DLLs) or executable files. COM objects expose their methods to other applications via vtable interfaces, as in the case of C++ written clients, or via dispatch interfaces, as in the case of clients written in Visual Basic or other simpler languages. The later approach to exposing component methods became known as "automation," a technology used in the development of an experiment described in Chapter IV.

A subset of these technologies became known as ActiveX controls, those components intended for desktop use, using automation, and implemented as DLLs. Therefore, ActiveX controls cannot be directly executed but, instead, require a container, such as a scripting language or an application like a web browser – i.e., Internet Explorer. Part of the allure of ActiveX controls is that it allows the development of applications in a manner similar to the assembly of electronics – through pre-built component parts, while maintaining the ability to easily download and execute these components over a network. ActiveX controls may be invoked remotely and are thus considered mobile code.

Unlike Java, ActiveX does not execute inside a "sandbox" protected environment. Instead, ActiveX code is native code executing directly on a physical machine. As such, it is able to access services and resources not generally made available to code running in a restricted environment [38]. Additionally, ActiveX controls do not require registration on a user-per-user basis in each computer but instead need only be registered once under any local user account and are then available to all user accounts in a computer. This may begin to explain the presence of malicious software across user accounts in a single computer system.

Microsoft developed a code signing feature for ActiveX controls. However, code signing simply provides verification that the control was in-fact produced by the signer and that it has not been modified. Code signing does not ensure the behavior of the code, its benevolence, trustworthiness or the competence of the author. This signing process simply binds the code to the author without necessarily providing increased confidence in the code. Additionally, Microsoft provides very little flexibility regarding the manner this code may be executed. It either may be forbidden to execute in a system, or it may be granted full access. This coarse-grained security approach leaves users in a position where functionality and security are directly opposing one another. When ActiveX controls are allowed to execute, they usually execute with the same privileges as the current user. Since these controls may be remotely invoked, a network channel is readily available for an attacker.

One final comment regarding ActiveX code signing is that since there is no assurance as to the correctness of the code, vulnerabilities in signed code may be exploited by attackers. The "repurposing" of "legitimately signed" code by means of scripting provides exploitation opportunities while hiding behind the perceived trust associated with a script's signature.

It is worth noting that ActiveX technology is now used by a variety of third-party applications. Therefore, vulnerabilities and exploitation opportunities provided by ActiveX may not be limited within the realm of Microsoft Internet Explorer, Microsoft Outlook, or Microsoft Outlook Express, but instead may open the door to a system via many other infection vectors.

2. Java

Java is both an object-oriented programming language and a platform. Originally developed by James Gosling and colleagues at Sun Microsystems, it was designed from the ground up to emphasize portability and security. It is both a compiled and an interpreted language. Java code is compiled once on any platform containing a Java compiler, and it is interpreted each time the program is executed on any hardware platform containing a Java interpreter. A Java interpreter consists of a Java Virtual

Machine providing a layer between hardware and software, thus giving rise to the concept of the secure Java "sandbox." The sandbox prevents misbehaving or malicious code from damaging the system.

The Java security sandbox consists of three main components – the Byte Code Verifier, the Applet Class Loader, and the Security Manager. The Byte Code Verifier ensures that the byte code to be executed complies with the predefined set of rules, that the format is correct, and that pointers and access restrictions are consistent. The Applet Class Loader ensures that Java classes can be added to a running Java environment. Finally, the Security Manager performs run-time checks on dangerous methods and grants higher privileges to built-in classes as opposed to remotely loaded classes.

As with ActiveX, Java has also been found to be exploitable in both Sun Microsystems's and Microsoft's implementations of Java Virtual Machines [55, 60].

3. JavaScript and JScript

Javascript is a computer language unrelated to the Java computer language. It was designed for Internet browsers and intended to allow the generation of small scripts to be imbedded inside HTML [17]. Javascript is an object-based interpreted language which supports event-driven programming. It has access to the browser DOM and is limited to executing within the browser application. However, Microsoft's implementation of Javascript, called Jscript, is able to execute outside browser applications by using the Windows Scripting Host environment. This allows JScript to directly access the file system, hard drives and printers, unlike Javascript. The Javascript language is used to improve the web surfing experience of users by providing simple animations, validation of forms, redirection of browsers, display of pop-up messages and detection of browsers and plug-ins, allowing customization of web sites to individual visiting users. The JScript language is primarily devoted to the execution of administrative-related scripts assisting with repetitive tasks as in the case of configuring desktops, managing the Windows file system and system resources, and administering user accounts.

Malicious web sites can use Javascript to launch attacks involving other technologies such as ActiveX. A combination of these technologies can grant full control of a compromised system to an attacker.

4. VBScript

VBScript is a scripting language closely related to Visual Basic and Visual Basic for Applications, and developed by Microsoft. VBScript operates within the Windows Scripting Host (WSH) and can be used as a stand-alone scripting language in system administration tasks or interact with such applications as browsers via the 32-bit and 16-bit Windows Application Programming Interface (API). The WSH creates an environment in which compliant scripts may execute. WSH is built into Microsoft Windows operating systems beginning with Windows 98.

VBScript can be executed as a client-side or a server-side scripting language used in the development of active content web sites. It is the default language for Active Server Pages (ASP), a Microsoft technology which allows dynamic content generation in web sites.

As with other technologies, VBScript has been used in Windows attacks, most famously perhaps with the creation of a VBScript worm named "The Love Bug" (also known as "ILOVEYOU")[3].

H. FREEWARE IMPACT ON SPYWARE

In 1982, Jim Knopf and Andrew Fluegelman embarked on a new software marketing and distribution approach. Jim Knopf developed "Easy File," later renamed PC-File, a small database application written in Applesoft BASIC. Andrew Fluegelman developed "PC-Talk," a communications program. Both decided, initially separately, to include in the documentation for the software a request for donations to help support and defer costs associated with software distribution and maintenance. These first two software programs were said to be Freeware, a term coined by Andrew Fluegelman [26].

[3] ILOVEYOU is both a virus and a worm which, o n May 4, 2000, spread around the world in a matter of hours. Over 20 government agencies and businesses such as AT&T, TWA, Ford Motor Company, Washington Post, ABC and others were affected causing damages ranging from $100 million to $10 billion [61].

In time the meaning of the term Freeware gave way to the term Shareware, software openly distributed and generally fully functional which requested voluntary donations.

By 1987 the concept of shareware was well established with numerous programs being marketed in this fashion and generating millions of dollars in profits for their developers. Through the years, shareware distribution transitioned from bulleting board systems and shareware disk vendors to CD-ROMs and, eventually to web sites which accepted credit card transactions.

By the 1990's, with the exponential growth of the Internet and increased connection speeds, shareware distribution expanded dramatically. At that time, Eudora was one of the first email applications to become truly popular on the Internet. First developed by Steve Dorner in 1988, it was eventually purchased by Qualcomm in 1994. Eudora soon found itself loosing ground to freely distributed email clients provided by Netscape and Internet Explorer. As with many other software programs, Eudora moved to what became know as ad-supported software distribution in an effort to maintain revenues. Under this model, software developers sell advertisement space within application windows. Qualcomm first announced the release of full-featured ad-supported Eudora software in December of 1999. This was seen as the first adoption of this model by a significant software developer. Until then, smaller software developers had experimented with this distribution model. Around the same time the freeware game "Elf Bowling" gained notoriety when it was reported to be bundled with tracking software, and transmitting user information to Nsoft, the game creator. By March of 2000, privacy concerns and difficulties uninstalling intrusive software were being voiced in such publications as The New York Times [20]. During this period it appears that ad-supported software and associated tracking modules, initially developed for the purpose of serving up customized advertisements, became independent of those legitimate full-featured software products. Also in March of 2000, the technology financial markets reached their peak, followed by what would become known as the bursting of the "dot-com bubble," a period of rapid and prolonged market corrections. These various factors have likely set the stage for the development of spyware as a significant industry.

I. CONCLUSION

This chapter provided a brief introduction to some of the concepts related to computer networks and software vulnerabilities associated with the development and propagation of spyware software. Additionally, a brief explanation of the genesis of spyware and adware software can be found in the early development of the freeware and shareware software distribution concepts which, by the late 1990's, lead to the development of ad-supported software. The next chapter provides a taxonomy of spyware software.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SPYWARE SOFTWARE

The term “spyware” was first used to refer to small video cameras [51]. By 16 October 1995, newsgroup forums began distributing a joke written as a C++ program in which the library header file spyware.h was used to partially describe Microsoft Corporation’s business model [70]. Subsequent postings in various newsgroups started using the term soon there after to describe malicious software not fitting squarely within the definition of a virus program. Steve Gibson is credited with writing the first anti-spyware program in 2000 [51] and since then, spyware has exploded into a multi-million dollar industry, both on the detection side as well as on the data gathering and actual “spying” side.

This chapter begins by discussing some of the definitions used so far to describe spyware and the difficulties encountered along the way in defining spyware. It then highlights reasons for that difficulty and offers an alternative definition. The chapter also discusses common qualities found in spyware, infection vectors, and activities associated with these programs.

A. CURRENT DEFINITIONS

The term “Spyware” has proven particularly difficult to define over the past few years. The Federal Trade Commission hosted a one-day workshop on April 19, 2004 to discuss issues associated with the distribution and effects of such software, which collects personal or organizational information and that is forwarded to another entity without the consumer’s knowledge or consent. It was the intent of the FTC to better understand “information practices of the online marketplace and their impact on consumers” as well as to promote discussion of the subject among government and industry [51]. The FTC report acknowledged difficulties in arriving at a working definition of spyware and presented a brief definition provided by Steve Gibson, instead. The report itself makes no new assertions as to the definition of spyware. Statements made by various panelists indicate the complexity and dynamic nature of software as a hindrance in the inability to reach an industry consensus on the definition of spyware.

At the heart of the problem has been the fact that spyware is unlike viruses, Trojan horses, and worms. In fact, it can utilize all three and more at various times. It can also be used for legal and illegal purposes and thus has a component of intention of use or deployment not present in Trojans, viruses, and worms, where malicious intent is clearly discernible. In 2000, spyware was first defined as follows:

> Any software that employs a user's internet connection in the background (the so-called 'backchannel') without their knowledge or explicit permission [18].

An alternative definition offered by Microsoft more recently is as follows:

> Spyware is a general term used for software that performs certain behaviors such as advertising, collecting personal information, or changing the configuration of your computer, generally without appropriately obtaining your consent [68].

Unfortunately, legislators and industry have become bogged down in defining spyware because of the fear that too broad a definition would encompass such software as Windows AutoUpdate and anti-virus signature update software – software with a clear benefit to users, and that too narrow a definition would fail to catch anything but the most egregious instances of spyware software.

The definition of spyware varies greatly among security professionals. At one end of the spectrum, spyware is limited to the actual collection of personally identifiable information, key logging, and password stealing. At the other end of the spectrum, spyware has been defined as software collecting practically any information from a system and forwarding this to a third party in a manner unknown to the computer user. Examples of the latter definition would include the previously mentioned Microsoft AutoUpdate and anti-virus updating utilities as well as web bugs, cookies, and browsing-habit monitoring applications. Unfortunately, attempts to better define spyware have lead to an explosion of confusing terminology including such terms as snoopware, scumware, junkware, thiefware, parasite software, undesirable software, and others. Since the term spyware is fairly well entrenched in security circles and known by the general population, albeit poorly defined, it will be used in reference to the whole software genus until a more precise definition is provided by this author.

1. Convergence Obstacle

Contributing to the problem in defining spyware is the issue of converging technologies. In the early days, viruses replicated code. They evolved to replicate code through a variety of mediums and subsequently modified or morphed their signatures by changing the code being replicated. But at their very essence, viruses simply replicate their code. Computer worms are similar in nature in that they burrow through networks also replicating their payload or code. Unlike viruses, worms are self contained and do not rely on other programs, files or documents to propagate.

In the case of spyware, however, full-fledged applications are installed onto computer systems via a variety of insertion vectors. In addition, these applications are far more feature and capability-rich than viruses or worms. Yet, the very same techniques used for malicious purposes by spyware writers are also used by a variety of legitimate business, providing enhanced features to computer users. Herein lies the conundrum, if a definition of spyware is to be arrived at, it cannot be solely based on activities or techniques used in the software but must consider intent behind its development and deployment.

To illustrate the point, consider the case of the previously mentioned Microsoft AutoUpdate software. This software collects or maintains information on the patch level of the host system and at regular intervals – when configured to do so – will connect to Microsoft servers and will download available patches or fixes for the operating system. The benefit to the user appears to be clear in this case, yet common underlying behaviors exist between this program and spyware programs, which also collect information and have the ability to download updates or additional applications onto a computer system. Both programs do this without necessarily informing the computer user about their activities, and both programs operate in the background. The case is similar for anti-virus programs which contact virus signature servers at regular intervals to maintain an up-to-date virus definition file.

At its essence, the convergence obstacle in defining spyware can be summarized in that legitimate and illegitimate, and desirable and undesirable software programs utilize the same common set of activities. These activities consist of the ability to operate

in the background, collect information, communicate this information to a third party, and maintain a presence in a computer system. These are discussed later in this chapter as spyware's four basic activities – hide, collect, communicate, and survive. It is the convergence of technologies possessing these four activities that lead to software packages assembled to compromise networks and hosts, in the form of spyware. These software packages may contain virus, worm, and Trojan horse-like qualities or behaviors.

2. A Note on Confinement

An interesting consideration when discussing the spyware problem is that it largely derives from the inability of a discretionary access control system to address the confinement problem [28]. Since spyware is able to operate within the same access rights as the user, a key logger, for example, is able to intercept keystrokes destined to a specific application and make them available to other programs. Mandatory access control was developed to address the confinement problem, but, covert-channels not withstanding, as commercial systems have not incorporated multilevel security to date, spyware may use many impersonation and Trojan horse techniques to exfiltrate information or steal resources. For example hijacking of bandwidth-rich applications such as Internet browsers allows spyware to communicate through firewalls, even when these are configured with egress filtering of applications or ports. Stateful firewalls are not sufficient in containing outbound fleeing information. By performing packet payload analysis, however, it may be possible to contain a subset of outbound information – as long as it is not encrypted.

It is this leakage of information with general impunity that is of concern not only to privacy advocates but also in relation to corporate trade secrets and national security.

B. SPYWARE BASIC ACTIVITIES

Spyware modus operandi consists of four basic activities which are common to all. From these four basic activities, spyware extends numerous both legal and illegal capabilities. These activities are described as follows:

1. Hide

Spyware software must be able to hide, at least in part, the mechanisms associated with its installation, execution, data collection, or communication. In legitimate programs, "hiding" can be seen as the desirable aspect of staying out of the user's way.

Installer programs and execution of such programs may be hidden by the exploitation of various system vulnerabilities. Chapter VI discusses just such an event in which spyware is surreptitiously installed onto an experiment test bed.

The name of the program may be chosen to hide its presence. It may appear to be an operating system process or service. More advanced techniques allow spyware processes and services to be invisible to process and service reporting applications, thus fitting the definition of "rootkits."

Data collection may be hidden by encrypting files, storing the data temporarily in alternate data streams (e.g., in Windows NTFS file systems), or in unallocated sectors of the hard drive. Data and spyware code may also be hidden by the use of random file names in folders known to contain thousands of files, such as system or browser cache directories, thus hiding in a crowd. Data or actual code may also be stored and hidden inside the Windows system registry [29].

Finally, spyware communications may be hidden by encrypting the transmission, by performing sparse, limited transmissions, or by compromising a legitimate application which possesses Internet access, and transmitting via such application (e.g., Internet web browsers).

2. Collect

Spyware must also be able to collect information. This information may be in the form of non-identifiable market demographics, personally-identifiable demographics, and targeted information. Each of these three is discussed in detail later in this chapter.

3. Communicate

Communication and sharing of collected information is vital to spyware. Communication channels may include high speed wired and wireless network connections such as Ethernet, wireless 802.11, and Bluetooth technologies, as well as dial-up modem connections.

4. Survive

The last basic activity of spyware is its ability to survive. Spyware resides in a hostile environment. Implicit in the desire to remain undetected is the consequence that when detected, users will act to remove it. There is a high likelihood attempts will be made to remove or disable the spyware software at some time during the deployment and maintenance phase of its lifecycle. Therefore, spyware must be resilient to remain within the compromised system for as long as possible.

This resiliency may be accomplished through a variety of techniques, some of which are briefly discussed in the following paragraphs. First, several mechanisms used to start spyware on the user's computer are described. Resiliency to removal may include utilizing multiple startup mechanisms so that if one is removed, others may be used.

a. Start-up Folders

Start-up folders are special folders in Windows systems where a user may place program or document shortcuts. The content of these folders is executed during the boot-up process. The following is a list of folders containing this characteristic:

Windows 95/98/ME:

%windows directory%\Main menu\Programs\StartUp\
%windows directory%\All Users\Main menu\Programs\StartUp\

Windows 2000/XP:

\\Documents and Settings\[user name]\Start Menu\Programs\Startup\
\\Documents and Settings\All Users\Start Menu\Programs\Startup\

Additionally, these folders are declared inside the Windows registry. By modifying the entries in the following two registry keys, a malicious user could place spyware in a hidden folder anywhere in the system and have it execute every time the system is booted.

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

b. Autorun Registry Keys

The Windows operating system registry possesses numerous ways to start programs during the boot-up, login, logoff, and shutdown process. Spyware can modify these registry keys to include a path to its installer or launcher program, thus allowing it to reinstall itself or remain active in a system. New autorun registry keys are still being discovered even by Microsoft itself. According to [64], 34 different registry keys could be used to start programs and an additional 17 previously unknown registry keys were discovered during the course of their experiments. The following is a limited list of some of the more common registry keys:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_USERS\.DEFAULT\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_USERS\.DEFAULT\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit

c. Services or Daemons

Window services or Unix daemons can be used to launch spyware or maintain control over a compromised system. This allows for spyware to be executed prior to the user logon process and conceivably prior to execution of spyware countermeasures.

d. Autoexec.bat and Initialization files

Legacy files no longer used by Windows XP can be exploited to launch spyware in older systems like Windows 95/98/ME/NT. These files may include config.sys, autoexec.bat, win.ini, system.ini, wininit.ini, and config.sys. Load and Run commands in some of these files allow the indiscriminate execution of programs.

e. Browser Helper Objects (BHO)

Browser Helper Objects extend the functionality of Internet Explorer by adding in-process Component Object Model (COM) components to the browser so that each time the browser starts, it exhibits this newly gained functionality. Spyware writers

have made use of BHO to launch and attach spyware to the browser application and since BHO have full access to the browser COM, spyware gains Internet access by impersonating the browser.

f. Interlocks

Software interlocks are mechanisms which restrict the actions that can be taken on a specific program. For example, in the case of spyware, interlocks are used to monitor the state or presence of the various processes or dynamic link libraries associated with its normal operation. If any one of these components is shutdown by a user, the remaining components detect the action and respond accordingly. This technique prevents users from manually disabling or removing spyware.

g. Hooks

In Microsoft Windows operating systems, system hooks are defined as functions which intercept system messages, mouse actions, or keystrokes destined to a specific application. The function intercepting these events may choose to forward them along to the corresponding application, discard them, or execute any operation it wishes based on this triggering event. This allows spyware writers to intercept system calls or actions which may attempt to reset settings previously modified by spyware or, which may attempt to modify or remove the spyware program itself.

C. ALTERNATIVE DEFINITION

Spyware borrows many qualities from already clearly defined and well understood malicious software such as viruses, worms, Trojan horses and backdoors. Similarly to the words used by Supreme Court Justice Stewart when considering a definition for hard-core pornography, “I know it when I see it” [56], “I know it when I experience it” appears to apply well in recognizing spyware. However, this does not suffice, and thus the following are considered in forming a better definition of spyware:

1. Development Intent

The intent behind the development or use of the software and the manner in which it is deployed is worthy of consideration. However, even though the software was developed for commercial and legal purposes, the key here is whether or not it is designed to monitor or steal resources from users. Therefore, software developed with

the express purpose of gathering information on a user, and obtaining demographic information (personally identifiable or not) or targeted specific information, should be considered spyware.

It is conceivable that as in the case of cookies and web bugs, which can be misused to achieve spyware-like results, programs not initially designed as spyware could also be misused to provide user information. Such programs would therefore not be considered spyware unless they also meet further criteria.

2. Profit Intent

The remote party must intend to use the information for financial gain through either legal or illegal means, or for strategic advantage as in the case of national security concerns.

3. Evidence of Basic Four Activities

The presence of the four basic activities of spyware – hide, collect, communicate, and survive – is a strong factor in labeling a program as spyware.

These four basic activities obviously may vary in emphasis among programs, changing the focus of the application according to the emphasis placed by the developer on each of these four basic activities. For example, applications with a low emphasis in the collection and communication of data may be likely to fall within a pop-up or general adverting server category. Conversely, a program with a high emphasis in collection and communication of data would be likely considered a system monitor such as a screen capturing device or a key logger application. Finally, an application with a high emphasis on hiding, communication and survivability, and low emphasis on data collection may be indicative of a resource stealing application as in the case of distributed computing schemes.

Figure 4 provides an illustration of spyware uses among the marketing, surveillance, and resource consuming industries. While spyware may be implemented by software or hardware, the following discussion is limited to the software implementation. Spyware is divided into three main areas of use – marketing, surveillance, and resource consumer-related activities. For the purposes of this discussion, the marketing area is defined as any business making use of demographic information whether it is anonymous

or directly identifiable to a user, and used strictly for the purpose of selling a legitimate product or service. The surveillance category is defined as having as its main objective the tracking of users or gathering user information in a far greater degree of detail than the marketing category. Its main use is in law enforcement, industry asset and employee monitoring, or intelligence gathering activities. Resource consumers are defined as those who financially benefit from utilizing system resources in compromised systems.

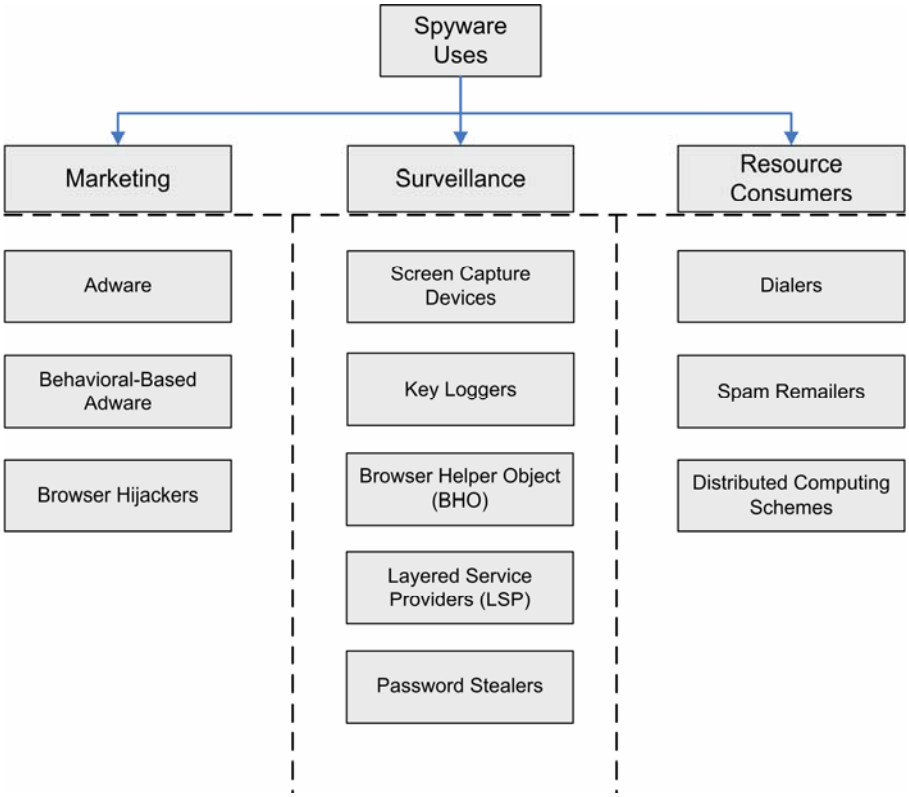


Figure 4 Spyware Uses and Mechanisms

Based on these definitions, adware and behavioral-based advertising (both of which may include pop-up advertising type behavior), and browser hijackers fall within the marketing category. They draw traffic to affiliated web sites and attempt to generate business transactions.

In the case of the surveillance category, screen capture devices, key loggers, password stealers and similar programs closely monitor user activities on a system.

Browser Helper Objects and Layered Service Providers are able to intercept web traffic before it is encrypted by Secure Sockets Layer (SSL), thus further expanding surveillance capabilities.

Resource consumers is the category of spyware designed to profit from distributed computing by taking system resources away from the user. For example, remailers utilize bandwidth-rich DSL-connected systems to distribute spam. Another yet more egregious instance utilizes unused storage or CPU cycles in a compromised system and sells them to clients with massive processing or storage requirements.

In summary, spyware is a computer program that is either (1) developed with the express purpose to steal resources or collect user or organization data, or (2) deployed with the intent to profit financially or strategically from data collected, and must have four common activities. These four activities are: hide, collect, communicate, and survive in a hostile environment. Of course, spyware of type 1 may exhibit one or more of these four activities as well.

The mechanics of spyware activities are as follows. Spyware hides by using deceptive or surreptitious techniques. Spyware collects system, organizational, or personal data. Spyware communicates collected data to a remote or third party. Spyware is designed with a degree of resilience, remaining present in a system as long as possible.

The extent of the information collected, and program activities or capabilities, are diminished or hidden from the user through deception or obfuscation.

Tracking of users may be accomplished without relying on the local execution of a specific program, as is the case with cookies and web bugs. Cookies and web bugs in and of themselves are unable to track users on the Internet. But when the same cookie or web bug is used by numerous affiliated web sites, or within numerous web sites in the same domain, patterns emerge allowing companies to closely watch user activities. Therefore, cookies and web bug technologies are in a sense, abused to provide a much greater degree of detailed information about a user than perhaps was initially intended. Additionally, cookies and web bugs are not considered spyware here because they lack the four basic activities present in spyware software.

D. SPYWARE CAPABILITIES

Spyware capabilities extend from the four basic activities observed among them. Figure 5shows three main capabilities observed in spyware.

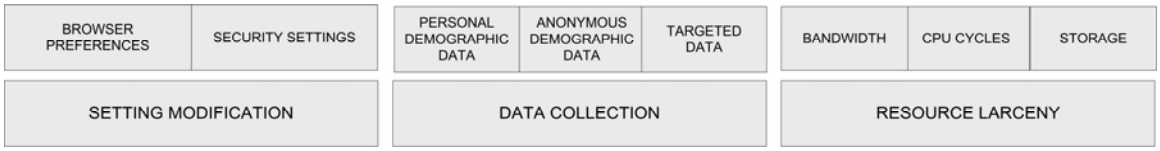


Figure 5 Spyware Behavior.

All spyware capabilities can be classified as being either passive (collection) or active (modification and larceny). Passive spyware consists of strictly monitoring software which communicates gathered data back to a third party. Such programs do not interfere with user activities and may go mostly undetected in a compromised system. Active spyware consists of programs with the logic or ability to detect specific events and act or react based on a specific criteria. Based on the degree of aggressiveness, such programs announce their presence in a compromised system by removing control of the system from the user, as in the instance of browser hijackers. For example, in the most benign instance, mistyping a URL at a hijacked web browser may redirect the application to an unintended spyware-affiliated web site. Alternatively, the visiting of specific web sites may trigger a pop-up or HTML injection of competing business advertisements. In the most egregious instances, a Browser Helper Object and key logger software was configured to monitor internet activity. When an unsuspecting user visited anyone of 50 banking institutions, the program would intercept POST and GET commands prior to their encryption and would, in turn, forward them to a Russian web server [1].

1. Setting Modification

A subset of spyware programs has the ability of changing system settings, including Internet browser application settings or system wide security settings. The programs may be called browser hijackers in their most benign form or malware in their most egregious instance. Browser hijackers will generally change the various settings associated with the default homepage, search page, and error page brought up by a browser. More menacing changes may include the lowering of security settings and

addition of malicious or spyware-affiliated web sites into the browser or software firewall Trusted Zone (e.g., Microsoft Internet Explorer and ZoneAlarm).

Spyware may also undermine the security posture of a system be either exposing the system to a new set of vulnerabilities present in the spyware program itself or by actively sabotaging and disabling the operation of defensive measures such as software firewalls, anti-virus, and anti-spyware programs. Such actions may include the temporary or permanent disabling of security-associated services and processes shutting down applications, the modification of the host file preventing anti-virus and autoupdate applications from obtaining the latest signature files from company servers, and the altering of the transport service provider by writing a winsock2 layered service provider (LSP) over the TCP/IP protocol stack [22]. LSPs implement higher-level custom communication functions, allowing malicious programs to monitor all communications associated with the browser application. As an LSP, spyware is able to intercept communications before they are encrypted and sent via Secure Socket Layer (SSL), thus exposing sensitive or confidential information.

2. Data Collection

Spyware collects three main kinds of information – anonymous demographic data, personally identifiable demographic data, and targeted data.

Data is forwarded to data repositories which can lead to simple market demographic analysis associated with advertisement serving, or may be associated with identification theft, corporate espionage, or classified government materials being disclosed. Spyware data collection can also lead to the use of system resources which can further serve as a launch pad for distributed Denial-of-Service (DOS) attacks or the creation of BotNets – armies of compromised systems remotely controlled, and acting in unison, by a malicious user.

a. Anonymous Demographic Data

Anonymous demographic data is comprised of client information such as files downloaded, browser type, screen resolution, colors displayed, plug-ins available, search queries, time spent at a particular web site, referrer web site, and the next web site

to be visited. It is generally limited to web surfing habits, but may also include user demographics when available. Such demographic information may include sex, age, zip code, marital status, dependents, and more.

b. Personal Demographic Data

Personal demographic data is comprised of names, telephone numbers, addresses, IP addresses, system serial numbers, behavior patterns or any other information leading to the direct identification of the user.

c. Targeted Data

Targeted data is a special category for data collection in that this area is considered to be strictly associated with malicious or intelligence gathering activities. At the user level, targeted data may include the active searching or compromising of passwords, social security numbers, credit card numbers, and financial or medical information. At the corporate or national level, targeted information includes specific corporate trade secrets, contract and project information or industrial process specifications, and information of a specific classification or topic relevant to national security.

3. Resource Larceny

Resource larceny pertains to those spyware programs designed to profit or benefit from using the computer resources of compromised systems. For the purposes of this discussion, the author does not consider resource larceny those spyware programs which consume excessive amounts of computer resources as a result of their monitoring or advertising serving activities but instead instances where the spyware perpetrator directly profits from the use of compromised system resources. Altnet is an excellent example of such a program [48]. Altnet was distributed as a companion program with Kazaa Media Desktop. The program was created by Brilliant Digital Entertainment (BDE), a company specifically selling storage and processing power based on a distributed storage and computing business model. A good illustration of what BDE had in mind can be found in the Kazaa and BDE end user license agreement (EULA):

> You hereby grant BDE the right to access and use the unused computing power and storage space on your computer/s and/or internet access or bandwidth for the aggregation of content and use in distributed computing. The user acknowledges and authorizes this use without the right of compensation. Notwithstanding the above, in the event usage of your computer is initiated by a party other than you, BDE will grant you the ability to deny access [48].

This is considered an example of resource larceny spyware because, although BDE states users will be given the ability to deny access to system resources, the statement is buried in a combined EULA. Furthermore, display of EULA statements is often made extremely difficult as documented by Benjamin Edelman [12]. In this case, Kazaa's EULA is 5,936 words long and displayed over fifty-six pages. Additionally, bold face, line breaks, heading styles, or other distinguishing techniques are purposefully omitted to blend text in confusing and obfuscating ways preventing a clear understanding of what the user is agreeing to.

Such applications are considered storage sinks if their primary role is to consume storage space, and they are considered CPU cycle sinks if their main objective is to consume processing power. Spam remailers are considered to mainly target bandwith and are thus considered bandwidth sinks, with CPU processing consumption as a secondary objective.

E. INFECTION VECTORS

Spyware infection comprises four main vectors – deception, bundled software, exploits, and inadequate security settings. Following is a detailed description of each of these vectors:

1. Deception

Deception is used extensively by spyware in attempting to gain control of a computer system. Social engineering and psychology play a role in enticing unsuspecting users to download or activate a program thus triggering the infection of the system. This is generally accomplished via pop-up windows or advertisements, some made to look like operating system error windows, requesting the user to accept the installation of "required" software to view or enter a web site. Others still offer purported

anti-virus or anti-spyware scanners, some “freely distributed” to remove phony detection alerts. Even if a user declines an offer or closes a window, installation takes place nevertheless.

Another practice when addressing bundled software is the hiding of overly permissive EULAs in the EULAs of other applications in the bundle. It is common to see EULAs extending for thousands of words, spread over tens of pages, viewable only via a small application window, effectively hiding their intentions [12].

2. Bundled Software

As indicated above, spyware relies heavily on freeware and shareware for distribution. Programs such as peer-to-peer (P2P) file sharing Kazaa or the WeatherBug – spyware which provides weather reports – are instances in which spyware programs are bundled with desirable applications which do not explicitly inform the user of the exchange of information to be taking place.

In this respect, these bundles are Trojan horses :in the absence of true, clear, and upfront disclosure of the functionality of the software bundle, many of these programs purport to do one thing but in fact, do much more than that, opening communication channels to a third party which in-turn, may install additional programs or take control of system resources.

3. Exploits

Spyware makes extensive use of known and zero-day exploits to gain access to a system. As shown in detail in the experiment discussed in chapter IV and V, spyware is able to exploit vulnerabilities present in Internet browsers to gain access to computer system. Figure 6 shows the anatomy of a spyware drive-by-download attack. Beginning with a user conducting web surfing activities, if a malicious web site is encountered, a triggering event is raised. This triggering event may consist of a script exploit used to download and install a mini-installer program. Small in size, the mini-installer can go mostly undetected while controlling the speed of spyware installation, thus preventing over consumption of resources and prematurely revealing its presence to the user. Once the mini-installer is executed, it may gain control of the system or it may contact other servers from where it may download the remaining spyware components or additional

affiliate programs. From this point forward, and with complete control of the system, spyware will monitor its health and may take actions to prevent removal from the system. Once it is firmly ingrained in the system, monitoring of user activities or collection of data begins. Infection by spyware may also lead to remote access by malicious users.

4. Inadequate Security Settings

The drive-by-download attack can be accomplished through configuration vulnerabilities as well as vulnerability "exploits." For example, Internet browsers configured to arbitrarily grant access to all scripting and trust all sources are likely to be fertile ground for spyware infection.

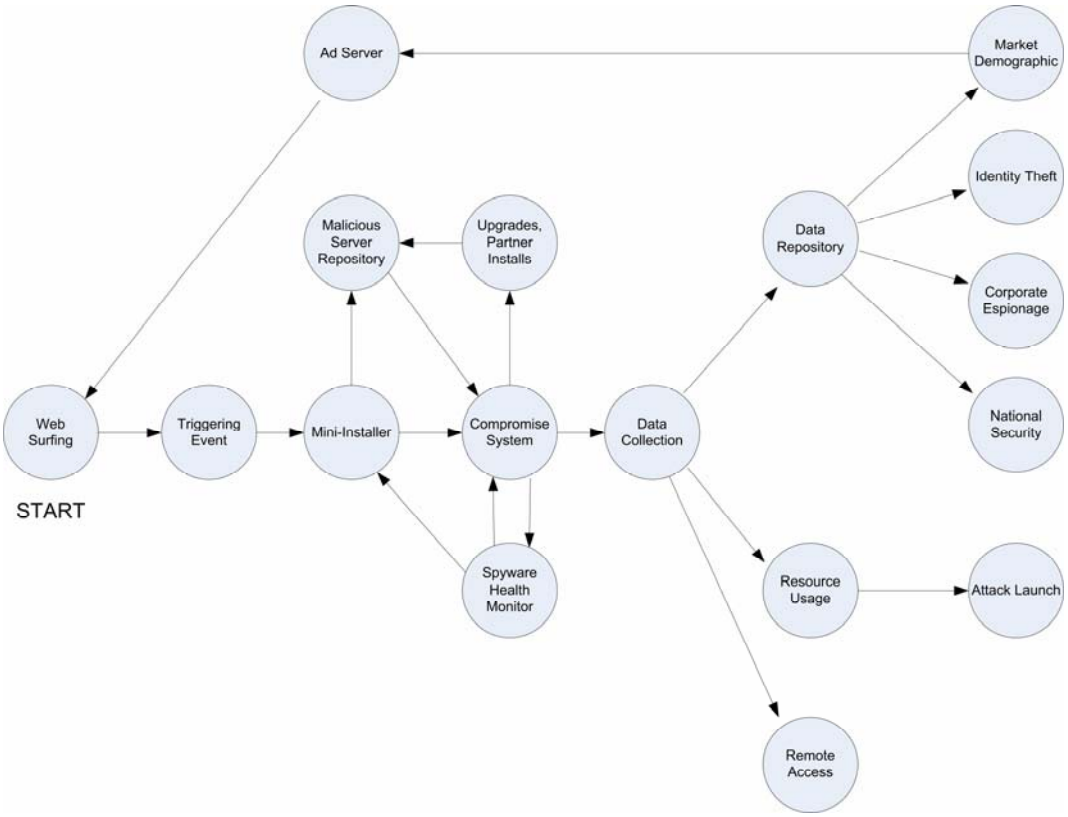


Figure 6 Anatomy of A Drive-By-Download Spyware Attack.

F. CONCLUSIONS

This chapter discussed some of the definitions given to spyware and the difficulties in arriving to a consensus working definition. Four basic spyware activities are discussed in this chapter: hide, collect, communicate, and survive. This chapter also proposed a concise spyware definition based on three main area considerations – the development intent, profitability intent, and the presence of the four basic spyware activities.

A discussion of spyware activities was provided from the perspective of passive versus active. Additionally, spyware activities are categorized as performing setting modifications, data collection, or resource larceny. The chapter concludes with a discussion of infection vectors and offers a description of a exploit drive-by-download infection.

The following chapter describes the test bed configuration for an experiment in which various sectors of the Internet were evaluated for infection of spyware via drive-by-download techniques.

IV. EXPERIMENT TOOL DEVELOPMENT

The following chapter describes an experiment designed to assess the potential for spyware infection through drive-by-download techniques in patched and unpatched Windows XP systems through the use of Internet Explorer and Firefox browsers. A description of the experiment methodology, test bed, choice of tools used in the experiment, scripts generated to conduct the experiment, selected Internet sectors, and browser configurations are discussed in this chapter.

A. METHODOLOGY

This experiment was designed to determine whether the practice of relatively "safe" web surfing to mainstream, popular web sites can lead to infection of computer systems by spyware. Specifically, the empirical analysis investigates the extent to which drive-by-download techniques and browser vulnerabilities result in the surreptitious installation of software in the system.

Although there are various web seal or trustmark programs attempting to provide a level of legitimacy and trust in web sites (i.e., TRUSTe, BBBOnLine, and others), such programs have not played a vital role in user activities or behavior when surfing on the Internet [65]. Therefore, for the purposes of this experiment, web sites displaying such web seals were not specifically targeted to satisfy the definition of "safe." Instead, the term "safe" is used from the perspective of an internet user conducting surfing activities while not specifically seeking high risk web sites from such sectors of the Internet as hacker or warez, adult entertainment, or online gambling-related web sites. Furthermore, our definition of safe web sites includes those sites a user would encounter simply by visiting the top or most popular hits in different topic areas as provided by a search engine (e.g., Google™). Additionally, the definition of safe web surfing activities does not include interaction with the individual homepages by clicking on banner ads or popup advertisements, except when necessary to dismiss the popup window by closing the window, or responding with a "Cancel" or "No" response. Therefore, "safe" web sites consist of relative mainstream popular internet destinations that are outside high risk sectors of the Internet.

One exception to the manner in which web sites were selected pertains to web sites developed for children. The American Library Association (ALA) compiles a list of web sites under their "Great Web Sites Seal of Approval" program [2]. These web sites are reviewed by librarians and deemed to be suitable web sites for children. The ALA defines children as those persons under the age of 14. The seal of approval criteria is based on authorship and sponsorship, purpose, design and stability, and content. Privacy and security are not considerations when granting this seal of approval. The presence of ALA seal of approval was used to develop the list of child-related web sites for inclusion in this experiment.

For comparison purposes, high risk sectors of the Internet were also included in this experiment.

The experiment can be broken down into the following tasks:

- Accumulate a list of web sites for each of eight safe and three unsafe sectors of the Internet.
- Visit each web site with four different web browser applications – two different browsers under a default unpatched Windows XP installation, and two browsers under a fully patched Windows XP installation.
- Collect system snapshot data.
- Data Analysis
    - Detection of infection for the Internet sector.
    - Identification of malicious web site responsible for infection

Execution of the experiment is accomplished by a collection of Visual Basic Scripts (VBScripts). These scripts drive browser applications to visit the various Internet sector Uniform Resource Locators (URLs) and idle for five seconds to allow spyware infections to take place, and proceed to collect system snapshots prior to visiting the next URL. These system snapshots are later compared against baseline conditions. The scripts are described in greater detail in Chapter IV Section D of this report. Internet browser applications were used to visit the various web sites and to permit a level of vulnerability to malicious web site attacks that a general internet user would encounter. By using Microsoft's Internet Explorer and Mozilla's Firefox browsers, spyware infection is possible as a result of the vulnerabilities present in each of the browsers.

Detection of infection on the workstation that hosts the browser is accomplished via three different techniques. The first is through the use of a host integrity monitoring system. This consists of software used in a client-server configuration which generates a baseline snapshot of a remote test bed file system and collects subsequent comparative snapshots over a defined period. This provides information on any files or folders added, modified, or deleted, as well as information on changes made to user accounts, running services, and open communication ports. The second technique consists of client-based commonly recommended anti-spyware scanning tools. These tools are used to determine spyware infection at the completion of each Internet sector experiment. The third technique consists of a collection of client-based third party tools used to collect system information after visiting each individual web site. Information collected includes a list of running processes and services, open communication ports and files associated with such ports, a list of applications or programs scheduled to autostart upon boot-up or login, browser security and preference settings, a snapshot of the hosts file, a list of browser favorites or bookmarks, and over eighty-seven different system registry sub keys. This information is compared to baseline snapshots in an effort to identify specific changes caused by a particular web site.

Based on the design of the experiment, conclusions may be made on the relative risk factor of various Internet sectors, the use of Microsoft's Internet Explorer browser versus Mozilla's Firefox browser, and the state of a default configuration unpatched Windows XP system versus a default configuration fully patched Windows XP system.

B. TEST BED DESCRIPTION

The test bed is comprised of a workstation, a file server, a hub, and a router. The workstation is configured with an operating system which hosts the integrity monitoring system and VMWare. The VMWare environment is configured with five client operating systems. The host operating system is a fully patched Windows XP system. This system is comprised of an Intel Pentium 4 3.2Ghz, 2GB of Ram, and two 120GB hard drives. The test bed environment is protected by a Cisco 2600 router implementing Network Address Translation (NAT). A Windows XP file server located on a separate computer is

used for the storage of data. This same server is also used for the collection of network traffic during the web surfing simulation phase of the experiment.

1. Network Topology

For the purposes of this experiment, an unrestricted T1 Internet connection was established. The Cisco 2600 router was used to protect the test bed from infection by means other than strictly spyware-related drive-by-download attacks. The router was further connected to a four-port 3Com hub to which the file server and the Host were connected. The VMWaret system hosted the five simultaneous VMWare images, each consisting of a separate Windows XP virtual machine. The virtual machines consisted of a passive experimental control identified as hostname PASSIVE, a default unpatched Windows XP and Internet Explorer installation identified as IE, a default unpatched Windows XP and Firefox installation identified as FF, a fully patched Windows XP and Internet Explorer installation identified as IESEC, and a fully patched Windows XP and Firefox installation identified as FFSEC. Figure 7 depicts the network topology as well as the IP addresses and "hostnames" assigned to each component of the experiment.

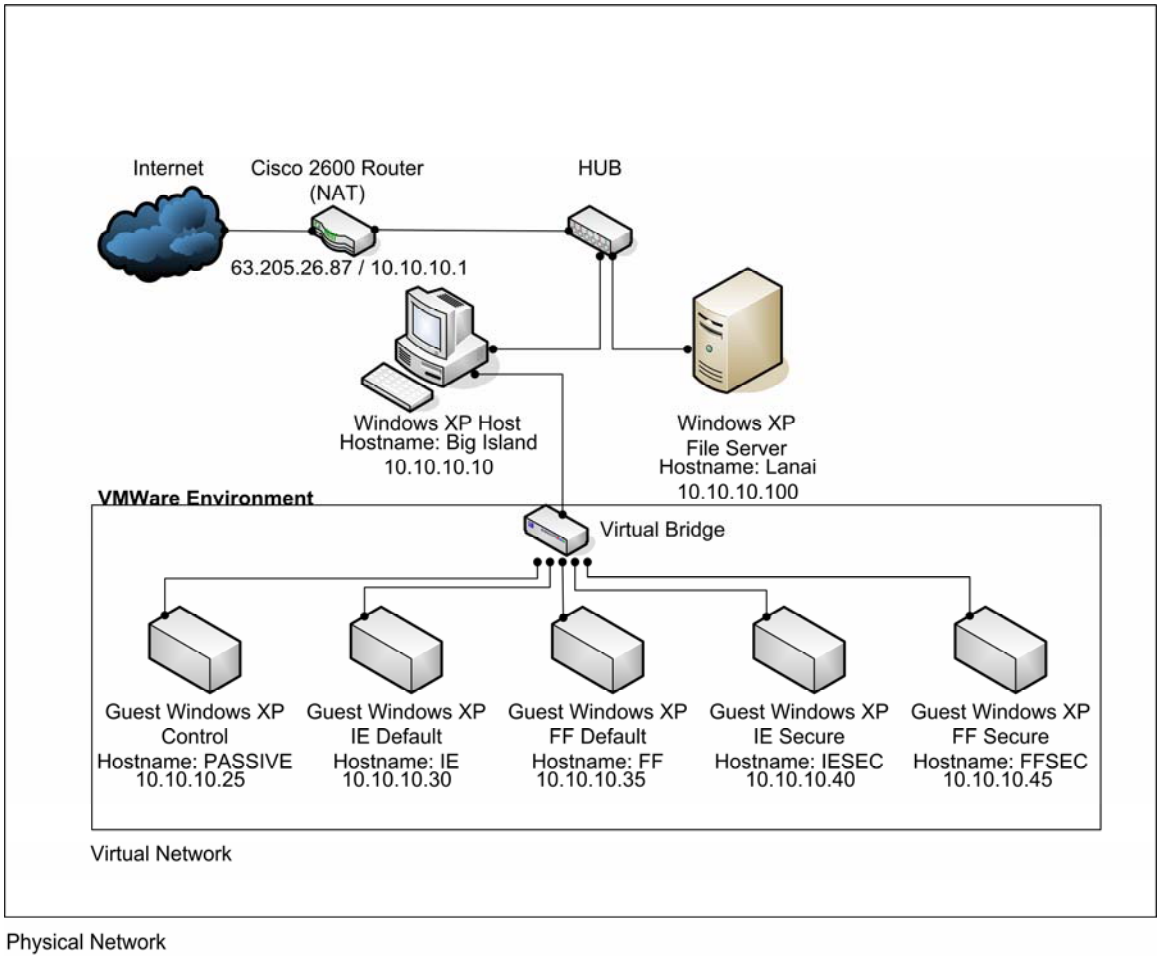


Figure 7 Test Bed Network Topology.

a. Router Configuration

The router was configured to implement NAT and only to accept web traffic on port 80, Secured Sockets Layer (SSL) traffic associated with encrypted HyperText Transfer Protocol (HTTP) on port 443, and Domain Name System (DNS) on UDP port 53. Figure 8 provides a list of the access-list configuration as reported by the router when executing the "show run" command.

```
access-list 101 permit tcp any any eq www
access-list 101 permit udp any any eq domain
access-list 101 permit tcp any any eq 443
```

Figure 8 Cisco 2600 Router Access-list Configuration

2. VMWare Environment

For the purposes of this experiment, a total of five virtual machines was created. All five machines consisted of Microsoft Windows XP installations logged in under the "Administrator" account and with the following configurations:

a. IE Guest Operating System

The IE Operating System was of a default installation of Microsoft Windows XP and Internet Explorer. Patches or fixes were not installed either for the operating system or for the Internet browser. No security enhancements were made or configuration settings modified. While it is recognized that this configuration is obviously exposed to attacks, it remains, nevertheless, the default out-of-the-box configuration of the operating system and thus, it is representative of systems encountered on the Internet and used by less-than-savvy computer users around the world.

b. IESEC Guest Operating System

The IESEC Operating System also consisted of a default installation of Microsoft Windows XP and Internet Explorer. However, unlike the IE configuration, this platform was fully patched with all Microsoft-available patches as of July 2005. A complete list of the installed patches is included in Appendix D. The configuration included Service Pack 2 and subsequent patches installed via the AutoUpdate feature included in Windows XP. Additionally, the software firewall included with Windows XP was allowed to remain enabled.

c. FF Guest Operating System

The FF Operating System is similar to the IE configuration with the exception that the default browser is set for Mozilla's Firefox 1.0.4.

d. FFSEC Guest Operating System

The FFSEC Operating System is similar to the IESEC configuration with the exception that the default browser is set for Mozilla's Firefox 1.0.4. No additional security updates were installed for the browser as there were none available at the time of the experiment.

e. PASSIVE Guest Operating System

The PASSIVE configuration consisted of an idle IE configuration. The virtual machine was simply allowed to remain active in the virtual network, but did not actively navigate to any web site. The intent of this virtual machine was to document any possible infection of the test bed by means other than the web surfing simulation.

For each of the above systems, a deviation from a simplistic default installation of Windows XP was, however, the inclusion of common third party applications associated with the enhancement of the web surfing experience. Macromedia's Shockwave plug-in software has reported 390 million downloads, and Macromedia's Flash Player is present in over 98% of internet-enabled desktops [32]. These applications are commonly found in Windows XP systems. To this purpose, all Windows XP Operating System installations were configured with the following applications:

- Sun JAVA Runtime Environment (Java 5.0 Update 4 jre-1_5_0_04-windows-i586-p-iftw.exe).
- Macromedia Shockwave Player 10.1.0.11.
- Macromedia Flash Player 7.0.
- Google™ Toolbar: This software was installed expressly for the sole purpose of limiting the number of unwelcome popup advertisements encountered in some sectors of the Internet. Although it is conceivable drive-by-download attacks may be launched from popup windows, this experiment intends to document infection as a result of visiting homepages. The Google™ Toolbar was only installed in the IE platform due to the fact that pre-Service Pack 2 Internet Explorer installations do not include a popup blocker feature. This popup blocker software prevents new windows from displaying advertisings which can interfere with the ability to view a website of interest. It should be noted, however, that the software was not completely successful in preventing all popups as advertisers utilize several different techniques. Additionally, once infection has taken place via a drive-by-download or vulnerability, and a third party application has been installed, the popup blocker software is unable to prevent further advertisements. All other features other than the popup blocking feature of the Google™ Toolbar were disabled for the purposes of this experiment.

3. File Server

The file server consisted of a Windows XP system located outside the VMWare virtual network environment. The computer was fully patched and configured to share a folder on the network for storage of data. The folder was further configured to allow only the creation of files but prevent the deletion of files or folders. Storage of data was placed under separate folders for each of the test bed platforms. Thus, the IE platform stored its data in the C:\Repository\IE folder and so on.

The system also contained an installation of the open source network traffic analyzer Ethereal and its accompanying command-line utility Tethereal. Tethereal was used during the course of the experiment to collect network traffic. All traffic was captured and stored in libpcap format files for possible subsequent analysis. Capture filters were used to limit the collection of network traffic only to those five virtual machines located in the virtual network and using either port 80 or port 443.

C. CHOICE OF TOOLS

During the course of the experiment, various previously developed tools were utilized to gather system state information to determine if infection of the guest operating system virtual machines had taken place, and if so, which web site was responsible for the infection. The tools used in this experiment were selected based on their free availability and popularity within the IT community, and the fact that most of them have command-line support facilitating scripting.

1. Anti-spyware Scanners

Four different anti-spyware scanners were used in this experiment, each of which is freely available: Microsoft's AntiSpyware (Beta 1), Lavasoft's Ad-Aware, Spybot Search and Destroy, and Earthlink's SpyAudit. Each of the tools generates a text or html formatted report. First, these scanners were used to document clean, uninfected test bed baseline conditions. These reports were stored and compared against subsequent scanning reports. Second, following the conclusion of a test run, i.e., successful execution of all 500 URLs associated with a given Internet sector, the scanners were used a second time to detect spyware infection as a result of web surfing activities within a specific sector. The use of these scanners only documented the presence of spyware but

did not assist in identifying the web site responsible for the infection. VBScripts and third party command-line tools were used to collect system snapshot information and analyze it to identify malicious web site URLs. Sample outputs generated by some of these third party command-line tools are provided later in this section.

a. Microsoft AntiSpyware

For this experiment, Microsoft AntiSpyware version 1.0.614 and definition file 5729 dated June 27, 2005, was used. The software conducts an extensive scan of registry keys, memory running processes, and the file system.

b. Lavasoft Ad-Aware

Ad-aware SE is an adware and spyware scanning program. According to Lavasoft's product web site, "Ad-aware conducts a comprehensive scan of memory, registry, hard drive, removable and optical drives for known data mining, aggressive advertising, parasites, scumware, selected traditional Trojans, dialers, malware, browser hijackers, and tracking components." For this experiment, Ad-Aware SE version 1.06R1 and definition file SE1R51 21.06.2005, was used.

c. Spybot Search and Destroy

Spybot Search and Destroy version 1.4 and definition file 2005.06.24 were used for this experiment. The product web site states that Spybot Search and Destroy identifies adware, browser helper objects (BHO), browser hijackers, dialers, key loggers, malware, spyware, Trojan horses, and worms.

d. Earthlink Online Spyware Scanner

Earthlink offers a scanning tool at their web site [11]. For this experiment, SpyAudit version 4.0.0.2 with digital signatures dated June 3, 2005, was used. This tool provides a limited fast scan of registry keys, key directories and browser hijackers, and reports on adware, Trojan horses, system monitors and adware related cookies found in the system.

2. SysInternals Utilities

SysInternals LLC maintains a suite of advanced utilities for system administrators and security researchers [40]. A limited version of many of these utilities is available free of charge. For this experiment, several were used.

a. Autoruns v8.0

This utility collects information on programs configured to execute during the system boot-up and login process, as well as in such areas in the registry as the Run and RunOnce sub keys [41]. Additionally, it enumerates any programs listed in the system startup folder, an area in Windows systems where programs may be launched during the startup process. This tool was invoked after each URL and used to complement information gathered by reg.exe. A review of the various startup parameters analyzed by this tool assisted in the development of a more complete tool created in VBScript. Scripts used as part of this experiment are discussed later in this section. Figure 9 provides sample output generated by Autoruns.exe. In this instance, the tool identified osirisd.exe and vmwareservice.exe as services scheduled to run in this computer. The first reported service is associated with the host integrity monitoring system while the second service is associated with the VMWare image environment. Subsequent registry keys are listed but not reported to contain any entries.

```
C:\tools>autorunsc -a -w –s -e -d -m –p
Autoruns v8.00 - Autostart program viewer
Copyright (C) 2002-2005 Mark Russinovich and Bryce Cogswell
Sysinternals - www.sysinternals.com

HKLM\System\CurrentControlSet\Services
   osirisd
       c:\windows\system32\osirisd.exe
   VMTools
       Provides support for synchronizing objects between the host and guest
operating systems.
       VMware, Inc.
       c:\program files\vmware\vmware tools\vmwareservice.exe

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify
HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Catalog9
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components
HKCU\SOFTWARE\Microsoft\Active Setup\Installed Components
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved
```

Figure 9 Sample Output for Autorunsc.Exe Tool.

b. PSService v2.13

PSservice generates a list of services currently running in a Windows system [43]. The list includes such information as the status of the service, configuration, and dependencies. This tool was used after visiting each URL and the report is compared against baseline conditions. Figure 10 shows sample output generated by this tool. In this instance, the tool reported two separate services, providing the name and description associated with each of them as well as the current status, along with other information.

```
C:\tools>psservice

PsService v2.12 - local and remote services viewer/controller
Copyright (C) 2001-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

SERVICE_NAME: Alerter
DISPLAY_NAME: Alerter
Notifies selected users and computers of administrative alerts. If the
service is stopped, programs that use administrative alerts will not
receive them. If this service is disabled, any services that explicitly
depend on it will fail to start.
      TYPE          : 20 WIN32_SHARE_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
      WAIT_HINT     : 0x0

SERVICE_NAME: ALG
DISPLAY_NAME: Application Layer Gateway Service
Provides support for 3rd party protocol plug-ins for Internet
Connection Sharing and the Internet Connection Firewall
      TYPE          : 10 WIN32_OWN_PROCESS
      STATE         : 1  STOPPED
                         (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE     : 1077 (0x435)
      SERVICE_EXIT_CODE : 0  (0x0)
      CHECKPOINT    : 0x0
```

Figure 10 Sample Output for Psservice Tool.

c. TCPView v2.4

TCPView generates a list of all TCP and UDP endpoints and state of TCP connections in a system [44]. Additionally, this utility can also report the name of a process that owns a particular endpoint. TCPvcon, a command-line version of TCPView, was used in conjunction with Netstat after each visited URL and current conditions were

compared against the baseline. Figure 11 shows sample output generated by tcpvcon. In this instance, several TCP and UDP ports are listed in various states, and the process and process identifier (PID) associated with each connection is also reported.

```
C:\tools>tcpvcon –a

TCPView v2.34 - TCP/UDP endpoint lister
Copyright (C) 1998-2003 Mark Russinovich
Sysinternals - www.sysinternals.com

[TCP] System
      PID:     4
      State:   LISTENING
      Local:   FF:microsoft-ds
      Remote:  FF:0
[TCP] C:\WINDOWS\System32\osirisd.exe
      PID:     1536
      State:   LISTENING
      Local:   FF:2265
      Remote:  FF:0
[TCP] System
      PID:     4
      State:   ESTABLISHED
      Local:   ff:1041
      Remote:  lanai:netbios-ssn
[UDP] C:\WINDOWS\system32\svchost.exe
      PID:     876
      Local:   FF:epmap
      Remote:  *:*
[UDP] System
      PID:     4
      Local:   FF:microsoft
```

Figure 11 Sample Output for Tcpvcon Tool.

d. Handle v3.1

Handle provides a list of open handles maintained by processes in a Windows system, a list of files, directories, or objects associated with running programs [42]. This tool was used after each visited URL and the results were compared against the baseline to detect any suspicious process behavior. Figure 12 shows sample output generated by Handle. In this instance, it shows three separate processes, smss.exe, csrss.exe, and handle.exe, respectively. Each process is reported as having various files or sections associated with it. Additionally, the tool also reports the PID and account under which the process is running.

```
C:\tools\handle –u

Handle v2.2
Copyright (C) 1997-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

------------------------------------------------------------------------
smss.exe pid: 552 NT AUTHORITY\SYSTEM
    8: File          C:\WINDOWS
   1c: File          C:\WINDOWS\system32
------------------------------------------------------------------------
csrss.exe pid: 616 NT AUTHORITY\SYSTEM
    c: File          C:\WINDOWS\system32
   40: Section       \NLS\NlsSectionUnicode
   44: Section       \NLS\NlsSectionLocale
   48: Section       \NLS\NlsSectionCType
   4c: Section       \NLS\NlsSectionSortkey
   50: Section       \NLS\NlsSectionSortTbls
  3b0: File          C:\WINDOWS\system32\ega.cpi
------------------------------------------------------------------------
handle.exe pid: 708 FF\Administrator
    c: File          C:\Tools
```

Figure 12 Sample Output for Handle.Exe Tool.

e. Bginfo v4.07

Bginfo is used to display system information on the desktop background for the purpose of easy identification of the test bed currently being worked on [9]. Information displayed by Bginfo included hostname, IP address, subnet address, among others.

3. Microsoft Utilities

Microsoft includes a variety of system administration tools with each of its operating systems. This experiment utilized the following tools:

a. Reg v3.0

This tool allows command-line interaction with the system registry files by providing support for querying, adding, deleting, copying, saving, restoring, loading, unloading, comparing, exporting, and importing registry keys [33]. Reg.exe was used for backing up registry keys prior to the commencement of the experiment and for restoring them to baseline conditions following each visited URL. It was also utilized for the querying of registry values following each visited URL, creating a text file containing a

report of registry keys and values. This text file was stored at the remote file server and subsequently compared against baseline conditions.

b. TaskList.exe

This tool provides a list of active processes and services running in a computer [34]. Reports generated after each visited URL were compared against baseline conditions that existed prior to the commencement of the experiment. Figure 13 shows a list of processes, associated PIDs, and services.

```
C:\tools>tasklist /SVC /FO TABLE

Image Name        PID   Services
=========================================================
System Idle Process     0     N/A
System                  4     N/A
smss.exe          552   N/A
csrss.exe         616   N/A
winlogon.exe      640   N/A
services.exe      692   Eventlog, PlugPlay
lsass.exe         704   PolicyAgent, ProtectedStorage, SamSs
svchost.exe       876   RpcSs
svchost.exe       968   AudioSrv, Browser, CryptSvc, Dhcp, dmserver,
                        ERSvc, EventSystem, helpsvc, lanmanserver,
                        lanmanworkstation, Messenger, Netman, Nla,
                        Schedule, seclogon, SENS, ShellHWDetection,
                        srservice, TermService, Themes, TrkWks,
                        uploadmgr, W32Time, winmgmt, WmdmPmSp,
                        wuauserv, WZCSVC
spoolsv.exe       1268  Spooler
osirisd.exe       1536  osirisd
VMwareService.exe 1584  VMTools
explorer.exe      1880  N/A
cscript.exe       508   N/A
tasklist.exe      1112  N/A
```

Figure 13 Sample Output for Tasklist.Exe Tool.

c. Netstat.exe

Netstat.exe provides command-line access to TCP/IP statistics and active connections. Netstat was used after each visited URL for the purpose of determining whether any new connections or listening ports were established. These reports were compared against baseline conditions captured at the commencement of the experiment. Figure 14 shows sample output generated with netstat. In this instance, various local TCP and UDP ports, connection status, and PIDs associated with the connection are reported.

```
C:\tools>netstat –ano

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       876
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING       968
  TCP    0.0.0.0:2265           0.0.0.0:0              LISTENING       1536
  TCP    0.0.0.0:5000           0.0.0.0:0              LISTENING       1168
  TCP    10.10.10.35:139        0.0.0.0:0              LISTENING       4
  TCP    10.10.10.35:1041       0.0.0.0:0              LISTENING       4
  TCP    10.10.10.35:1041       10.10.10.100:139       ESTABLISHED     4
  UDP    0.0.0.0:135            *:*                                    876
  UDP    0.0.0.0:445            *:*                                    4
  UDP    0.0.0.0:500            *:*                                    704
  UDP    0.0.0.0:1027           *:*                                    1136
  UDP    0.0.0.0:1030           *:*                                    968
  UDP    10.10.10.35:123        *:*                                    968
```

Figure 14 Sample Output for Netstat.Exe Tool.

4. Osiris Host Integrity Monitoring

Osiris Host Integrity Monitoring system is a freely available host tool that can be used to monitor changes in a computer system or network of hosts by taking regular snapshots of the file system as well as of user lists, group lists, open ports, running services, and other parameters [71]. Subsequent scans of these systems are compared against a baseline. Osiris was used to establish baselines for each of the test beds prior to the commencement of the experiment and subsequent scans were performed following the completion of each Internet sector experiment.

5. Ethereal Network Protocol Analyzer

Ethereal is a freely available network protocol analyzer generally used in troubleshooting network traffic [15]. For the purposes of this experiment, Ethereal V. 0.10.11 was installed in the file server. Traffic was collected with Tethereal, a non-graphical utility provided with Ethereal, and stored in libpcap formatted files for possible further analysis.

Figure 15 depicts the location of both the Osiris Host Integrity Monitoring system and of Ethereal with respect to the overall network.

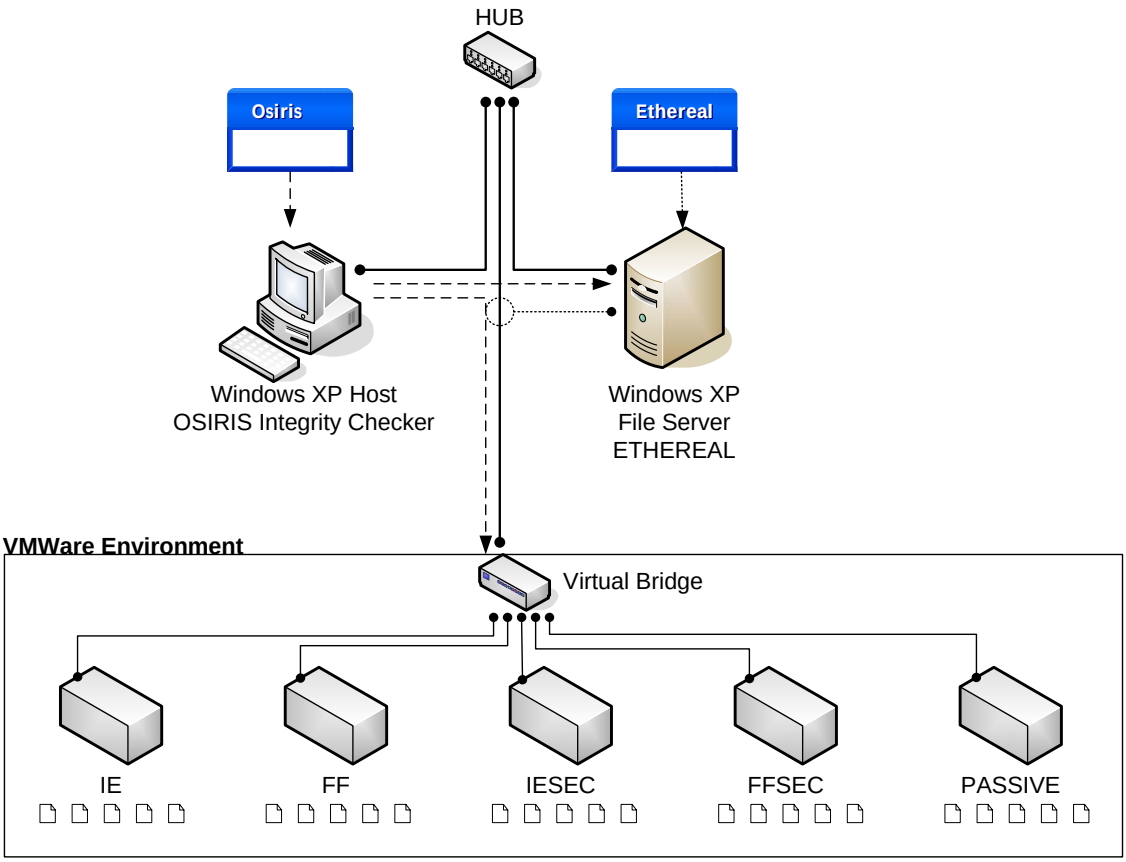


Figure 15 Osiris Host Integrity Monitoring System and Ethereal System Location within the Network.

D. SCRIPTS

The experiment was controlled by five main VBScript scripts tasked with the invoking of Internet browsers and collection of data. Additional VBScripts were used to support converting, parsing, and formatting text files generated during the course of the experiment. VBScript is a subset of Microsoft's Visual Basic and it is interpreted by the Windows Scripting Host (WSH) which is included with Windows XP. Thus, one of the benefits taken into consideration in the selection of a scripting tool set was the complexity of software installation in each of the test beds. Other scripting languages such as Perl or Python would require a separate interpreter installation in each of the test beds, so VBScript was chosen. Additionally, VBScript provides built-in support for

Component Object Model (COM) Automation which was used by the scripts to access Internet Explorer objects. One last factor contributing to the selection of VBScript over another scripting language was its readability. Although wordy, VBScripts are easy to read and can be understood by novice and experienced programmers alike.

The first and second scripts were browser "drivers," one for Internet Explorer and one for Firefox. The scripts, intuitively named ie.vbs and firefox.vbs, create an instance of the browser and opens a text file from which URLs are fetched, passing them to the browser navigation function. Five seconds of idle time is allowed to the browser prior to the collection of system snapshots. The selection of a window of time prior to collection of the system snapshot was based on empirical analysis of malicious web sites. Refer to Chapter 4 Section E for a detailed discussion on this topic. Various system snapshot parameters were then collected over the next thirty to sixty seconds (time intervals varied based on system CPU utilization).

The driver scripts make an attempt to dismiss browser windows requiring attention by either sending Alt-N keystrokes to handle "No" responses, {ESC} to dismiss the window, or Alt-C to reply with a "Cancel" response. This addressed web sites that requested to install software or make the visiting web site the default homepage for the browser. Therefore, when offered an option, the scripts would decline to install software or change settings, thus attempting to take a safe web surfing approach.

The driver scripts accepted three parameters consisting of the platform being tested (i.e., IE, IESEC, FF, or FFSEC), the Internet sector being tested (i.e., Government, Banks, Online Travel, etc), and a third optional parameter used to restart the script from a particular URL. This third parameter was added to the script to handle instances in which the browser crashed, possibly as a result of malicious web site activities. For those instances when the browser crashed, the scripts invoked themselves passing the URL at which it crashed as its third parameter, thus allowing the test run to continue. The driver scripts invoke three other scripts: registry.vbs, collector.vbs, and miner.vbs, in turn. A URL access log record of the URL and corresponding access time was sent both to the screen and to a text file at the remote file server.

Script registry.vbs collected registry keys pertinent to modifications in browser settings or registry keys associated with the starting of files, services or processes upon boot-up, login, or other triggering events. Eighty-seven different registry keys were queried using reg.exe. Values obtained from these queries were stored in a text file at the file server, which would later be compared against baseline snapshots. Additionally, the script enumerates files in the browser cookies folder, Internet Explorer Favorites folder, and Administrator and All Users startup folders. It was, therefore, possible to detect start-up and browser configurations changes by a specific web site by comparing information collected by this script against baseline conditions. Prior to navigating to the next URL in the list, the script would reset the registry and folder content to baseline conditions. A list of the eighty-seven registry keys is provided in Appendix B.

Script collector.vbs collected host file information for Internet Explorer platforms, and host file, cookies, bookmarks, and preferences for Firefox platforms. The script read the files and performed a difference comparison against baseline files. If a difference was noted, the file was copied to the file server and the original baseline file replaced the changed file, resetting the system to its initial conditions.

The miner.vbs script executed six different third party tools consisting of autorunsc.exe, handles.exe, tasklist.exe, tcpvcon.exe, netstat.exe, and psservice.exe. Each of these tools generated output which was redirected to separate text files in the remote file server for later comparison against baseline conditions. These reports provided comprehensive information on programs scheduled to start upon boot-up, login, or other triggering events, files and folders associated with running processes, a list of running processes and services, and open TCP and UDP communication ports.

Figure 16 depicts the order in which the browser automation and data collection took place.

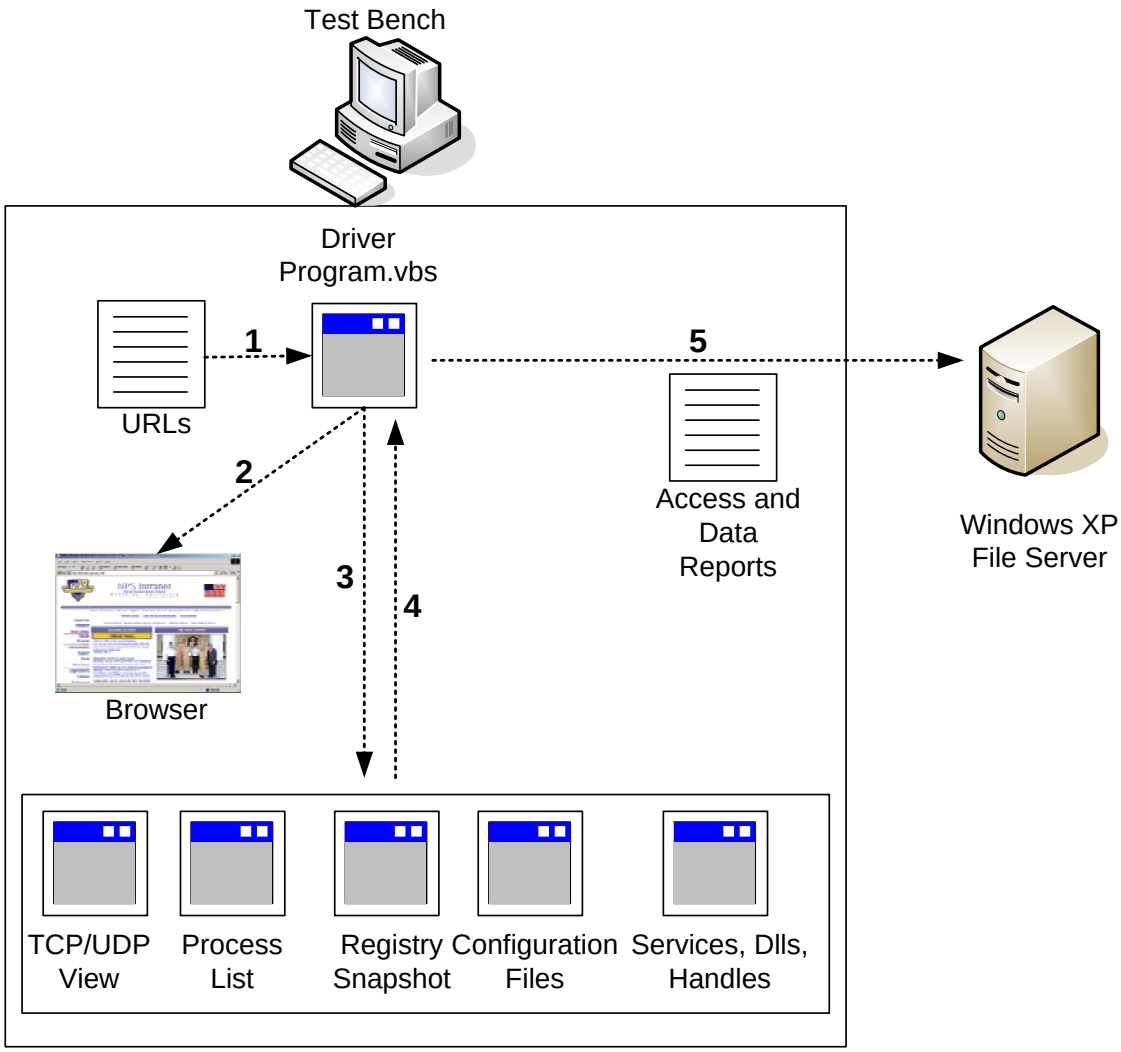


Figure 16 Analysis Process

Additional scripts were prepared for handling various administrative tasks associated with the formatting and preparation of links for input into the experiment, and formatting and processing of data files. Such scripts are considered peripheral to the experiment itself and administrative in nature so no further discussion is included in this work.

E. INFECTION VALIDATION

In designing the experiment, preliminary tests were performed to determine if infection of the test bed using the proposed methodology was in fact possible. For this purpose, the IE test bed was sent to five known malicious web sites, since it was the least secure and most basic of the two Windows XP platforms. Figure 17 shows a sample

screen capture of the download times associated for web site "www.unix-time-format.dzwonki.pruszkow.pl." This web site URL was obtained in the same manner as URLs obtained for the various sectors of the experiment (Described in Chapter IV Section F). Table 1 shows a comparison of download times for three of the five different malicious web sites. Two of these web sites accomplished the installation of spyware by crashing the web browser, thus halting the script and the timer. Therefore, no download timing data is available for these two web sites. Additionally, Table 1 also includes the time it took to collect the system state snapshots. Increased CPU activity associated with the installation and execution of spyware is reflected in these time intervals – reported as "total collection time." Observations from this limited evaluation of malicious web sites lead to the configuration of the ie.vbs script with a 15-second timeout and a 5-second idle parameters, respectively. If the web site took longer than 15 seconds to download, the browser would be directed to the next URL in the list of URLs compiled for such Internet sector. The 5-second idle parameter causes the browser to wait a minimum of 5 seconds after the web site has been downloaded prior to starting the collection of system state snapshots. Based on this limited evaluation of malicious web sites, a 5-second idle parameter was considered adequate to allow infection of the test bed. In fact, the crashing of the browser in two of the five malicious web sites evaluated occurred within ten seconds of arrival to the specific web site, well within the 5-second idle and 15-second timeout parameters.

```
8/6/2005
INFECTION_TEST # 1:Initiating Experiment at 8/6/2005 2:00:31 PM
-------------------------------------------------------------
Count Access Time       Download Status   URL Visited
1     8/6/2005 2:00:46 PM           1     www.unix-time-
format.dzwonki.pruszkow.pl

End Time: 8/6/2005 2:01:09 PM
Total Run Time (h:m:s): 0:1:38
Started Download at t1 seconds after 12:00am:         50431.52
Ended Downloading at t2 seconds after 12:00am:        50446.67
Total Time to download web page (seconds):            15.15625
Idle time starts at t1 seconds after 12:00am:         50446.67
Idle time ended at t2 seconds after 12:00am:          50451.67
Total Idle time (seconds):                            5

Started Collection of system state 20.15625 seconds after arriving to
the page
Started Collection at t1 seconds after 12:00am:       50451.67
Ended Collection at t2 seconds after 12:00am:         50468.73
Total Collection time (seconds):                      17.0625
```

Figure 17 Malicious Web Site Download Time Evaluation.

Table 1 Preliminary Malicious Web Site Download Comparisons

URL	Download (seconds)	Collection (seconds)	Notes
www.unix-time-format.dzwonki.pruszkow.pl	15.1	17.1	Drive-By-Download
Viking-supply-net.to.opole.pl	N/A	N/A	Browser crashed
Food-pyramid.ok.opole.pl	N/A	N/A	Browser crashed
Sex-archive.biz/movies/	9.5	38.1	Drive-By-Download
m.cpa4.org/reality	17.6	21.6	Drive-By-Download

To document infection by each of these malicious web sites, the test bed was examined with each of the four previously described anti-spyware tools. Table 2 provides the number of individual spyware artifices identified by each of the tools. Numbers in parenthesis represent the number of separate signatures associated with the spyware artifice.

Table 2 Preliminary Malicious Web Site Infection Comparisons

Malicious URLs	Earthlink SpyAudit	Ad-Aware	Spybot S&D	Microsoft AntiSpyware
www.unix-time-format.dzwonki.pruszkow.pl	5	9 (99)	11 (41)	10 (259)
Viking-supply-net.to.opole.pl	3	7 (63)	6 (20)	7 (130)
Food-pyramid.ok.opole.pl	3	7 (65)	6 (20)	7 (130)
Sex-archive.biz/movies/	3	9 (100)	11 (41)	10 (242)
m.cpa4.org/reality	1	3 (26)	2 (2)	1 (4)

F. URL DETERMINATION

The experiment is intended to assess generally "safe" sectors of the Internet. To that end, links associated with banking, insurance, children, real estate, online travel, universities, government, and military-related web sites were evaluated. Additionally, high risk areas of the Internet were also evaluated for comparison purposes. These high risk areas are particularly prolific on the Internet and enjoy a great deal of traffic and interest. These sectors included online gambling, hacker and warez, and adult entertainment-related web sites. Where feasible or clearly identifiable, the links were pruned to the homepages of each domain by the use of regular expression scripts, explained in further detail later in this section.

A list of banking institution-related web sites was obtained from the Federal Deposit Insurance Corporation (FDIC) which maintains a list of member banks. This source provided good-quality banking-related web sites in the sense that all links where

truly associated with banks. A total of 7308 links were obtained but due to time constraints, 500 web sites were selected at random.

A list of child-related web sites was obtained in part from the American Library Association and their Great Web Sites Seal of Approval Program. Additional links were obtained at other minor child-related directories. A total of 500 links where compiled in this group.

University-related links were compiled from the University of Texas at Austin which maintains an alphabetical list of all U.S. community colleges and universities. The links to the individual institution web sites are provided in this listing. A total of 2135 links were obtained, from which 500 were selected at random.

Government and military-related web sites were compiled using the Google™ search engine. Searches were conducted by filtering for the .gov or .mil domains. A total of 500 links were collected for the government sector of the Internet, and 350 distinct web sites were collected for the military sector of the Internet. Here, a government-related web site is defined to be web sites hosted by a federal or state government agency. A military-related web site is defined to be a web site hosted by a military-related agency or branch, in the .mil domain.

Insurance-related web sites are web sites that discuss or provide insurance services, brokers, training classes, or insurance associations. Real estate-related web sites are web sites that discuss or provide real estate related services as well as Internet-related real estate, as is the case with web hosting services. Online gambling-related web sites are web sites that provide or discuss gambling activities. Online travel-related web sites are web sites that sell travel packages, make reservations, provide information on travel destinations, or discuss or provide services to the travel industry, as in the case of training schools or certifications and associations.

Web sites for the remaining sectors of the Internet used in this experiment were compiled by conducting search engine queries for specific key words. For example, when compiling the online gambling sector of the Internet, the search query consisted of keywords “online gambling” or “online casino.” When compiling real estate-related web

sites, search queries included keywords such as "real estate realtor mortgage homepages." In this manner, 500 links were collected for each of the insurance, online travel, and real estate-related sectors of the Internet. Only 418 web sites were compiled for the adult entertainment sector, 346 web sites were compiled for hacker and warez-related sectors, and 392 web sites were compiled for the online gambling sector, respectively. Table 3 provides a list of queries used in compiling links for the various Internet sectors as well as the number of web sites tested under each sector.

Table 3 Link Source Or Search Query String

Sector	**Sector Size (urls)**	**Search query or source**
Safe Sectors		
Banks	501*	Source: www.fdic.gov
Children	500	Source: www.ala.org
Government	500	Homepage site:.gov
Insurance	500	Source: www.ultimateinsurancelinks.com
Military	350	Homepage site:.mil
Online Travel	500	online travel reservations "travel agent"
Real Estate	500	Real estate realtor mortgage homepages Mortgage company homepages
Universities	500	Source: www.utexas.edu/world/univ/state/
Unsafe Sectors		
Adult Entertainment	418	Free porn XXX
Hacker / Warez	346	Cracks exploits virus Trojans download –infosec –zdnet –cnet –site:.mil,_.gov
Online Gambling	392	online gambling, online casino

* An additional web site in excess of the 500 intended for this sector was mistakenly included in the test sequence. All 501 URLs are bank-related web sites obtained from FDIC.

Due to the nature of search engine algorithms and heuristics, queries may yield different results over time. Thus, the above referenced queries may not yield the exact same links used in this experiment.

Two scripts were used to filter out the links from html pages, remove duplicates, and format the links in a manner appropriate for the experiment. The links were sorted alphabetically and Google™ related links were removed. Using regular expressions, a script was written to trim URLs to the domain level. In VBScript, the matching expression was crafted as follows:

Re.Pattern = "\S+(\.\w+)+\.[a-z]{2,3}"

This expression begins by matching one or more non-white space characters followed by one or more combinations of a dot and one or more word characters, followed by a dot and two or three letters. The intent for the application of this filter was to remove duplicate URLs and broaden the number of different domains visited during the course of the experiment. It was not the intent of this experiment to visit multiple web sites hosted under the same domain but target the homepages for each of the domains.

Unlike the government, military, and university domains which are .gov, .mil, and .edu, respectively, the remaining sectors of the Internet are located under a variety of domains such as .org, .net. .com, etc. There is, however, a problem with the quality of the links obtained through this method. There were instances in which unrelated links to the Internet sector were presented by the search engine. This introduced a degree of "noise" into the accuracy of the links. While it was unfeasible to confirm that each link was, in fact, related to the target sector, efforts were made to filter out obviously unrelated links. Evaluations of the number of false links present in each of these Internet sectors is discussed in Chapter VI.

It is worth noting that since the intent was to simulate activities by a casual Internet user, it is reasonable to expect unrelated web sites to be encountered during a specific topic query. Such unrelated web sites may lead to infection by spyware.

G. BROWSER CONFIGURATIONS

Two different internet browsers were selected for this experiment. Microsoft Internet Explorer and Mozilla's Firefox represent both the most popular and prevalent browser in use and the fastest rising browser in growth of use, respectively.

1. Microsoft Internet Explorer

Internet Explorer version 6.0.2600.0000.xpclient.010817-1148 was used in the IE platform while version 6.0.2900.2980.xpsp_sp2_gdr.050301-1519 was used in IESEC, respectively. The browser was generally left in its default configuration with the exception of changes from a default of "prompt" to a setting of "enabled" or "allowed" for a limited number of categories. Table 4 provides a list of deviations from default configurations under the Internet Zone [10] and the Advanced Settings configuration tabs:

Table 4 Microsoft Internet Explorer Deviations From Default Configuration

Categories	Default Setting	Test Bed Setting
Internet Zone Tab		
Download signed ActiveX controls	Prompt	Enable
Font download	Enabled	Disabled
Display Mixed Content	Prompt	Enabled
Installation of Desktop Items	Prompt	Enabled
Launching Programs and Files in an IFRAME	Prompt	Enabled
Submit Nonencrypted Form Data	Prompt	Enabled
Advanced Settings Tab		
Automatically check for Internet Explorer Updates	Enabled	Disabled
Enable Install on Demand (Internet Explorer)	Enabled	Disabled
Notify when Download Complete	Enabled	Disabled
Warn if Changing Between Secure and Unsecure Mode	Enabled	Disabled

All cookies were blocked, as tracking cookies was not the focus of this experiment. For a complete list of all browser settings, please refer to Appendix E

2. Mozilla Firefox

Mozilla Firefox version 1.0.4 was used in this experiment. The only modification made to the default installation of Firefox was the destination of downloaded files. A copy of the preferences file pref.js is included in Appendix E.

H. SUMMARY

This experiment was designed to detect infection of the test bed as a result of drive-by-download techniques when visiting web sites. By collecting system state snapshots via a combination of VBScripts and third party tools, it is possible to discern the infecting web site and the spyware installed in the test bed. Additionally, by utilizing Internet Explorer and Firefox browsers in unpatched and patched default configurations – as an out-of-the-box installation of Windows XP and, after applying Microsoft Service Pack 2 and subsequent patches – it is possible to assess the efficacy of these patches.

In the following chapter, a discussion of the web surfing simulation utilizing the setup described in this chapter is provided, including a description of the effects observed in the test bed as a result of spyware infection.

THIS PAGE INTENTIONALLY LEFT BLANK

V. INTERNET WEB SURFING SIMULATION

This chapter describes the execution of the experiment to determine if infection by spyware through drive-by-download techniques is present in various main-stream sectors of the Internet. It starts with the collection of URLs, and progresses to web surfing simulation, collection of data, and subsequent detection process. This chapter also discusses some characteristic spyware symptoms observed in the infected test beds.

A. DESCRIPTION OF EVENTS

Collection of Internet sector URLs was conducted between the third and fourth week of June 2005. VMWare test bed images were also created during this same period. The test bed images were equipped with the appropriate scripts and Internet sector URL files in a "Tools" directory, which was created off the C Drive. Additionally, third party tools were placed in this same directory and anti-spyware scanning software was installed and configured to "scan mode" only. The file server "repository" folder was mapped in each of the test beds to drive letter "S."

Following the configuration of the test beds, each of the four anti-spyware scanning tools was used, starting with the least intrusive – Earthlink's SpyAudit. An html report documenting the "clean" state of the system was saved on the test bed desktop as well as in a baseline folder located at the remote file server. LavaSoft's Ad-Aware, Spybot Search and Destroy, and Microsoft Anti-Spyware were used in this order, with the most comprehensive scanning settings selected. Scan text reports for each of these tools were stored at the baseline folder of the remote file server. All five test beds were found not to contain spyware.

Once each of the test beds was configured and scanned for spyware, Osiris was used to establish a baseline report of the file system, user accounts, running services, and open ports. These reports were stored in the Host system (Refer to Figure 7). Additionally, miner.vbs, registry.vbs, and collection.vbs scripts were executed, collecting a baseline snapshot of system conditions prior to the commencement of the experiment. Files generated by these scripts were also saved in the baseline folder for later comparison.

URLs associated with the Internet sector to be tested were copied from a text file in a sub-directory to c:\tools\urls.txt. It is this file which is used by the web browser driving scripts to fetch test URLs.

The browser driving script – either ie.vbs or firefox.vbs depending on the test bed – was executed, allowing the web surfing simulation to commence. Upon execution, a web browser window was instantiated and navigation to the web sites began, logging access times to a command-line window and to a text file at the remote file server. A total of eight report files are generated per visited URL. The files were named in the following manner:

`[url number][Internet Sector][Tool].txt`

The first parameter consists of the number of the URL being accessed, based on its row number within the c:\tools\urls.txt file. This number can be directly correlated to the full URL for each web site. The second parameter consists of the Internet sector being tested (i.e., banks, universities, online travel, etc). The third parameter consists of the tool generating the report or the type of data being generated – AutoRuns.exe, Handles.exe, Netstate.exe, ProcessList, Reg.exe, Service.exe, TCP, and Negative Diff Report. This last file parameter provided information on whether the host file had changed with respect to the baseline copy of the same file. The host file is located at c:\windows\system32\drivers\etc\host. For test beds configured with the Firefox browser, additional information on whether the preferences file "pref.js," and the bookmarks file "bookmarks.html" had changed when compared to baselines.

A total of 4,000 text files is generated for a 500-URL Internet sector in a given test bed. Therefore, given that four test beds are used for actively surfing the Internet, 16,000 files were created per Internet sector. The PASSIVE test bed virtual machine does not actively surf the internet as it is used as a passive control. Therefore, this test bed did not generate any text file reports. Web surfing simulations and data collection took place between July 6 and July 24, 2005. When possible, all four test bed platforms

and the passive control test bed were executed simultaneously. However, due to differences in browser download times, CPU utilization, internet bandwidth, and server performance, the four test beds would fall out of synchronization with one another. This behavior leads to differences in web site access times between just a few seconds to a few hours. Additionally, the time required to test all 500 URLs in a given sector ranged from approximately seven to nine hours, given no catastrophic anomalies such as virtual machine crashes.

Upon conclusion of each Internet sector experiment, the test bed was scanned with Osiris to determine any changes in the file system, user accounts, processes and services, and communication ports. Reports indicating differences noted between this scan and baseline conditions were generated and stored in the Host machine. Osiris reports for test beds exhibiting spyware infection are included in Appendix C. A limited number of these reports showed administrative changes made to the test bed image between the establishment of the baseline and the completion of the test bed experiment. Administrative changes included the replacement or upgrading of scripts as improvements were implemented. Additionally, as URL text files were gathered and added to the test bed images, Osiris reports reflected their inclusion or modification. For these reasons, such changes are not the result of spyware activity and have therefore been edited out from the reports included in the Appendices in the interest of brevity and relevance.

Following the completion of each sector's experiment and the test bed Osiris scan, each of the anti-spyware scanning tools was invoked in the same order as described previously in this chapter. Reports generated by these tools were stored in the remote file server for later analysis. The virtual machine images were then powered down through normal Windows XP shutdown procedures and the files were copied to backup storage device.

1. Experiment Execution Anomalies

During the course of the experiment, three main anomalies were experienced:

a. Browser Crashes

Based on preliminary tests conducted during the development of the browser driving scripts, it was noted the browser application would crash in certain instances. As described in Chapter IV, the driving scripts were created with the ability to re-invoke themselves with a third optional parameter containing the URL number being accessed at the time of the crash. This would allow the script to restart, with a new instantiation of the browser application, at the next URL in the list of URLs for that sector.

b. VMWare Test Bed Image Crashes

In a few instances, the VMWare image itself crashed, shutting off a given test bed, and thus the operating system, browser application, and scripts executing within it. Although conclusive reasons for the cause of these crashes was not determined, it is strongly suspected power saving settings in the host machine under which all five virtual machines were located, may have contributed to the virtual machine instability. Once these settings were changed, the frequency of these crashes decreased dramatically.

However, the crashing of the test bed image did not affect the data collected by the experiment since this data was stored at the remote server. Analysis of the test bed image further revealed that improper shutoff of the image did not affect changes made on the image file as a result of spyware infection. The virtual machine test bed image files were configured to be persistent, meaning that changes made to the image file would be written immediately and permanently to the virtual hard disk. Additionally, upon experiencing the first crash of the test bed image, anti-spyware scanning tools were used to determine if any infection of the test bed had been lost as a result of the untimely and unexpected end of the test bed experiment. It was found that the test bed image had indeed been infected by spyware, a change from initial baseline conditions. The physical computer hosting the virtual machine was unaffected by the test bed crash.

c. Script Crashes

Browser scripts were also noted to crash, thus halting the experiment. This type of crash occurred when a particular web site presented the script with a modal window – a child window created by a parent application which had to be closed prior to granting control back to the parent application. If such modal windows could not be dismissed by a combination of {Alt-F4}, {ESC}, {Alt-N}, or {Alt-C} keys sent to the application interface, the experiment would halt. Encounters of such modal windows were rare and limited to web sites associated with the distrusted sectors of the Internet.

B. SIMULATION RESULTS

A detailed discussion of infection data and analysis is provided in Chapter VI. However, the following is a high level description of infection symptoms observed during the course of the experiment.

Depending on the Internet sector being tested, various degrees of disruption were observed during the course of the experiment. For Internet sectors such as banks, government, military, and universities, few additional windows other than the one instantiated by the test script, were observed. Additionally, browser crashes were rarely noted. On the other hand, while testing such Internet sectors as real estate and online travel, and certainly those sectors associated with high risk areas of the Internet such as hacker or warez and adult entertainment-related web sites, increased instances of unhandled popup advertisements and browser crashes were noted. Not coincidently, these same Internet sectors exhibited a high degree of spyware infection. One explanation for the presence of unhandled popup advertisements and browser crashes in these test beds is that once a test bed has been infected by spyware, popup blocking features offered by either Google™ Toolbar or Windows XP Service Pack 2 become ineffective. Furthermore, the type of popup advertisement associated with this undesirable software is generally aggressive in nature in the sense that they are designed to be highly persistent and have maximal exposure on a system in an attempt to gain full attention by users.

Specific spyware software was observed to be generally common across the various Internet sectors found to have been infected. Table 5 shows a list of all spyware

software identified by the anti-spyware scanning tools. The table indicates both the test bed platform and the Internet sector the software was found in. It was immediately obvious that Mozilla's Firefox browser was not infected by any of the Internet sectors tested.

Table 5 Spyware Found Across the Tested Internet Sectors by Test Bed

Spyware / Adware	Internet Sector										
	Banks	**Children**	**Gov.**	**Insurance**	**Military**	**Online Travel**	**Real Estate**	**Univ.**	**Adult Ent.**	**Hacker / Warez**	**Online Gambling**
180search Assistant						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Adintelligence.AproposToolbar										IE, IESEC	
Admilli Service						IE, IESEC	IE		IE, IESEC	IE, IESEC	
AlwaysUpdateNews Spyware										IE, IESEC	
AprosMedia										IE, IESEC	
AutoProxy Trojan						IE, IESEC	IE				
AvenueMedia.DyFuCA Browser Plug-in						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Bargain Buddy										IE, IESEC	
BookedSpace Browser Plug-in						IE, IESEC	IE				
ClearSearch Browser Modifier						IE, IESEC	IE				
ClimaxBucks						IE, IESEC	IE		IE, IESEC	IE, IESEC	
CoolWebSearch Browser Modifier						IE, IESEC	IE		IE, IESEC	IE, IESEC	
DealHelper										IE, IESEC	
EffectiveBandToolbar	IE			IE							
Exact.BullseyeNetwork										IE, IESEC	
Exact.CashBack Adware										IE, IESEC	
Exact.Downloader Trojan Downloader										IE, IESEC	
Exact.SearchBar Browser Plug-in										IE, IESEC	
Hijacker.TopConverting						IE, IESEC	IE		IE, IESEC	IE, IESEC	
IBIS Toolbar									IE, IESEC		
IEPlugin Spyware						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Internet Optimizer						IE, IESEC	IE		IE, IESEC	IE, IESEC	
ISearchTech.SideFind						IE, IESEC	IE		IE	IE, IESEC	
IST.ISTBar Browser										IESEC	

Spyware / Adware	Internet Sector										
	Banks	**Children**	**Gov.**	**Insurance**	**Military**	**Online Travel**	**Real Estate**	**Univ.**	**Adult Ent.**	**Hacker / Warez**	**Online Gambling**
Modifier											
IST.ISTbar.ActiveX Spyware										IE, IESEC	
IST.ISTbar.ContentMatch Control Browser Plug-in										IE, IESEC	
IST.SlotchBar Toolbar										IE, IESEC	
IST.XXXToolbar										IE, IESEC	
ISTBar/AUpdate										IE, IESEC	
KrAIMer Remote Access Trojan											IE
Laypros						IE, IESEC	IE		IE, IESEC	IE, IESEC	
MoneyTree Dialer						IE, IESEC	IE		IE, IESEC	IE, IESEC	
n-Case						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Pacimedia										IE, IESEC	
PeopleOnPage										IE, IESEC	
Possible Browser Hijack Attempt										IE, IESEC	
PowerScan										IE, IESEC	
Rotue						IESEC					
SahAgent						IE, IESEC	IE		IE, IESEC	IE, IESEC	
SearchExtender Spyware										IE	
SexList						IE, IESEC	IE		IE, IESEC	IE, IESEC	
ShopAtHome						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Spy # 32676						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Spy # 46b42										IE, IESEC	
Spy # 5a2ea										IE, IESEC	
Spy # 6d78b	IE			IE							
Spy # 87366										IE, IESEC	
Spy # 9594c										IESEC	
Spy # d566d										IE, IESEC	
Spy # d9c03										IE, IESEC	
Spy # da439										IE, IESEC	
Spy # dcce8						IE, IESEC	IE				

Spyware / Adware	Internet Sector										
	Banks	**Children**	**Gov.**	**Insurance**	**Military**	**Online Travel**	**Real Estate**	**Univ.**	**Adult Ent.**	**Hacker / Warez**	**Online Gambling**
Spy # fd1e0						IE, IESEC	IE		IE, IESEC	IE, IESEC	
TinyBar										IE, IESEC	
TopMoxie						IE, IESEC	IE				
TopRebates.WebRebates Adware						IE, IESEC	IE				
Transponder.ABetterInternet.Aurora Adware										IE, IESEC	
Trojan.Startup.NameShifter.I						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Unclassified.Spyware.57 Spyware										IE, IESEC	
Unclassified.Spyware.61										IE, IESEC	
Win32.Trojan.ByteVerify.A										IESEC	
Winad						IE, IESEC	IE		IE, IESEC	IE, IESEC	
WindUpdates Browser Plug-in						IE, IESEC	IE		IE, IESEC	IE, IESEC	
WindUpdates.MediaAccess						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Winsecure										IE, IESEC	
Xrenoder Browser Plug-in										IE, IESEC	
Zango Search Assistant Adware						IE, IESEC	IE		IE, IESEC	IE, IESEC	
ZyncosMark										IE, IESEC	
Other*						IE, IESEC	IE		IE, IESEC	IE, IESEC	
Quicktime video player **											IE, IESEC

* Unspecified spyware detection.

** Although QuickTime video player is not generally considered spyware or adware, in this instance the software was installed without any interaction or approval by the script.

It is interesting to note that the online gambling Internet sector was found to have minimal spyware infection of the test bed. As it pertains to this sector, infection of computers may be limited to the installation of shareware or freeware software, and less so associated with the use of drive-by-downloads.

Upon completion of the experiments in each of the infected test beds, immediate operating system instability was observed. The experiment was not concerned with documenting the degree of disruption to the test beds by the presence of spyware. Therefore, system metrics for CPU cycles, memory availability, and overall responsiveness, were not collected before or after infection. However, obvious system degradation in responsiveness was noted and certain Windows features such as Explorer – used to view folders and files – were inoperable. Additionally, spontaneous advertisings would popup while attempting to conduct the anti-spyware scans.

C. SUMMARY

This chapter described the sequence in which the various components of the experiment were carried out. Based on this process, test beds were infected by spyware when visiting online travel and real estate-related web sites as well as with the high risk hacker and warez and adult entertainment-related web sites.

A detailed discussion of the infection of the test beds follows in Chapter VI. This chapter will describe some of the findings as a result of the collection of system snapshots after each visited URL. It will further identify specific malicious web sites encountered during the course of the experiment.

VI. IN-DEPTH ANALYSIS

This chapter provides the in-depth analysis of the experiment. Each of the sectors found to have been infected is discussed in detail and an analysis of spyware infection for the sector is provided. Additionally, detection comparison among the various anti-spyware scanning tools is provided for each of the relevant sectors. The chapter also discusses the infection of both default and patched test beds and the possible encounter of zero-day exploits during the course of the experiment. Lastly, the chapter provides a list of some of the servers associated with the download of spyware-related binary files as well as some of the URLs identified as contributing to the infection of the test beds.

A. BANKING SECTOR

Analysis of the banking sector found infection of the IE test bed by at least one instance of spyware software.

Figure 18is a print out of the HTTP network traffic as captured by Ethereal. The figure shows the initial test bed request to download a file named object.cfm. It is indicated in this request that the referrer web site is http://stats4all.ws/fa/. This web site is registered with a Samoan domain and pretends to be stats4all.com, a web site which provides statistics and counters for web masters with information on the number of visits, type of browser used, and various other parameters associated with each visit. In fact, stats4all.ws appears to be just a front domain with broken links, associated with a Russian domain hosting company. This web site is receiving information about the client, including browser type, client screen resolution, number of colors shown, the existence of a java plug-in, and most interestingly, a “ref” parameter, presumably the referring web site, providing the URL of a web site from which stats4all.ws/fa was called. In this instance, the referring web site is listed as one of the banks the test bench visited. The reply from the server included a JScript script which invoked ActiveX objects to create a file in the local test bench. The local file is named EjAGgj1l.exe and is located in a special temporary folder established by the script. The file is not simply downloaded through a normal request but instead, it is crafted by the script within the test bench itself. The script creates the file, and writes ASCII characters ‘MZ’ before closing the file. This represents in hexadecimal the signature of a Windows Portable Executable (PE) file.

Once these first characters are written to the file, the script opens the file a second time and proceeds to write UNICODE representations of characters in the format "\u0090\u0003\u0000…." The previous example consists of the first three characters written to the file after the PE signature is written. The script concludes by executing the file via the WScript.Shell command. Please note the UNICODE portion of this script was truncated for presentation purposes. Bolded sections of this figure emphasize some of the items discussed above. The first bolded line shows the presence of object.cfm and the second and third line show the web statistics provider and the referring web site. Of interest are the fourth and fifth bolded lines in which a call to create and run an executable file is made.

```
GET /fa/p8YUroP3Fp6VFA/object.cfm HTTP/1.1
Accept: */*
Referer:
http://stats4all.ws/fa/?wmid=2seller&nav=MSIE&version=400&screensize=1152*864&
colors=32&sver=13&java=y&ref=http%3A//URL279/&mainref=http%3A%2F%2F
URL279%2F&navlan=&plug=&sUrl=http%3A//trustbid.ws/%3Fid%3D2seller&sExtra=None

Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: stats4all.ws
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/0.1.36
Date: Mon, 11 Jul 2005 00:39:35 GMT
Content-Type: application/hta
Connection: keep-alive
Keep-Alive: timeout=20
X-Powered-By: PHP/4.3.9
Content-Length: 18823

<script language=jscript>
try{
self.moveTo(5000,5000);
}catch(e){}
try{fs=new ActiveXObject("Scripting.FileSystemObject");
fname=fs.GetSpecialFolder(2)+'\\EjAGgj1l.exe';
a=fs.CreateTextFile(fname,true);
a.Write('MZ');
a.Close();
a=fs.OpenTextFile((fname),8,false,true);
a.Write("\u0090\u0003\u0000\u0004\u0000\uFFFF\u0000\u00B8\u0000\u0000\u0000\u0
040\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0000\u0
000\u0000\u0000\u0000\u0000\u0080\u0000\u1F0E\u000\u0000\u0000\u0000\u0000");
a.Close();
(new ActiveXObject("WScript.Shell")).Run(fname);
self.close();
}catch(e){}
</script>
```

Figure 18 Creation Of An Executable File In Test Bench

A subsequent attack on the test bench originated soon after the crafting of this file. The content of the UNICODE written into file EjAGgj1l.exe was not interpreted. However, it is possible that this first file is responsible for subsequent attacks observed in the network traffic captures. One in particular was associated with a GET method invoked by the IE test bench client to an IP address not associated with a bank URL. The GET command requested the download of a file named "X.exe" from URL "189.dapfeed.com." This author reassembled the executable file from the network traffic capture files and proceeded to execute the file within a virtual machine, conducting limited black-box binary analysis. A search for strings inside the executable file found the word "DIALER" as well as a 900-telephone number. It should be noted that Earthlink's SpyAudit and Spybot Search & Destroy identified one adware related program after the execution of this file. SpyAudit preliminarily identified the program as Spy#6d78b while Spybot Search and Destroy identified the program as EffectiveBandToolbar. The latter was based on a single registry entry and apparently failed to identified the many other changes and additions made to the registry by this program, as shown in Figure 19. This may be indicative that a complete installation of the dialer program did not actually take place in the test bed, explaining the single registry entry, or that the Spybot Search and Destroy did not contain a more comprehensive list of identifying signatures for this spyware program at the time of the scan. It appears this file is a dialer program designed to generate revenue for 1-900 type businesses by causing computers to dial out the number via modem communications. Additional analysis of registry queries conducted by this file revealed numerous queries to phonebook and Remote Access Server (RAS) related registry keys, further supporting the opinion that this file is associated with a malicious dialer program. Upon conclusion of program execution, X.exe deletes itself. As an ephemeral executable file, it is missed by the integrity checker software.

```
---------------------------------
Values added:4
---------------------------------
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-11d0-
96d8-00aa0051e51d}\{9B365890-165F-11D0-A195-0020AFD156E4}"
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-
11d0-96d8-00aa0051e51d}\{9B365890-165F-11D0-A195-0020AFD156E4}"
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Start_ShowNetConn_Should
Show: 0x00000042
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-
9888-006097DEACF9}\Count\HRZR_EHACNGU:P:\Qbphzragf naq
Frggvatf\Nqzvavfgengbe\Qrfxgbc\K.rkr: 05 00 00 00 06 00 00 00 40 14 4B BA 44 A9 C5 01

---------------------------------
Values modified:8
---------------------------------
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\RNG\Seed: E1 49 29 B3 2D 0F 3D 16 F9
5D 77 FD 67 06 14 32 B7 70 9E 45 77 94 68 96 A2 27 57 59 61 BD 52 C5 04 2D D5 DB 42 F2
37 06 03 B7 65 2E 47 F4 A2 BD 5C EF 43 B6 83 F5 54 DC 98 71 F3 A4 96 76 48 3C 33 6A A8
FE DF 82 8E 7B C3 31 3B 7A 98 DB 57 51
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 88 4A 6F 44 C8 17 B8 A6 B5
E3 7A BA 6E 9F 14 01 42 9D 4D 54 D0 0D 7C 5A 30 2B AB 2E DF EB 09 CA 51 98 E5 56 1A 79
7D E6 69 52 8B AF C4 F0 13 E1 D5 B7 71 36 B4 F7 0D EE 18 9C EC B8 24 F7 AB EA AA E1 5A
EB 01 78 6F 7A 81 AF E0 E4 EF 82 A0 B0
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\kmixer\Enum\Count: 0x00000000
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\kmixer\Enum\Count: 0x00000001
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\kmixer\Enum\NextInstance: 0x00000000
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\kmixer\Enum\NextInstance: 0x00000001
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kmixer\Enum\Count: 0x00000000
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kmixer\Enum\Count: 0x00000001
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kmixer\Enum\NextInstance:
0x00000000
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kmixer\Enum\NextInstance:
0x00000001
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-500\Software\Microsoft\Internet
Explorer\Main\conc: 0x42D1BADE
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-500\Software\Microsoft\Internet
Explorer\Main\conc: 0x430D6FFD
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-
9888-006097DEACF9}\Count\HRZR_EHACNGU: 05 00 00 00 8F 00 00 00 00 93 3D B1 44 A9 C5 01
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-
9888-006097DEACF9}\Count\HRZR_EHACNGU: 05 00 00 00 90 00 00 00 40 14 4B BA 44 A9 C5 01
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-
9888-006097DEACF9}\Count\HRZR_HVFPHG: 05 00 00 00 39 00 00 00 10 21 A1 AF 44 A9 C5 01
HKEY_USERS\S-1-5-21-220523388-790525478-839522115-
500\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-
9888-006097DEACF9}\Count\HRZR_HVFPHG: 05 00 00 00 3A 00 00 00 40 32 46 BA 44 A9 C5 01

---------------------------------
Total changes:12
---------------------------------
```

Figure 19 Registry Modifications Following Execution Of X.Exe

Subsequent research into the specific 900-telephone number included in the executable file found that the SANS Internet Storm Center confirmed the findings provided here [21].

A review of the URL access report generated by the experiment script ie.vbs revealed the browser suffered the first crash event of the experiment run when accessing URL number 279 out of the total of 501 URLs in the banking sector. Additional crash events were observed from this point forward in the experiment. URL 279 is associated with the bank web site referenced during the first attack.

Figure 20 shows a limited view of the TCP connections present while accessing URL 279. This information shows that while the browser was accessing the URL associated with bank 279, it was also accessing IP address 66.232.140.95. This IP address was also seen in the traffic capture files collected by Ethereal. This capture file shows the downloading of the JScript attack. Items in bold show established TCP connections with IP address 66.232.140.95, the IP address associated with the downloading of the Jscript script shown in Figure 18.

```
[TCP] [System Process]
      PID:     0
      State:   TIME_WAIT
      Local:   ie:3465
      Remote:  somebank.com:http
[TCP] [System Process]
      PID:     0
      State:   TIME_WAIT
      Local:   ie:3466
      Remote:  somebank.com:http
[TCP] [System Process]
      PID:     0
      State:   TIME_WAIT
      Local:   ie:3519
      Remote:  somebank.com:http
[TCP] [System Process]
      PID:     0
      State:   TIME_WAIT
      Local:   ie:3520
      Remote:  somebank.com:http
[TCP] C:\Program Files\Internet Explorer\iexplore.exe
      PID:     1332
      State:   ESTABLISHED
      Local:   ie:3541
      Remote:  66.232.140.95:http
[TCP] C:\Program Files\Internet Explorer\iexplore.exe
      PID:     1332
      State:   ESTABLISHED
      Local:   ie:3550
      Remote:  66.232.140.95:http
[TCP] [System Process]
      PID:     0
      State:   TIME_WAIT
      Local:   ie:3568
      Remote:  somebank.com:http
```

Figure 20 TCP State while accessing URL 279.

Based on this information, it appears that a visit to URL 279 may have initiated stats4all.ws and trustbid.ws to transmit malicious traffic to the test bed. Both of the files

described above were launched from stats4all.ws and trustbid.ws, possibly causing the infection of the test bed with a dialer program. However, although referrer fields in HTTP traffic during this Internet sector experiment show URL 279 referring traffic to the malicious stat4all.ws web site, it is possible to spoof such HTTP traffic, especially if the virtual machine was compromised by spyware prior to the start of the experiment. For this reason, a conclusive link between the specific bank web site and the malicious attack could not be established.

Additional Internet research on the nature of the attack observed during the experiment in the banking sector appears to point towards a cross-site-scripting type of attack. Furthermore, between March and July of this year, a considerable number of comments were posted on various Internet forums regarding the injection of a script in web servers. The script described in these comments is similar to the one described above, in which malicious files were downloaded from stats4all.ws and trustbid.ws.

Finally, a literature search indicates that financial institutions were targeted for similar attacks in 2004. According to [4, 8, 52, 69] attacks on IIS and Apache web servers may have placed millions of users at risk of infection with malware or spyware and attempts to steal personal and financial information.

All four anti-spyware scanning tools did not find any spyware infection in the IESEC, FF, FFSEC, or PASSIVE test beds.

B. INSURANCE SECTOR

Analysis of the insurance sector test bed images found apparent infection by spyware. According to SpyAudit, Spy#6d78b was identified as adware present in the IE test bed. Spybot Search and Destroy reported EffectiveBandToolbar present in the IE test bed. Spyware infection was not found by any of the anti-spyware tools in the IESEC, FF, FFSEC, or PASSIVE test beds.

A closer review of network traffic captured during the course of the experiment failed to identify suspicious network activity. It is suspected the reported spyware infection may be associated with baseline corruption as described in greater detail in Chapter VI Section H.

C. ONLINE TRAVEL SECTOR

The online travel sector experiment revealed infection by spyware of the IE and IESEC test beds. All four anti-spyware scanning tools reported spyware-related software in these test beds. Table 6 shows a brief break-down of the number of spyware reported by each scanning tools. The number reported on the left side of the slash symbol is the number of individual spyware items. Individual spyware items consist of a given name as assigned by the scanning tool, and associated executable files, dynamic link libraries, folders, or specific registry keys. The numbers reported to the right of the slash pertain to specific signatures found by the scanning tools, for example, the number of files, folder, or registry keys associated with spyware. Therefore, many signatures can be combined into one spyware item or reported under a given name.

Table 6 Spyware Infection in the Online Travel Sector

Test Bed	SpyAudit	Ad-aware	Spybot S&D	Microsoft AntiSpyware
IE	6	8 / 164	11 / 61	14 / 620
IESEC	6	8 / 168	12 / 61	14 / 614
FF	0	0	0	0
FFSEC	0	0	0	0
PASSIVE	0	0	0	0

A review of the integrity checker log revealed a total of 118 changes to the system since the baseline was established. Some of these changes pertained to upgrades to the testing scripts and text files used to feed the URLs to the Internet sector being tested. Figure 21 shows a list of the spyware-related changes to the file system in the IE test bed. Additionally, TCP port 2400 and UDP port 1145 are now active and associated with WebRebates0.exe and salm.exe, respectively. The Osiris-generated output shown in Figure 21 has been summarized for presentation purposes.

[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\cfout.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\djtopr1150.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\jkill.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\umqltg4cl_.exe]
[203][IE][new][C:\Program Files\180searchassistant\salm.exe]
[203][IE][new][C:\Program Files\180searchassistant\salmhook.dll]
[203][IE][new][C:\Program Files\Internet Optimizer\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\install.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\optimize.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\install.exe]
[203][IE][new][C:\Program Files\Media Gateway\Info.txt]
[203][IE][new][C:\Program Files\Media Gateway\MediaGateway.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\40754594.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\5b6fayfg.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\66708108.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\8wfvt0n5.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\Uninstall.EXE]
[203][IE][new][C:\Program Files\ProSiteFinder\fsn9m6kq.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ldhznpoc.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinder.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinderh.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\saqegvlp.DLL]
[203][IE][new][C:\Program Files\Web_Rebates\README.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\log.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rbh.htm]

```
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates0.exe]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates1.exe]
[203][IE][new][C:\Program Files\Web_Rebates\disp1150.exe]
[203][IE][new][c:\windows\Downloaded Program Files\ClientAX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\MediaGatewayX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\clientax.inf]
[203][IE][new][c:\windows\ci05p91g.exe]
[203][IE][new][c:\windows\gnwdsrut.exe]
[203][IE][new][c:\windows\nem220.dll]
[203][IE][new][c:\windows\system32\6v4mf229.dll]
[203][IE][new][c:\windows\system32\atl71.dll]
[203][IE][new][c:\windows\system32\drivers\PROCEXP.SYS]
[203][IE][new][c:\windows\system32\fqnkc7um.exe]
[203][IE][new][c:\windows\system32\qfmrgpis.exe]
[203][IE][new][c:\windows\wsem303.dll]
[221][IE][new][mod_ports][TCP:0.0.0.0:2400][TCP:2400;exe=C:\Program
Files\Web_Rebates\WebRebates0.exe;pid=5556;local=0.0.0.0;remote=0.0.0.0]
[221][IE][new][mod_ports][UDP:1145][UDP:1145;exe=C:\Program
Files\180searchassistant\salm.exe;pid=1976]
```

Figure 21 Spyware-Related Changes To The IE Test Bed File System As Reported By Osiris.

Figure 22 shows a similar report generated by Osiris for the IESEC test bed. Although it clearly shows fewer changes to the file system, it also shows considerable continued success by spyware in gaining access to the test bed. Salm.exe has successfully opened UDP port 1146 as it was the case in the IE test bed and the Background Intelligent Transfer Service appears to have been stopped. According to Microsoft's description for this service, if this service is stopped, Windows is unable to conduct automatic updates. This appears to be an attempt to maintain the compromised system in a vulnerable and accessible state.

```
[203][IESEC][new][C:\Documents and Settings\Administrator\Local Settings\Temp\cfout.txt]
[203][IESEC][new][C:\Documents and Settings\Administrator\Local Settings\Temp\djtopr1150.exe]
[203][IESEC][new][C:\Documents and Settings\Administrator\Local Settings\Temp\jkill.exe]
[203][IESEC][new][C:\Documents and Settings\Administrator\Local Settings\Temp\umqltg4cl_.exe]
[203][IESEC][new][C:\Program Files\180searchassistant\salm.exe]
[203][IESEC][new][C:\Program Files\180searchassistant\salmhook.dll]
[203][IESEC][new][C:\Program Files\Fqhgckb\Mlbv.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\actalert.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\install.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\optimize.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\update\actalert.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\update\install.exe]
[203][IESEC][new][C:\Program Files\Internet Optimizer\update\rogue.exe]
[203][IESEC][new][C:\Program Files\Media Gateway\Info.txt]
```

```
[203][IESEC][new][C:\Program Files\Media Gateway\MediaGateway.exe]
[203][IESEC][new][C:\Program Files\Web_Rebates\README.txt]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_r.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rb.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rbh.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_u.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ub.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ubh.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\pref1150a.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\scri1150a.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_r.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rb.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rbh.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_u.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ub.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ubh.htm]
[203][IESEC][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\log.txt]
[203][IESEC][new][C:\Program Files\Web_Rebates\WebRebates0.exe]
[203][IESEC][new][C:\Program Files\Web_Rebates\WebRebates1.exe]
[203][IESEC][new][C:\Program Files\Web_Rebates\disp1150.exe]
[203][IESEC][new][c:\windows\Downloaded Program Files\ClientAX.dll]
[203][IESEC][new][c:\windows\Downloaded Program Files\MediaGatewayX.dll]
[203][IESEC][new][c:\windows\Downloaded Program Files\clientax.inf]
[203][IESEC][new][c:\windows\aducljtv.exe]
[203][IESEC][new][c:\windows\buz.exe]
[203][IESEC][new][c:\windows\nem220.dll]
[203][IESEC][new][c:\windows\system32\4i8i2on4.exe]
[203][IESEC][new][c:\windows\system32\drivers\PROCEXP.SYS]
[203][IESEC][new][c:\windows\system32\e14l5p5t.dll]
[203][IESEC][new][c:\windows\system32\pb34j0sh.exe]
[203][IESEC][new][c:\windows\wsem303.dll]
[223][IESEC][cmp][mod_kmods][service:BITS][service:BITS;dname:Background Intelligent Transfer
Service;status:running][service:BITS;dname:Background Intelligent Transfer Service;status:stopped]
[221][IESEC][new][mod_ports][UDP:1146][UDP:1146;exe=C:\Program
Files\180searchassistant\salm.exe;pid=892]
```

Figure 22 Spyware-Related Changes To The IESEC Test Bed File System As Reported By Osiris.

A review of the network traffic captures for this experiment revealed the downloading of the executables through HTTP GET commands. In this instance, files 180SAInstaller.exe and RBoomerang.1 are downloaded and installed on the IE test bed. Figure 23 shows the initial GET command by the client and the following response by the servers providing the binary files. Text beginning with the letters "MZ…." pertains to the commencement of transmission of a Windows Portable Executable binary file. Areas pertaining to the names of the binary files, the servers hosting the files, and the

commencement of the binary file transmission are shown in bold. Figure 23 shows the installation of 180SAInstaller.exe from static.flingstone.com and the installation of RBoomerang.1 from installs.180solutions.com.

GET /softwares/180SAInstaller.exe HTTP/1.1
Accept-Encoding: gzip
Host: static.flingstone.com

HTTP/1.1 200 OK
Server: thttpd/2.25b 29dec2003
Content-Type: application/octet-stream
Date: Tue, 19 Jul 2005 04:27:58 GMT
Last-Modified: Wed, 29 Jun 2005 20:53:26 GMT
Accept-Ranges: bytes
Connection: close
Content-Length: 419976

MZ..........@.....!..L.!This program cannot be run in DOS mode.
$.....{.....,J.....U.....u.....H.....P.....,H.....u.....J.....C.K.....O.....Rich.....
PE..L....G.B.....
.....@.....@.....p.....6.....i.....P.....`.....
..

GET /downloads/boom/2.0/RBoomerang.1 HTTP/1.1
Accept: */*
Host: installs.180solutions.com

HTTP/1.1 200 OK
Server: Apache
Last-Modified: Tue, 31 May 2005 20:47:24 GMT
ETag: "1509727-17000-429ccd5c"
Accept-Ranges: bytes
Content-Length: 94208
Content-Type: text/plain
Date: Tue, 19 Jul 2005 04:27:14 GMT
Connection: keep-alive

MZ..........@.....!..L.!This program cannot be run in DOS mode.
$.....3..i]..i]..i]..eR..i]..e...i]..JD..i]..a...i].aa...i]..i\.wi]..e=..i]..b...i]..e...i].Rich.i].....PE..L.....r@.....
.....
.....m.....@.....

Figure 23 HTTP Get Commands Downloading 180Solutions Related Spyware Executable Files

Table 7 provides a list of infection-related downloads during the course of two separate experiment runs for each platform. During the first experiment execution, the IE platform was assigned IP address 10.10.10.31 and the IESEC platform was assigned 10.10.10.41, respectively. During the second experiment execution, IE was given IP

address 10.10.10.30 and IESEC was given IP address 10.10.10.40. This table shows various files with extensions ".exe", ".dll", and ".cab" being downloaded from common servers.

Table 7 Online Travel Infection-Related Downloads

Test Bed	Source	Destination	HTTP GET
IE	10.10.10.31	205.205.86.51	/cab/MediaAccessVerisign/ie/Bridge-c139.cab
IE	10.10.10.31	205.205.86.51	/Release/v21/MediaGateway.exe
IE	10.10.10.31	67.114.52.28	/downloads/dll/5.0/clienthook.dll
IE	10.10.10.31	205.240.15.19	/cdt/setup4030.cab
IE	10.10.10.31	146.82.109.210	/Dnl/T_50245/toolbar3.cab
IE	10.10.10.30	205.205.86.51	/cab/MediaAccessVerisign/ie/bridge-c420.cab
IE	10.10.10.30	205.205.86.51	/Release/v21/MediaGateway.exe
IE	10.10.10.30	205.240.15.19	/cdt/setup4030.cab
IE	10.10.10.30	67.114.52.28	/downloads/dll/5.0/clienthook.dll
IE	10.10.10.30	146.82.109.210	/Dnl/T_50245/toolbar3.cab
IESEC	10.10.10.40	205.205.86.51	/cab/CDT/ie/bridge-c420.cab
IESEC	10.10.10.40	205.205.86.51	/cab/CDT/ie/bridge-c420.cab
IESEC	10.10.10.41	205.205.86.51	/cab/WebsiteAccess/ie/Bridge-c139.cab
IESEC	10.10.10.41	205.205.86.51	/Release/v21/MediaGateway.exe
IESEC	10.10.10.41	205.240.15.19	/cdt/setup4030.cab
IESEC	10.10.10.41	69.225.175.11	/downloads/dll/5.0/clienthook.dll
IESEC	10.10.10.41	146.82.109.210	/Dnl/T_50245/toolbar3.cab
IESEC	10.10.10.41	69.28.178.8	/io/downloads/3/nem220.dll

D. REAL ESTATE SECTOR

The real estate sector experiment was found to infect the IE test bed. Table 8 provides a break-down of spyware as detected by each of the anti-spyware scanning tools. As in Table 6, the numbers to the left of the slash symbol pertain to spyware counts while the numbers to the right pertain to the number of signatures detected by each of the scanners.

Table 8 Spyware Infection in the Real Estate Sector

Test Bed	SpyAudit	Ad-aware	Spybot S&D	Microsoft AntiSpyware
IE	6	8 / 152	11 / 60	14 / 613
IESEC	0	0	0	0
FF	0	0	0	0
FFSEC	0	0	0	0
PASSIVE	0	0	0	0

A review of the integrity checker log revealed numerous changes made to the file system, similarly observed as in the case with the online travel sector. As with Figure 21 and Figure 22, the output of Osiris was edited for presentation purposes. Changes made to the test bed clearly show files and folders associated with 180Solutions, Optimizer, ProSiteFinder, and Web_Rebates, to name a few. All of these programs were identified as spyware by the anti-spyware scanning tools.

```
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\cfout.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\djtopr1150.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\jkill.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\umqltg4cl_.exe]
[203][IE][new][C:\Program Files\180searchassistant\salm.exe]
[203][IE][new][C:\Program Files\180searchassistant\salmhook.dll]
[203][IE][new][C:\Program Files\Internet Optimizer\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\install.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\optimize.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\install.exe]
[203][IE][new][C:\Program Files\Media Gateway\Info.txt]
[203][IE][new][C:\Program Files\Media Gateway\MediaGateway.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\38ts9dio.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\40754594.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\66708108.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\9c172bf2.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\9sva9xma.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\Uninstall.EXE]
[203][IE][new][C:\Program Files\ProSiteFinder\gx3x3ujv.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinder.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinderh.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\wbk6f8lb.DLL]
[203][IE][new][C:\Program Files\Web_Rebates\README.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\log.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rb.htm]
```

```
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates0.exe]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates1.exe]
[203][IE][new][C:\Program Files\Web_Rebates\disp1150.exe]
[203][IE][new][c:\windows\7if8uge6.exe]
[203][IE][new][c:\windows\Downloaded Program Files\ClientAX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\MediaGatewayX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\clientax.inf]
[203][IE][new][c:\windows\fwnkjsb.exe]
[203][IE][new][c:\windows\nem220.dll]
[203][IE][new][c:\windows\system32\atl71.dll]
[203][IE][new][c:\windows\system32\bn9v8uqv.exe]
[203][IE][new][c:\windows\system32\drivers\PROCEXP.SYS]
[203][IE][new][c:\windows\system32\fka3ngn0.dll]
[203][IE][new][c:\windows\system32\irkuap4b.exe]
[203][IE][new][c:\windows\wsem303.dll]
```

Figure 24 Spyware-Related Changes To The IE Test Bed File System As Reported By Osiris.

As with the online travel sector experiment, various attacks were observed in the network traffic capture file. Figure 25 shows the initial steps associated with the installation of spyware. The first block of text shows the IE client requesting web site www.angioedema-research.dzwonki.pruszkow.pl, a web site which was initially included in the list of URLs associated with the real estate sector. It is apparent this URL is not associated with real estate as previously defined and thus is considered one of the previously identified “noise” web sites. This web site contacts URL www.xxxcenter.org. This URL has been identified by various anti-spyware-related web sites as being associated with spyware software distribution. The next phase of the attack consists of the downloading of script “prompt_ie_win.js” hosted at web server static.windupdates.com. This server appears repeatedly in network traffic capture files and it is associated with the downloading of several spyware-related programs. The last phase of the attack shows the static.windupdates.com server transmitting a javascript file.

A detailed analysis of this script was not performed, however, it is strongly suspected this script launches the downloading and installation of MediaGateway.exe. Areas shown in bold pertain to the above referenced discussion.

GET / HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: www.angioedema-research.dzwonki.pruszkow.pl
Connection: Keep-Alive

HTTP/1.1 302
Date: Mon, 18 Jul 2005 20:08:37 GMT
Server: Apache/1.3.33 (Unix)
X-Powered-By: PHP/4.3.11
Set-Cookie: Hose_Stat_Visited=1; expires=Tue, 19 Jul 2005 20:08:42 GMT
Location: http://www.xxxcenter.org
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8858-2
Content-Language: pl

GET /prompts/js/prompt_ie_win.js HTTP/1.1
Accept: */*
Referer: http://www.xxxcenter.org/
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: static.windupdates.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: thttpd/2.25b 29dec2003
Content-Type: application/x-javascript
Date: Mon, 18 Jul 2005 20:09:07 GMT
Last-Modified: Wed, 20 Apr 2005 20:04:36 GMT
Accept-Ranges: bytes
Connection: close
Content-Length: 2329

/*FLxwjZ1GS Eao*/var _c3b=_p_rm;var _r3W=_i_ci;var _e35=_p_cu;var _A3F=_p_lf;var _e3p=_p_cm;var _a3W=_p_dl;var _I38=_p_ry;var _z3z=_p_cd;var _O35=_p_pr;var _W3K=_i_br;var _Q33=_p_cl;var _a3M=_p_ct;var _k3o=_i_da;var _T3f=_p_cr;var _u3P=0;var _d3l=unescape;var …

Figure 25 Infection Setup by xxxcenter.org

Table 9 provides a list of infection-related downloads during the course of the real estate test bed experiment. This table shows test bed IE with IP address 10.10.10.30 accessing six separate servers and downloading a total of six binary files. A seventh

entry shows the transmission of encrypted information to s.dll at IP address 199.221.131.90. This file is associated with Sah spyware, identified by the anti-spyware scanning tools. The first line in this entry shows the text "command=update," possibly associated with the installation of additional spyware programs. This line is shown in bold. Decryption of the parameter was not attempted. Files with extensions ".exe", ".dll", and ".cab" are shown on this table.

Table 9 Online Travel Infection-Related Downloads

Test Bed	Source	Destination	HTTP GET
IE	10.10.10.30	205.205.86.51	/cab/MediaAccessVerisign/ie/bridge-c420.cab
IE	10.10.10.30	205.205.86.51	/Release/v21/MediaGateway.exe
IE	10.10.10.30	67.114.52.28	/downloads/dll/5.0/clienthook.dll
IE	10.10.10.30	downloads.shopathomeselect.com	/cdt/setup4030.cab
IE	**10.10.10.30**	**199.221.131.90**	**/s.dll?MfcISAPICommand=update&pa**r am=%b0%01%01%001_r8Q-Q6VQI4B_9o9w6Rl4t5VH8dmaj6dsqDY7e8nvj3E8iLA-Ra678l6OJtiz8_vI5TjN9gVfp8JLegYsa6L3ZqOYi44BetlYWGfx4nJKwMDwEDSvxJkPAm_dSkuDAoT0SJwZ_L13Ya6B02N_461xmpFujbpQtxwilw29USIxTymfXlSZA4p2hNDsrKLlJUf_i_n8DjAD0CGgYCEeIpLHjNcpi88WM9kt894vm-utOfexnX6tU_IijXLfz11COtGGuDF4adbVWPcPObn93g0Ix09BMUXovGrOA4rLQ4LHFESX-WE5eyj0CRJnx9HhBEqEDYSCLiguzh79hNfK1edKPrM3UIoImO4F8yl6Fg0BOOOBQStGpp2PrUYEZ2TbZ__ebj9iYX0CdR6jh04biJ8mG29_3EdKelce--pO1Pw7nDtBDmlDfS5mLmUTuvfOnfxjhOij0U8cGk3OLUj6g08dxe_y3NrbN5nVpvc4JQ86UTWS511UpYd44dgz6o1VY_rL9sYMUpXXk_Enrm4ZAKV1a6uJAKQ8Aq2n4OeUdFVo3l0ZNCAA2RnJCHd1wtbf_H
IE	10.10.10.30	146.82.109.210	/Dnl/T_50245/toolbar3.cab
IE	10.10.10.30	cdn-68-142-79-68.lax.llnw.net	/io/downloads/3/nem220.dll

E. REMAINING SECTORS

Infection of spyware was not found in the remaining sectors of the Internet. Anti-spyware scanning tools and integrity checking software was used to determine whether the virtual machine of each sector warranted in-depth analysis. Therefore, based on these preliminary results, child, university, government, military, and insurance-related sectors

of the Internet tested by this experiment were not found to have been infected and thus were not analyzed in further detail. It should be noted that although preliminary analysis reported the presence of one spyware related software in the insurance sector, further analysis indicated that the baseline used for the experiment may have been corrupted. A detailed explanation of this is provided later in this chapter.

Table 10 provides a summary of the number of infections detected by each of the scanning tools in each of the three high risk Internet sectors. This table shows both IE and IESEC test beds becoming infected while visiting web sites in the adult entertainment and hacker-related sectors of the Internet.

Table 10 Spyware Infection in High Risk Internet Sectors

Sector	Test Bed	SpyAudit	Ad-aware	Spybot S&D	Microsoft AntiSpyware
Adult Entertainment					
	IE	5	8 / 103	10 / 40	10 / 270
	IESEC	5	8 / 105	10 / 40	10 / 267
	FF	0	0	0	0
	FFSEC	0	0	0	0
	PASSIVE	0	0	0	0
Hacker/Warez					
	IE	16	17 / 395	23 / 180	30 / 1021
	IESEC	16	18 / 391	23 / 182	30 / 794
	FF	0	0	0	0
	FFSEC	0	0	0	0
	PASSIVE	0	0	0	0
Online Gambling					
	IE	0	0	0	1 / 1
	IESEC	0	0	0	0
	FF	0	0	0	0
	FFSEC	0	0	0	0
	PASSIVE	0	0	0	0

One infection was detected by Microsoft's AntiSpyware in the online gambling sector. This infection was reported to be the KrAIMer remote access Trojan and the command.exe Windows file was reported to have been replaced. However, the Osiris host integrity monitoring software did not report changes in this file. None of the Firefox-based test beds or the PASSIVE test bed reported spyware infection.

F. GENERAL EXPERIMENT FINDINGS

Through the course of the experiment several common spyware-related files and servers associated with these files were noted. These sources of infection were observed for both the real estate and online travel sectors. Table 11 shows a list of those files and associated servers. This list is not meant to be comprehensive in nature but rather a list of the most common sources of infection. Numerous other sources may be responsible for infection in the hacker and warez-related sector and in the adult entertainment-related sector of the Internet. However, due to time constraints and the fact that these two sectors where only included in the experiment to allow a preliminary comparison between safe and unsafe areas of the Internet, detailed analysis of sources of infection was not conducted.

Table 11 Observed Infectious Binaries And Associated Servers

Malicious Binaries	Servers
Bundle_cdt1006.exe	tatic.flingstone.com
cdt1006.sah setup4030.cab	Downloads.shopathomeselect.com
Toolbar3.cab	download.websearch.com
Bridge-c139.cab bridge-c420.cab MediaGateway.exe	static.windupdates.com
clienthook.dll	installs.180solutions.com
s.dll	gr2.cc
Nem220.dll	cdn2.movies-etc.com
agentprefs2.sah validate.sah global.sah	www.shopathomeselect.com
optimize.exe	cdn.climaxbucks.com
ProSiteFinder.exe	sds.qckads.com

Figure 26 and Figure 27 show a comparison of the four anti-spyware scanning tools in the IE and IESEC test beds for each of the experiment sectors. These two figures show that Microsoft AntiSpyware consistently reports a higher number of spyware hits across the various sectors. It is uncertain whether this information supports the findings of [31] since the data provided in that study pertains to the successful removal of adware while the data presented in these two figures pertains only to the detection of spyware. It does appear, however, that the success of detection increases from left to right starting

with Earthlink's SpyAudit, Ad-Aware, Spybot Search and Destroy, and Microsoft AntiSpyware, respectively. Earthlink's SpyAudit performed as expected given the fact that it performs a far faster, and presumably less comprehensive, scan of the system when compared against the other three scanning tools.

A second important observation from these two figures is the fact that both the IE and the IESEC test beds were considerably infected by not just the high risk sectors of the Internet such as adult entertainment and hacker and warez-related web sites but also by web sites found within the online travel sector. Three web sites were identified as malicious and responsible for the infections in this sector, based on the name of the URLs, these web sites did not appear to be false positives. This indicates that although conducting the experiment in July of 2005 with a fully patched Windows XP platform, including Service Pack 2 and subsequent patches, the performance of the IESEC test bed was not significantly different from that of the IE test bed when encountering these three Internet sectors. This data reveals that zero-day exploits were encountered by the test beds during the course of the experiment.

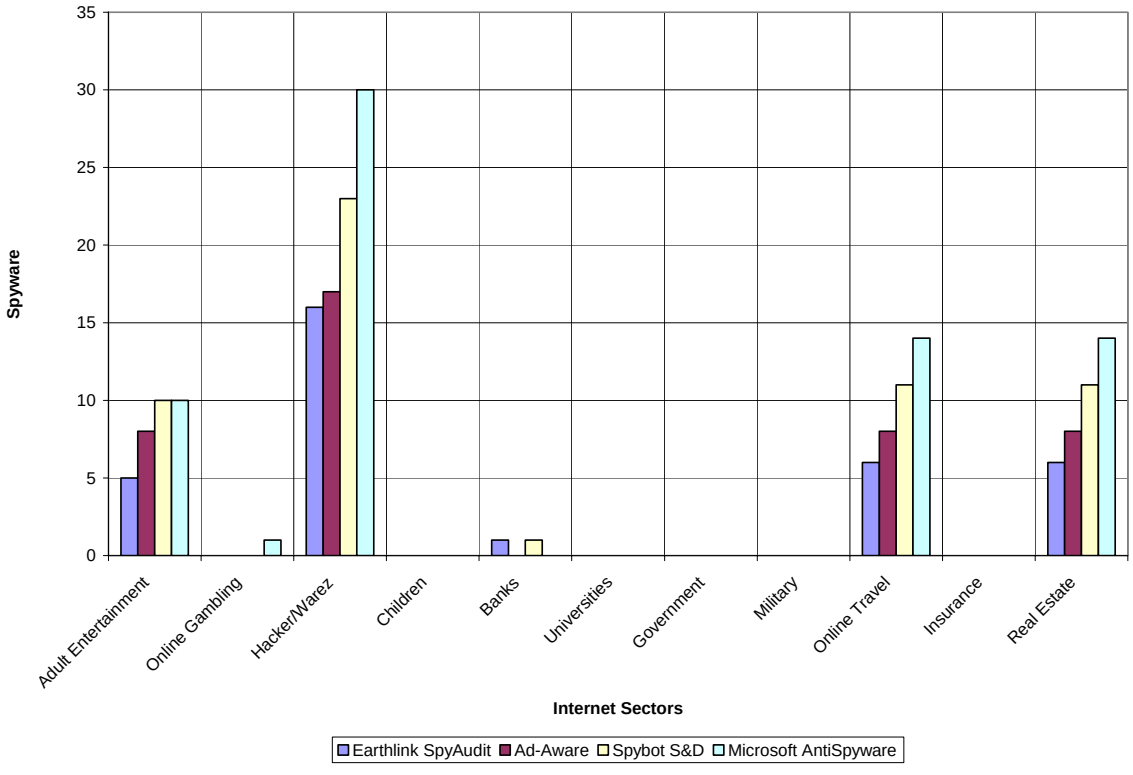


Figure 26 IE Test Bed Spyware Infection by Sector

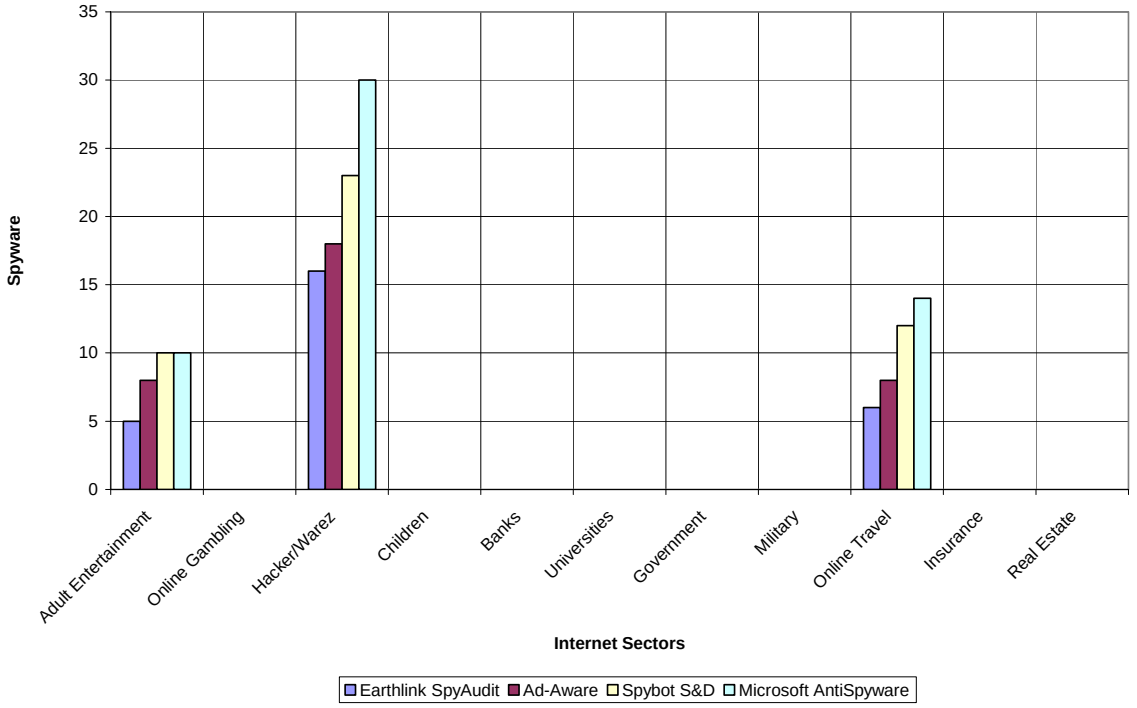


Figure 27 IESEC Test Bed Spyware Infection by Sector

Figure 28 and Figure 29 provide comparisons between all five separate test beds (IE, IESEC, FF, FFSEC, and PASSIVE) for each of the Internet sectors tested and for each of the platforms, respectively. Figure 29 clearly shows two of the high risk sectors, adult-entertainment and hacker and warez-related sectors with significant infection for both IE and IESEC test beds. For the purposes of these two figures, the total infections reported by each of the anti-spyware scanning tools were added for each of the test beds. Only Microsoft Internet Explorer-based test beds reported spyware infection. Additionally, the online travel sector reported a greater number of spyware infections than the adult-entertainment sector. Lastly, the real estate sector showed a dramatic difference in infection between IE and IESEC test beds. Apparently Service Pack 2 and subsequent patches installed in IESEC prevented infection encountered in this sector. A single hit was reported for the IE test bed in the online gambling, and two hits were reported for the bank and insurance sectors. It should be noted that two hits are reported

for the latter sectors due to the summation of anti-spyware scanning results for each test bed. In the case of the bank and insurance sectors, both Earthlink's SpyAudit and Spybot Search and Destroy reported each an instance of spyware.

Figure 29 shows that spyware infections associated with the adult entertainment, hacker and warez, and online travel-related sectors of the Internet was not significantly diminished with the installation of Service Pack 2 and additional patches.

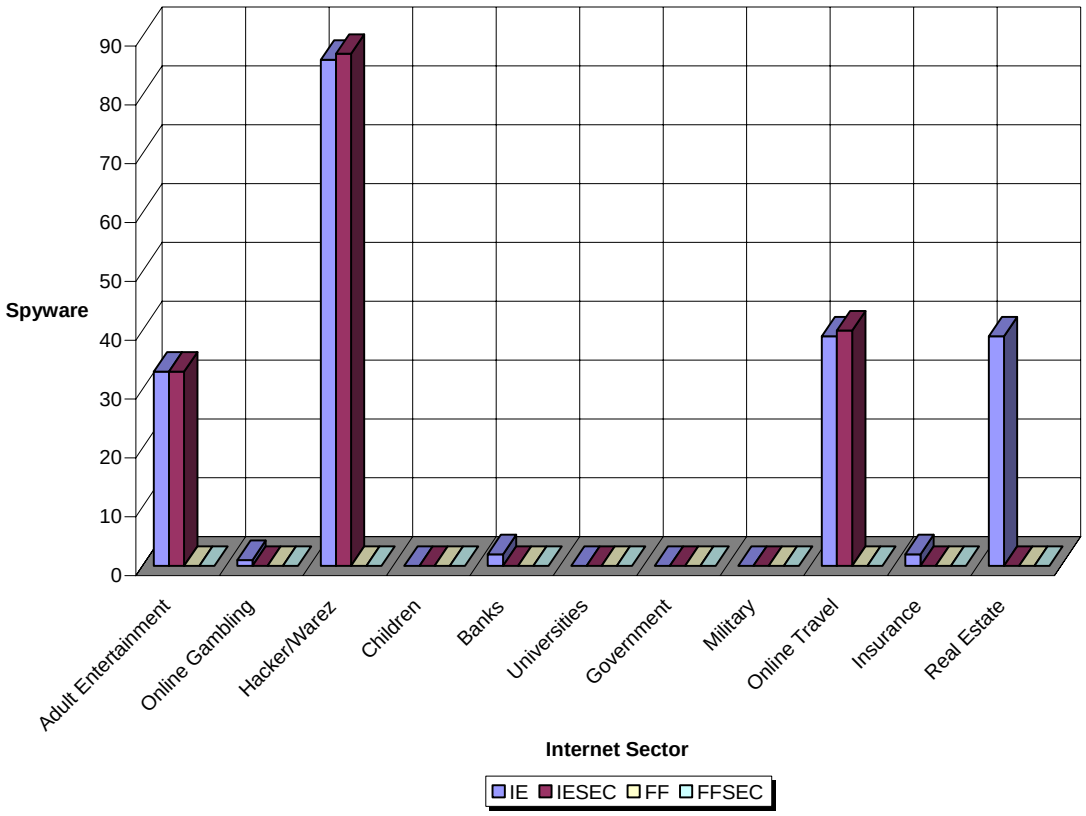


Figure 28 Test Bed Infection Comparison by Sector

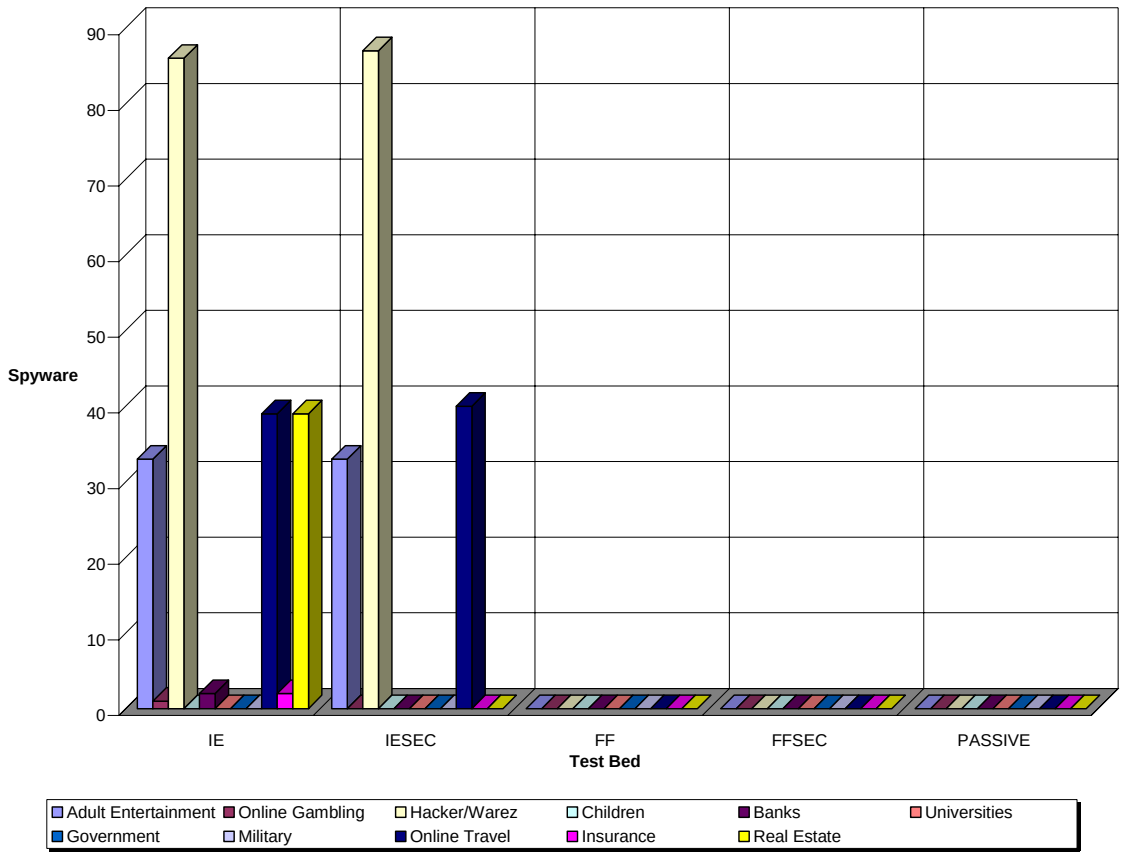


Figure 29 Test Bed Infection Comparison by Platform

G. MALICIOUS WEB SITES

Analysis of the experiment data lead to the identification of various web sites as responsible for infecting the test beds with spyware. These web sites are not specifically related to a particular industry but rather associated with the sector of the Internet in which they came up as a result of a search query in Google™ or as otherwise noted in Chapter IV. Therefore, as in the case of the twelve malicious web sites found in the real estate sector, none of them are truly related to either property real estate or Internet real estate as previously defined. Nevertheless, web sites identified as contributors to infection of the test beds are reported under the sector in which they conducted the infection. The following is a list of these web sites:

Bank Sector:

- Bank http://URL # 279

Online Travel Sector:

- agent.sjuper.com
- agent-travel.cdeedc.com
- cher-ticket.sjuper.com

Real Estate Sector:

- www.angioedema-research.dzwonki.pruszkow.pl
- www.conan-the-adventurer.dzwonki.pruszkow.pl
- www.discounted-colognes-online.dzwonki.pruszkow.pl
- www.fingernail-fungus.dzwonki.pruszkow.pl
- www.girlsgonewildcom.dzwonki.pruszkow.pl
- www.hustler-barely-legal.dzwonki.pruszkow.pl
- www.insulin-syringes.dzwonki.pruszkow.pl
- www.nationwide-process-servers.dzwonki.pruszkow.pl
- www.penn-yann-1962.dzwonki.pruszkow.pl
- www.printable-stitchery-patterns.dzwonki.pruszkow.pl
- www.snoring-remedies.dzwonki.pruszkow.pl
- www.unix-time-format.dzwonki.pruszkow.pl

H. POSSIBLE BASELINE CORRUPTION

During the course of this analysis, it was noted that both the banking and insurance sectors were found to have identical infection reports as provided by the anti-spyware scanning tools. According to SpyAudit, an adware category software identified as Spy#6d78b was reported present in the IE test bed. Spybot Search & Destroy reported the presence of EffectiveBandToolbar in the same IE test bed. The anti-spyware scanning tools reported the presence of these two items in the baselines as well as in the experiment virtual machines.

Analysis of the network traffic captured during the course of each experiment found that the X.exe and JScript attacks described in Chapter VI Section A were present in the banking sector experiment but not in the insurance sector experiment. The banking

sector experiment collected data on 10 July 2005 while the insurance sector experiment collected data on 11 July 2005. Similar anomalies were not observed in the remaining test bed baseline images. This information supports the theory that an apparent corruption of the baseline images of the host computers in the banking and insurance sectors took place between 10 and 11 July 2005. The lack of corresponding exploit or attack related network traffic in the insurance sector experiment indicates that the infection of the insurance baseline image took place during the execution of the banking sector experiment.

Furthermore, infection of the banking test bed baseline image is suspected to have occurred during the configuration of the test beds in preparation for the insurance sector experiment since the network traffic captures show downloading of the X.exe file as well as of the suspected malicious JScript script previously described. Lastly, attempts to replicate infection of a similarly configured test bed approximately six weeks after the initial data collection did not succeed, perhaps due to the dynamic nature of web servers and exploit use.

I. FALSE POSITIVE URLS

The selection of URLs for each of the Internet sectors tested in the experiment lead to the identification of URLs previously referred to as "noise" or false positives. These are URLs which, although found as a result of a particular query in the Google™ search engine, are not truly associated with the sector of the Internet intended to be tested. The criteria used in determining these false positives consisted of:

- Web sites which do not sell or provide services or products associated with the industry in question.
- Web sites consisting of generic non-sector-specific search directories.
- Web sites which return a custom invalid URL or request error page within the requested domain.

An estimate of the number of false positive URLs was calculated by visually inspecting 75 URLs from the insurance, child, real estate, and online travel-related sectors. Web sites in government, military, and university-related sectors were obtained through means other than a Google™ search query and are considered to contain few or no false positives. Table 12 shows the number of false positives encountered when visually inspecting 75 web sites in each of these sectors. An estimate of the number of

false positive URLs present in a set of 500 URLs is also provided. Calculations were made using the hypergeometric distribution model with at least a 95% confidence.

Table 12 False Positive URLs by Sector

Sector	Sector Size	False Positives	False Positive URLs Estimate (for 500 web sites)	Confidence
Insurance	75	1	1 to 30	0.95181
Children	75	9	30 to 101	0.95007
Real Estate	75	31	156 to 260	0.95029
Online Travel	75	2	3 to 40	0.95181

J. SUMMARY

This chapter presented the findings of the in-depth analysis of the infection of various test beds by spyware. Spyware infected test beds in the banking, online travel, and real estate-related sectors of the Internet. Additionally, and as expected, high risk sectors of the Internet also infected the test beds with spyware. The data shows that only Internet Explorer-based test beds were infected by spyware while Firefox-based test beds were not. Furthermore, both unpatched and patched test beds were infected, supporting the conclusion that zero-day exploits were encountered by the experiments.

Chapter VII will discuss three other research studies conducted in the area of spyware infection and detection, and how they compare with the experiment reported here.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. RELATED WORK

This chapter discusses and compares three recent spyware-related studies with the work conducted in this research.

A. UNIVERSITY OF WASHINGTON PROJECT

The University of Washington Computer Science and Engineering department conducted a study on the presence of spyware within their network in 2003 [46]. The study provided a background introductory description of spyware, discussing various types of spyware programs and typical manner of host infection. The study also described four spyware programs in particular, providing a detailed description of their behavior, and discussing vulnerabilities associated with two of these programs.

The study uses definitions provided by the freeware software Spybot Search and Destroy in discussing various types of spyware such as browser hijackers, key loggers, tracks, malware, spybots, and adware. The study considers cookies and web bugs to be spyware.

The study collected network traffic over a period of one week starting on August 26, 2003. The campus network was reported to contain between 40,000 to 50,000 hosts with an average bandwidth of 238 Mb/s between the university domain and its ISP. Passive network traffic analysis was conducted and identification of infected hosts and spyware associated servers was performed through the analysis of HTTP connections. Correlations were made among client behavior and spyware infection, demonstrating that the greater the number of different web servers visited and the greater the number of executable files downloaded, the more likely the host would be infected by spyware. Additionally, peer-to-peer program installation and spyware presence appeared to be related, showing that in 38% of Kazaa-containing hosts, spyware appeared to be present.

Spyware was noted to have successfully penetrated most organizational boundaries; finding that 69% of the organizations within the university environment contained at least one host infected with at least one variety of spyware. Regardless of security policies and perimeter protection, spyware successfully infected internal users.

The report concluded that 5.1% of active hosts on campus were infected and that the security practices were not effective in preventing infection. Additionally, vulnerabilities in Gator and eZula programs allowed arbitrary installation and execution of programs, revealing a significant security risk to infected hosts.

The current study is similar to the University of Washington study in that network traffic analysis was performed to identify infection of the test beds in the experiment phase of the work. While their study monitored traffic associated with 40,000 to 50,000 hosts, and the experiment performed here monitored traffic generated by four test beds, their study did not provide empirical evidence for the hard drive of infected computers as was done here. Finally, the University of Washington study did not consider infection as a result of surfing different sectors of the Internet.

B. MICROSOFT PROJECT

Microsoft Corporation has undertaken the study of spyware through at least three different studies [62, 63, 64]. Beginning with the concept of AskStrider, a scanning tool which "automatically scans a system for active components, matches them against a change log to identify recently updated and hence more interesting state, and searches for context information to help users understand the changes" [63]. This tool attempts to show recently changed files or states in a system by comparing against System Restore points, reviewing running processes and loaded modules, and reviewing all device drivers loaded into the system. It queries file change history and ranks these files based on file age, showing the degree of system stability. A case study is discussed in which drive-by-downloads are responsible for the installation of HotBar Internet Explorer extensions. Infection by Internet Washer is also discussed in a separate case study. In both instances, AskStrider shows a change has been made to the system state. The tool can also be of assistance in troubleshooting problems associated with installations or upgrades of legitimate applications.

In the second study [64], the concept of Auto-Start Extensibility Points (ASEPs) is introduced and a framework – Gatekeeper – is established for the detection of spyware infection. According to the study, spyware can utilize various ways to start and maintain a process in a compromised system. By monitoring these ASEPs, the study claims to be able to detect certain spyware infections. The study relies on signature based scanning

tools such as Ad-Aware and Spybot Search and Destroy to detect initial infection. Gatekeeper provides continuous monitoring of ASEPs, including the recording, alerting, and blocking of any suspicious or undesirable ASEP hooking operations. The study identified 34 ASEPs in 5 categories: ASEPs associated with the start of new processes, ASEPs that hook system processes, ASEPs that load drivers, ASEPs that hook multiple processes, and Application-specific ASEPs. Statistics were provided on the various ASEPs hooked by spyware showing the use of Browser Helper Objects and Run commands as the two most often used out of 34 ASEPs. Gatekeeper integrates with System Restore and AskStrider. Combined, these led to the discovery of 17 previously undiscovered ASEPs. The study concludes by asserting that the questions of "Where did it come from?"; "When was it installed, where was it installed, and what was installed?"; "How does it get instantiated?"; and "How do I disable/remove it?" are addressed by the study, leaving for future study the question "What does it do?"

The two Microsoft studies discussed above led to a third report [62] pertaining to the identification of browser-exploiting malicious web sites. The concept of an Automated Web Patrol is proposed as a way to significantly reduce the costs associated with monitoring malicious web sites. The authors used programs designed to control Internet Explorer, named "monkeys," to visit malicious web sites searching for browser exploits. The approach consisted of selecting 5000+ potentially malicious URLs from lists compiled by the anti-spyware community and visiting these web sites with monkeys. The monkeys operated within virtual machines. The virtual machines are used to facilitate the reinstallation of the test bed upon detection of a malicious web site and did not interact with the web sites other than by simply displaying the page. Monkeys with various patch levels were used, starting with the lowest patch level (Windows XP SP1 unpatched). When a monkey encountered a malicious web site, it would shutdown, reset its state, and forward the malicious URL to the next higher patch-level monkey. When the highest patch level monkey (Windows XP SP2 fully patched) detected infection, they assumed that a zero-day exploit was discovered. This detection framework made use of Strider and Gatekeeper. Additional links encountered in each of the identified malicious web sites were followed, leading to further malicious web site identification.

This study identified 752 malicious URLs. Additionally, a web site using a zero-day exploit was found in early July 2005. The study also provides a comparison of the number of exploit URLs provided by each of three different popular search engines (Google, Yahoo, MSN). The study also mentions that the top one million click-through links from an unnamed search engine are being monitored in order to asses if the "exploit industry" has penetrated popular sites.

The study is similar to the experiment conducted here in that the concept of ASEPs are used to identify infection by specific URLs. Scripts written here collected many auto-start data points mentioned in this and other studies as potential ways to identify spyware infection. Furthermore, in both studies the preliminary analysis for possible spyware infection is conducted with Spybot Search and Destroy and Ad-Aware software, although the current experiment uses two additional scanning tools and a host integrity scanner. Both the above referenced studies and the current experiment, visited web sites and attempted to document infection by spyware as a result of drive-by-download exploits, requiring no interaction between the user and visiting web site.

Significant differences exist in the manner in which URLs used in the experiment were collected. The Microsoft researchers collected URLs from "host" files provided by anti-spyware scanning tools while the current study collected URLs through a combination of search engine queries and web site listings (e.g., FDIC bank list). Finally, while this study mentions the monitoring of popular web sites served-up by search engines, the current study utilized a sector model of the Internet and a statistically significant sampling of sites from each Internet sector.

C. UNIVERSITY OF NEBRASKA PROJECT

Researchers at the University of Nebraska-Lincoln conducted an experiment between mid-October to mid-November of 2004 in which a total of 600 web sites were visited [49]. The experiment consisted of individually visiting each of the web sites and interacting with the site in a manner to "simulate the behavior of naïve users." Such behavior included clicking on advertisement banners and selecting "yes" when prompted to download or install software. These web sites were obtained from www.trafficranking.com, a provider of market research information on the most visited web sites on the Internet. The researchers utilized a "trust gauge" provided by

TrustGauge™ in determining the degree of trustworthiness associated with each of the web sites. The experiment was organized into four sectors of the Internet. The sectors of the Internet were comprised of E-Commerce, Recreation and Entertainment, Download Search and Directory, and News and Education related web sites. Each of these four sectors was further subdivided into groups of 50 web sites according to a low, medium or high TrustGauge rating, for a total of 150 web sites per Internet sector. The study used Ad-Aware and Spybot Search and Destroy to detect spyware infection. As with the other experiments [46, 62], the problem concerning spyware is recognized to be large in scope. The study concluded that user browsing behavior is "responsible for much of the spyware dissemination on computers"

Significant differences exist between the Nebraska study and the experiment conducted here. While the Nebraska study [49] involved significant interaction with each visited web site by clicking on banners, links, and accepting any offer to install software presented to the user, the current experiment demonstrated the risks of surfing with minimal interaction. Additionally, the current experiment involved more and larger Internet sectors. Lastly, the concept of "safe" web sites is very different in the two studies. Where the Nebraska study used the analysis of a third party, based on web seals and rankings, the current experiment designated various sectors of the Internet as safe or unsafe based on business type and other orientations.

D. CONCLUSIONS

While many similarities exist in work performed in the experiment reported here and previously published spyware-related reports, significant differences exist in the methodology, manner in which URLs were collected and grouped in Internet sectors, and the level of interaction between user and web site. Nevertheless, the results of all four studies, while utilizing different examination tools and techniques, are largely conformant in their conclusions and together provide a large corpus of data for future study, and a significant contribution to understanding vulnerabilities in the Internet today. The experiment and work conduced here lead to the conclusions discussed in the following chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSIONS

This thesis has resulted in a new definition of spyware and describes the extent of its presence, browser effectiveness in stemming infection, system patching effectiveness, relative perceived risk associated with various Internet sectors, perceived detection rates by various scanning tools, and penetration of malicious web sites into search engine results. The following sections summarize the conclusions arrived at as a result of this work.

A. DEFINITION OF SPYWARE

Since its modest beginnings in the late 1990's, spyware has steadily grown in complexity and presence, becoming a major security concern in just a few years. Defining spyware has been a challenge to security professionals due to the convergence of common technologies and the dynamic nature of software development. Work conducted as part of this thesis has lead to the following new spyware definition:

> Spyware is a computer program that is either (1) developed with the express purpose to steal resources or collect user or organization data, or (2) deployed with the intent to profit financially or strategically from data collected, and must have four common activities. These four activities are: hide, collect, communicate, and survive in a hostile environment. Of course, spyware of type 1 may exhibit one or more of these four activities as well.
>
> The mechanics of spyware activities are as follows. Spyware hides by using deceptive or surreptitious techniques. Spyware collects system, organizational, or personal data. Spyware communicates collected data to a remote or third party. Spyware is designed with a degree of resilience, remaining present in a system as long as possible.
>
> The extent of the information collected, and program activities or capabilities, are diminished or hidden from the user through deception or obfuscation.

Cookies and web bug technologies are not spyware because they do not posses all four basic activities (e.g., hide, collect, communicate, survive) and because they are not computer programs. The misuse of cookies and web bugs can, however, achieve end results similar to spyware. Thus it is possible to track user activities on the Internet through the use of cookies and web bugs common to affiliated web sites.

B. UBIQUITOUS PRESENCE

Spyware has penetrated personal, business and government systems despite common defense-in-depth approaches. Up-to-date patched computer systems, firewalls, and anti-virus programs have thus far failed to stem the tide of spyware infection. Empirical data from this thesis shows spyware infection is possible by visiting "safer" sectors of the Internet (e.g., banking, online travel, and real estate-related web sites) as well as "high risk" sectors of the Internet (e.g., Hacker/warez and adult entertainment-related web sites).

C. BROWSER PERFORMANCE

Comparative empirical analysis of the performance of Internet browsers with respect to the likelihood of infection by spyware through drive-by-downloads has yielded dramatic results. Experiments conducted by visiting web sites in eight "safe" and three "high risk" sectors of the Internet indicate infection through the use of drive-by-download techniques to be strictly limited to Microsoft Internet Explorer. Mozilla Foundation's Firefox Internet browser did not experience a single instance of infection.

The reasons for these differences may lie in the fact that many spyware programs are currently using ActiveX exploits. ActiveX is not supported by Firefox. Additionally, it is speculated that greater emphasis is placed on the development of spyware for an Internet browser which is used by approximately 90% of the users on the Internet. Firefox currently holds approximately 8% of the browser market.

The preponderance of the results towards Microsoft Internet Explorer is likely to change, then, as browser flaws are discovered and exploited in alternative browsers like Firefox. Evidence of this can be seen in the disclosure of flaws in the way Firefox parses through Internationalized Domain Names (IDNs), allowing an attacker to execute arbitrary code [36].

D. PATCH PERFORMANCE

While common wisdom dictates maintaining a computer system up-to-date with respect to security patches, empirical analysis of the various virtual machines has shown little difference in the spyware infection rates. Comparisons among a default installation of Windows XP and a fully-patched installation which included Service Pack 2 and numerous other operating system and browser patches available up until before the

commencement of the experiments in July 2005 reveal similar infection. Both IE and IESEC test beds showed infection when visiting web sites in the adult entertainment, hacker and warez, and online travel-related sectors of the Internet. The IE test bed showed additional infection while visiting online gambling, banks, and insurance-related sectors of the Internet.

While the results of the experiment show little difference in infection between patched and unpatched computer systems, a plausible explanation for this may reside in the use of overly permissive browser security configuration settings. The design of the experiment considered maintaining browser settings at a level which provided commonly used or desired functionality, such as allowing ActiveX and other script execution, Java applets, Flash and Shockwave. In this manner, test bed configurations are representative of commonly found systems in the real world.

E. INTERNET SECTORS

While it has been widely reported that "high risk" Internet surfing habits associated with the visiting of hacker and warez, online gambling, and adult entertainment-related web sites can lead to an increased risk of spyware infection, this thesis proposed to evaluate the safe sectors of the Internet. Child-related, banking, university, government, military, online travel, insurance, and real estate-related web sites were evaluated for the use of drive-by-download spyware infection techniques.

Consistent with commonly given advice, adult entertainment, and hacker and warez-related web sites were found to make use of drive-by-download techniques and infected both patched and unpatched test beds, with the latter sector showing the greatest number of infections.

Surprisingly, the online gambling sector of the Internet, considered high risk by the author, showed only one infection.

Both the online travel and the real estate sectors of the Internet showed spyware infection greater than that observed in the adult entertainment Internet sector. Lastly, a single infection was also noted in the banking sector.

F. ANTI-SPYWARE SCANNING TOOLS

Four different and freely available scanning tools were used to detect test bed spyware infection. These tools were comprised of Earthlink's SpyAudit, Lavasoft Ad-Aware, Spybot Search and Destroy and Microsoft's AntiSpyware.

Empirical analysis of the virtual machines and comparative analysis among the four scanning tools shows that Microsoft's AntiSpyware product consistently reported a higher number of infection, followed by Spybot Search and Destroy, Ad-Aware, and SpyAudit, in that order. Since there is no standard way of reporting results among scanning tools, these results are not necessarily indicative of the detection success rate for a give tool, but rather, they are an indication of the perceived detection rate presented to a user. Having said that, it is not surprising to see SpyAudit as the scanning tool with the smallest detection rates since it is a smaller application conducting a much faster scan, and therefore, is presumably less comprehensive.

G. SPYWARE SITE PENETRATION OF GOOGLE™

While not all URLs used in the experimental phase of this thesis were obtained as a result of Google™ search queries, all URLs for the adult entertainment, online gambling, hacker and warez, government, military, online travel, insurance, and real estate-related sectors of the Internet were. The results of this experiment show that malicious web sites have successfully gained a degree of prominence in the results returned by the Google™ search engine for the adult entertainment, online gambling, hacker and warez, online travel, and real estate-related Internet sectors.

Since the details of the Google™ search engine algorithms are not public, it is uncertain if the presence of these malicious web sites in the search results is due to the amount of traffic visiting these web sites, the number of links referring to the web sites, or the length of their presence on the Internet.

H. FURTHER WORK

The development of the experiment phase of this thesis has revealed the need for alternative and improved spyware infection testing and detection techniques. Based on the experiences noted in Chapter V, including a number of virtual machine, browser, and script crashes, it is apparent a more reliable approach is warranted. This author proposes migrating the detection technique away from real-time system registry snap shots to a

more integrated application which could instead hook system calls associated with common spyware installation activities. It is proposed to intercept system calls, and log attempted activities for subsequent analysis.

Additionally, improved integration, including scripting and logging of the test environment from outside the virtual machines is desired, similar to the work described in [62]. This approach allows for increased automation of the experiment.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Acohido, Byron, Swartz, Jon. Security Risks Swell for Microsoft's Explorer. July 1, 2004. USA Today, Money. Available at http://www.usatoday.com/money/industries/technology/2004-07-01-cyber-threat_x.htm. (Last accessed September 7, 2005)

[2] American Library Association. Great Web Sites for Kids. Available at http://www.ala.org/ala/alsc/greatwebsites/greatwebsitesforkidssealofapproval/GreatWebSitesSeal.htm. (Last accessed July 2005).

[3] Berger, Sandy. Cookies Explained. CompuKiss web site available at http://www.compukiss.com/populartopics/computercenterhtm/article816.htm (Last accessed September 20, 2005).

[4] Boyd, Christopher. Xpire/Splitinfinity.info Server Hack and Malware injection using IFRAMES Vulnerability – Condense Version. November 21, 2004. Available at http://www.vitalsecurity.org/xpire-splitinfinity-serverhack_malwareinstall-condensed.pdf (Last accessed September 12, 2005)

[5] Carnegie Mellon Software Engineering Institute. CERT Coordination Center. CERT/CC Statistics 1988-2005. Available at http://www.cert.org/stats/cert_stats.html. (Last accessed August 22, 2005)

[6] Chappel, David, Linthicum, David. ActiveX Demystified, It's invasive. It's ubiquitous. But what, exactly, is ActiveX? September 1997. BYTE.com magazine. Available at http://byte.com/art/9709/sec5/art1.htm (Last accessed August 20, 2005)

[7] Clark, David D. The Design Philosophy of the DARPA Internet Protocols, Proc. SIGCOMM '88, Computer Communication Review Vol. 18, No. 4, August 1988, pp. 106-114.

[8] CNN. Trojan virus attacks popular web sites. June 26, 2004. Available at http://www.cnn.com/2004/TECH/internet/06/25/internet.attack/ (Last accessed September 12, 2005)

[9] Cogswell, Bryce. BGInfo v4.07 software. Freeware SysInternals Website available at http://www.sysinternals.com/Utilities/BgInfo.html. (Last accessed July 10, 2005).

[10] Description of Internet Explorer Security Zones Registry Entries, Article ID: 182569, Last Review: March 17, 2005. Revision 12.3. Available at http://support.microsoft.com/default.aspx?scid=kb;en-us;182569. (Last accessed May 24, 2005).

[11] Earthlink SpyAudit. Earthlink website available at http://www.earthlink.net/spyaudit/press/. (Last accessed March 20, 2005).

[12] Edelman, Benjamin. Claria License Agreement is Fifty Six Pages Long. Available at http://www.benedelman.org/spyware/claria-license/ (Last accessed September 16, 2005)

[13] Egevang, K, Francis, P. RFC 1631. The IP Network Address Translator (NAT). Available at http://rfc1631.x42.com/ (Last accessed August 19, 2005)

[14] El-Moukaddem, Fatme; Maher, Jim; Omotosho, Andrew. Exploitation of an Internet Explorer. Available at www.cse.msu.edu/~maherjam/present.ppt. (Last accessed September 18, 2005)

[15] Ethereal Network Protocol Analyzer v.0.10.11 software. Available at http://www.ethereal.com (Last accessed July 11, 2005).

[16] Firefox 50,000,000 download announcement available at http://www.spreadfirefox.com/fifty.html (Last accessed July 2005).

[17] Ford, Jerry Lee. Learn JavaScript in a weekend, Premier Press, 2nd Edition.

[18] Gibson, Steve. OptOut – Tell Unwanted Spyware to Pack its Bags! Gibson Research Corporation. Available at http://www.grc.com/optout.htm. (Last accessed September 3, 2005)

[19] Glavan , Dalibor. ShowHelp("file:") disables security in IE – Sandblad advisory # 11. February 6, 2003. Xatrix Security website available at http:// www.xatrix.org/print.php?s=2635. (Last accessed March 18, 2005).

[20] Greenman, Catherine. Enjoy Your New Software, and Check Out the Advertisements. March 23, 2000. The New York Times on The Web. Available at http://partners.nytimes.com/library/tech/00/03/circuits/articles/23bund.html (Last accessed August 22, 2005)

[21] Handler's Diary July 20th 2005. SANS Internet Storm Center. Available at http://isc.sans.org/diary.php?date=2005-07-20. (Last accessed August 30, 2005)

[22] How to Determine and Recover from Winsock2 Corruption, Article ID: 811259, Last Review: April 20, 2005. Revision 8.0. Available at http://support.microsoft.com/default.aspx?scid=kb;en-us;811259. (Last accessed May 24, 2005).

[23] Huston, Geoff. Anatomy: A Look Inside Network Address Translators. The Internet Protocol Journal – Vol. 7, No. 3, September 2004. Available at http://www.cisco.com/en/US/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html (Last Accessed July 2005)

[24] Inventor of the Week. Hypertext. Lemelson – MIT Program. Available at http://web.mit.edu/invent/iow/bush.html. (Last accessed August 21, 2005)

[25] Irongeek, Web Buggery: Analyzing Tracking Images. October 15, 2004. AntiOnline, Maximum Security for a Connected World available at http://www.antionline.com/showthread.php?s=&threadid=263046. (Last accessed March 20, 2005).

[26] Knopf, Jim. Association of Shareware Professionals. The History of Shareware. Available at http://www.asp-shareware.org/users/history-of-shareware.asp (Last accessed August 21, 2005).

[27] Krebs, Brian. PC Users Warned of Infected Web Sites, Washington Post, Friday June 25, 2004. Available at www.washintonpost.com/wp-dyn/articles/A5524-2004Jun25.html (Last accessed July 11, 2005).

[28] Lampson, Butler W. A note on the Confinement Problem. Communications of the ACM, 16, 10. pp 613-615. October 1973.

[29] Lemos, Robert. SecurityFocus. Hidden-code flaw in Windows renews worries over stealthy malware. August 31, 2005. The Register. Available at http://www.theregister.co.uk/2005/08/31/spyware_writers_get_more_sophisticated/. (Last accessed August 31, 2005).

[30] Licklider, J.C.R., Clark, W. On-Line Man Computer Communication, August 1962.

[31] Livingston, Brian. Anti-spyware misses most malware. Available at http://windowssecrets.com/050127/#story1. (Last accessed July 2005)

[32] Macromedia Flash Player 7 available at http://www.macromedia.com/software/flashplayer/ (Last accessed July 2005).

[33] Microsoft Corporation, Reg.exe v3.0 software. Included in Windows XP.

[34] Microsoft Corporation, Tasklist.exe software. Included in Windows XP. Description available at http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prmb_tol_aqmd.asp. (Last accessed July 10, 2005)

[35] Microsoft warns of future security danger. February 20, 2005. Crit.org available at http://www.crit.org/articles/48/1/Microsoft-warns-of-future-security-danger.html. (Last accessed March 18, 2005).

[36] Naraine, Ryan. Researchers Flag Firefox Code Execution Flaw. eWeek, September 9, 2005. Available at http://www.eweek.com/article2/0,1895,1857322,00.asp (Last Accessed September 17, 2005)

[37] Ranum, M., Avolio, F. A Toolkit and Methods for Internet Firewalls. Proceedings of the summer USENIX conference, 1994. Available at http://www.avolio.com/papers/fwtk.html (Last accessed August 19, 2005)

[38] Results of the Security in ActiveX Workshop, CERT Coordination Center, Software Engineering Institute. December 21, 2000. Available at http://www.cert.org/reports/activeX_report.pdf (Last accessed August 20, 2005)

[39] Roberts, L. Multiple Computer Networks and Intercomputer Communication. ACM Gatlinburg Conf., October 1967.

[40] Russinovich, Mark and Cogswell, Bryce. Freeware SysInternals Website available at http://www.sysinternals.com. (Last Accessed July 10, 2005)

[41] Russinovish, Mark and Cogswell, Bryce. AutoRuns v8.0 software. Freeware SysInternals Website available at http://www.sysinternals.com/Utilities/Autoruns.html. (Last Accessed July 10, 2005).

[42] Russinovish, Mark. Handle v3.1 software. Freeware SysInternals Website available at http://www.sysinternals.com/Utilities/Handle.html. (Last accessed July 10, 2005).

[43] Russinovish, Mark. PSService 2.13 software. Freeware SysInternals Website available at http://www.sysinternals.com/Utilities/PsService.html. (Last Accessed July 10, 2005).

[44] Russinovish, Mark. TCPView.exe v2.4 software. Freeware SysInternals Website available at http://www.sysinternals.com/Utilities/TcpView.html. (Last Accessed July 10, 2005).

[45] Sandblad, Andreas. How to execute programs with parameters in IE. November 6, 2002. SecurityFocus website available at http://www.securityfocus.com/archive/1/298748. (Last accessed March 18, 2005).

[46] Saroiu, Stefan; Gribble, Steven D.; Levy, Henry M. Measurement and Analysis of Spyware in a University Environment, in Department of Computer Science & Engineering, University of Washington. (Proceedings of the 1st Symposium on Operating Systems Design and Implementation (NSDI), San Francisco, CA March 2004) Available at http://www.cs.toronto.edu/~stefan/publications/nsdi/2004/spyware.html. (Last accessed March 14, 2005).

[47] Schmidt, Charles, Darby, Tom. The Morris Internet Worm. The What, Why, and How of the 1988 Internet Worm. Available at http://snowplow.org/tom/worm/worm.html (Last accessed August 19, 2005)

[48] Schwartz, Ari. Ghost in Our Machines: Background and Policy Proposals on the "Spyware" Problem. Center For Democracy & Technology. November 2003.

[49] Shukla, Sudhindra, Fui-Hoon Nah, Fiona. Web Browsing and Spyware Intrusion. Communications of the ACM. August 2005. Vol. 48. No. 8. pp 85.

[50] Smith, Richard. The Web Bug FAQ. November 11, 1999. Electronic Frontier Foundation available at http://www.eff.org/Privacy/Marketing/web_bug.html. (Last accessed March 20, 2005).

[51] Staff Report. Federal Trade Commission. Monitoring Software on your PC:Spyware, Adware, and other Software. March 2005. Available at http://www.ftc.gov/os/2005/03/050307spywarerpt.pdf. (Last accessed September 3, 2005).

[52] Team Register. Bofra Exploit Hits Our Ad Server Supplier. November 2004. Available at http://www.theregister.co.uk/2004/11/21/register_adserver_attack/print.html (Last accessed September 12, 2005)

[53] The National Strategy To Secure Cyberspace. February 2003. The White House.

[54] Tomlinson, Ray. The First Email. Available at http://openmap.bbn.com/~tomlinso/ray/home.html. (Last accessed August 18, 2005)

[55] U.S. Department of Energy, Computer Incident Advisory Capability (CIAC). M-060: Sun Bytecode Verifier Vulnerability. Available at http://www.ciac.org/ciac/bulletins/m-060.shtml. (Last accessed August 20, 2005)

[56] U.S. Supreme Court. Jacobellis v. Ohio, 378 U.S. 184 (1964). FindLaw for Legal Professionals available at http://caselaw.lp.findlaw.com/scripts/getcase.pl?court=US&vol=378&invol=184. (Last accessed September 8, 2005)

[57] United States Computer Readiness Team, Microsoft Internet Explorer Does Not Properly Interpret IFRAME elements when Displaying URLs In The Status Bar, Vulnerability Note VU#960454. Available at http://kb.cert.org/vuls/id/960454. (Last accessed August 17, 2005)

[58] United States Computer Readiness Team, Microsoft Internet Explorer Vulnerability To Buffer Overflow via FRAME and IFRAME Elements, Vulnerability Note VU#842160. Available at http://kb.cert.org/vuls/id/842160. (Last accessed August 17, 2005)

[59] United States Computer Readiness Team, Microsoft Internet Explorer does not properly validate source of redirected frame, Vulnerability Note VU#713878. Available at http://kb.cert.org/vuls/id/713878. (Last accessed August 17, 2005)

[60] United States Computer Readiness Team, Microsoft Windows Virtual Machine (VM) ByteCode Verifier fails to properly check Java applets for malicious code, Vulnerability Note VU#447569. Available at http://www.kb.cert.org/vuls/id/447569. (Last accessed August 20, 2005)

[61] United States General Accounting Office. Critical Infrastructure Protection – "ILOVEYOU" Computer Virus Highlights Need for Improved Alert and Coordination Capabilities. May 18, 2000. Available at http://www.gao.gov/archive/2000/ai00181t.pdf. (Last accessed August 22, 2005)

[62] Wang , Yi-Min, Beck, Doug, Jiang, Xuxian, Roussev, Roussi. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities. Technical Report MSR-TR-2005-72. Cybersecurity and Systems Management Research Group, Microsoft Research, Redmond, Washington. June 4, 2005.

[63] Wang, Y., Roussev, R., Verbowski, C., Johnson, A., Ladd, A., AskStrider: What Has Changed on My Machine Lately? Microsoft Research Technical Report MSR-TR-2004-03, January 5, 2004.

[64] Wang, Y., Roussev, R., Verbowski, C., Johnson, A., Wu, M., Huang, Y., Kuo, S., Gatekeeper: Monitoring Auto-Start Extensibility Points (ASEPS) for Spyware Management, USENIX Association Proc XVIII LISA 2004.

[65] Web Seals of Approval – Final Report. Standing Committee of Officials of Consumer Affairs E-commerce Working Party. Department of Justice, Melbourne, Australia. January 2005.

[66] Web Seals: A Review of Online Privacy Programs. The Office of the Information and Privacy Commissioner, Ontario, Canada and Office of the Federal Privacy Commissioner, Sydney, Australia. 22nd International Conference on Privacy and Personal Data Protection, Venice, September 2000.

[67] WebSideStory data obtained at http://www.websidestory.com/products/web-analytics/datainsights/spotlight/05-10-2005.html (Last accessed August 19, 2005)

[68] What you can do about spyware and other unwanted software. May 1, 2005. Microsoft Corporation. Available at http://www.microsoft.com/athome/security/spyware/spywarewhat.mspx. (Last accessed September 3, 2005)

[69] What You Should Known About Download.Ject. Microsoft Corporation. Available at http://www.microsoft.com/security/incident/download_ject.mspx, June 2004. (Last accessed September 12, 2005)

[70] WIN95 Market Source Code. Rec.humor newsgroup. October 16, 1995. Available at http://groups.google.com/group/rec.humor/browse_thread/thread/b4ce62633b90473b/6aef24c0e13cd161?lnk=st&q=spyware&rnum=4&hl=en#6aef24c0e13cd161 (Last accessed September 3, 2005)

[71] Wotring, Brian. Osiris Host Integrity Monitoring v4.1.8 software. Osiris | Host Integrity Monitoring Website available at http://www.hostintegrity.com/osiris. (Last accessed July 11, 2005).

APPENDIX A – SCRIPTS

This appendix provides VBScript scripts written for the experiment. The scripts were used to drive the browser applications and collect data. Five scripts are provided; ie.vbs and firefox.vbs were used to drive browser applications to Internet web sites, miner.vbs called various third party tools to generate a system state snap shot, registry.vbs queried registry values and wrote baseline registry values to the test bed registry, and collector.vbs gathered browser specific information such as bookmarks and favorites.

A. IE.VBS

```
'**********************************************************************
'*
'*     File:         ie.vbs
'*     Created:      July 10, 2005
'*     Version:      2.75 Beta
'*     Author:             Mark Barwinski, Naval Postgraduate School
'*
'*     Parameters:   [Platform] such as IE, IESEC
'*                   [SECTOR] such as Military, Universities, etc
'*                   [SKIP] number of lines (urls) to skip from original
'*                   run.
'*
'*     Description:  Drives Internet Explorer by reading URLs from
'*                   c:\tools\urls and writing time stamps to a
'*                   s:\[Platform]\[Platofrm] URL Access Report.txt file.
'*
'*
'*
'*     Output:             Saves one text file for each URL visited
'*                   in the s:\PLATFORM folder. name consists of
'*                   registry + url.txt and is created by
'*                   c:\tools\registry.vbs
'*
'*     Assumptions:  1.- Browser running as Administrator
'*                   2.- Makes use of c:\tools\reg outside registry tool
'*                   3.- Location of Reset Files: c:\tools\baseline\
'*                   4.- Makes use of c:\tools\registry.vbs
'*                   5.- Makes use of c:\tools\urls
'*                   6.- Makes use of c:\tools\report.txt
'*
'*
'*
'*     NOTES:        Should change path to Reg tool
'*
'*
'**********************************************************************

Option Explicit
Dim fso, f, r, info, timeoutSeconds, elapsedSeconds, args, argNames, Platform,
Destination, URLReport, Sector
Dim objShell, oIE, ReadTextFileTest, URLCount, strFName, startTime, endTime,
sTime, mTime, hTime
Dim shl, wins, i, x, win, mainIE, IE, popup, colWindow, TestDate, FilePosition,
SkipLoop, SkipFLAG
```

```
Dim oDismiss, success
dim command2, objExec, IEState

URLCount = 1
Const ForReading = 1
Const ForWriting = 2
Const ForAppending = 8

'Wait Time between completion of site loading and test execution
Const RateSpeed = 5000
Dim AccessTime
TimeoutSeconds = 15
elapsedSeconds = 0

Set args = WScript.Arguments
Set argNames = WScript.Arguments.Unnamed

   if (args.Count <= 0) then
       Platform = ""
       WScript.Echo "Must enter a platform parameter" & _
       vbCRLF & "IE = Internet Explorer" & vbCRLF & "IESEC = Internet Explorer
Secure"
       WScript.Quit()
   else
       if (args.Count > 2) then
              Platform = args.Item(0)
              Sector = args.Item(1)
              FilePosition = args.Item(2)
              SkipFLAG = 1
       else
              Platform = args.Item(0)
              Sector = args.Item(1)
              FilePosition = 0
              SkipFLAG = 0
       end if
   end if


Destination = "s:\" & Platform & "\" & Sector & " URL Access Report.txt"

Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.OpenTextFile("c:\tools\urls.txt", ForReading)
if (fso.FileExists(Destination)) then
       Set URLReport = fso.OpenTextFile(Destination, ForAppending)
else
       Set URLReport = fso.CreateTextFile(Destination, False)
end if
Set objShell = CreateObject("WScript.Shell")
Set oDismiss = WScript.CreateObject("WScript.Shell")
success = oDismiss.AppActivate("IEXPLORE.EXE")



'Set oIE=CreateObject("InternetExplorer.Application", "IE_")
'This is how I trap and event such as IE_onQuit
Set oIE=WScript.CreateObject("InternetExplorer.Application", "IE_")
With oIE
       .navigate "about:blank"
       .visible=1
       .statusBar=1
       .toolbar=1
       .menubar=1
       .resizable=1

End With

startTime = Now()
TestDate = Date()
URLReport.WriteBlankLines(2)
URLReport.WriteLine TestDate & vbNewLine & Sector & ":Initiating Experiment at "
& startTime
```

```
URLReport.WriteLine ("-------------------------------------------------------------
-------------------")
URLReport.WriteLine "Count" & vbTab & "Access Time" & vbTab & vbTab & "Download
Status" & vbTab & "URL Visited"
WScript.Echo TestDate & vbNewLine & Sector & ":Initiating Experiment at " &
startTime
WScript.Echo ("-------------------------------------------------------------------
------------")
WScript.Echo "Count" & vbTab & "Access Time" & vbTab & vbTab & "Download Status"
& vbTab & "URL Visited"


'Need to add a not to exceed timeout so the page does not hang
'if it fails to load

Do While f.AtEndOfStream <> True

       if (SkipFLAG = 1) then

                     for x = 1 to FilePosition
                            f.SkipLine
                            URLCount = URLCount + 1
                     next
                            SkipFLAG = 0

       else

       ReadTextFileTest = f.Readline

       on error resume next
       oIE.navigate ReadTextFileTest
       Wscript.Sleep 5

       Do while oIE.ReadyState <> 4 and elapsedSeconds < timeoutSeconds
       Wscript.Sleep 1000
       elapsedSeconds = elapsedSeconds + 1
       Loop
       IEState = oIE.ReadyState
       if err.number <> 0 then
              oIE.Quit()
       end if

       AccessTime = Now()
       WScript.Sleep RateSpeed
       URLReport.WriteLine(URLCount & vbTab & AccessTime & vbTab & vbTab &
IEState & vbTab & ReadTextFileTest)
       WScript.echo URLCount & vbTab & AccessTime & vbTab & vbTab & IEState &
vbTab & ReadTextFileTest

       On Error Goto 0

       strFName = URLCount & Sector
       objShell.Run("cmd /c c:\tools\registry.vbs " & Platform & " " &
strFName), 7, True
       objShell.Run("cmd /c c:\tools\collector.vbs " & Platform & " " &
strFName), 7, True
       objShell.Run("cmd /c c:\tools\miner.vbs " & Platform & " " & strFName),
7, True

       elapsedSeconds = 0
       URLCount = URLCount + 1

       oDismiss.SendKeys "%n"
       WScript.Sleep 200
       oDismiss.SendKeys "{ESC}"
       WScript.Sleep 200
       oDismiss.SendKeys "%c"
       end if

Loop
endTime = Now()
WScript.Sleep 300
```

```
sTime = DateDiff("s", startTime, endTime) Mod 60
mTime = DateDiff("n", startTime, endTime) Mod 60
hTime = DateDiff("h", startTime, endTime)
URLReport.WriteLine(vbNewLine & "End Time: " & endTime & vbNewLine & "Total Run
Time (h:m:s): " & hTime & ":" & mTime & ":" & sTime)
WScript.echo (vbNewLine & "End Time: " & endTime & vbNewLine & "Total Run Time
(h:m:s): " & hTime & ":" & mTime & ":" & sTime)
f.close()
URLReport.close()

set shl = nothing
set oIE = nothing
set objShell = nothing

Sub IE_onQuit

URLReport.Close()
WScript.Sleep 2000
WScript.Echo "Executing counterpart Script"
command2 = "cmd /c ""c:\tools\ie.vbs"" " & Platform & " " & Sector & " " &
URLCount
WScript.Echo command2
'set objExec = objShell.Exec(command2)
objShell.Run(command2), 3, True

End Sub

Sub IE_NewWindow2(ByVal pDisp, ByVal URL)
       WScript.Echo "Killing New Window"
       URLReport.WriteLine "Killing New Window"
       pDisp.Quit()
end sub

Sub  IE_NavigateError(ByVal  pDisp,  ByVal  URL,  ByVal  TargetFrameName,  ByVal
StatusCode, ByRef Cancel)
       WScript.Echo "Navigation Error: " & StatusCode
       URLReport.WriteLine "Navigation Error: " & StatusCode
end sub
```

B. FIREFOX.VBS

```
'*********************************************************************
'*
'*     File:         firefox.vbs
'*     Created:      June 29, 2005
'*     Version:      2.2
'*     Author:             Mark Barwinski, Naval Postgraduate School
'*
'*     Parameters:   [Platform] FF, FFSEC
'*                   [SECTOR] such as Military, Universities, etc
'*                   [SKIP] number of lines (urls) to skip from original
'*                   run.
'*
'*     Description:  Drives Firefox by reading URLs from
'*                   c:\tools\urls and writing time stamps to a
'*                   s:\[Platform]\[Platofrm] URL Access Report.txt file.
'*                   It also invokes the following files:
'*                   c:\tools\registry.vbs
'*                   c:\tools\miner.vbs
'*                   c:\tools\collector.vbs
'*
'*
'*     Output:             Saves one text file for each URL visited
'*                   in the s:\[Platform] folder. name consists of
'*                   registry number + [Platform] + [type of file].txt
'*
'*
'*     Assumptions:  1.- Browser running as Administrator
'*                   2.- Makes use of c:\tools\reg outside registry tool
'*                   3.- Location of Reset Files: c:\tools\baseline\
'*                   4.- Makes use of c:\tools\registry.vbs
'*                   5.- Makes use of c:\tools\urls
'*                   6.-  Makes  use  of  s:\[Platform]\[Platofrm]  URL  Access
Report.txt file.
'*
'*
'*
'*     NOTES:
'*
'*
'*********************************************************************

Option Explicit
Dim fso, f, r, info, Sector, URLCount, strFName, args, argNames, x
Dim  Platform,  Destination,  URLReport,  startTime,  endTime,  url,  TestDate,
FilePosition, SkipLoop, SkipFLAG, sTime, mTime, hTime
Dim objShell, Firefox
'Wait Time between completion of site loading and test execution
Const RateSpeed = 5000
Const ForReading = 1
Const ForWriting = 2
Const ForAppending = 8
Dim AccessTime
URLCount = 1

Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.OpenTextFile("c:\Tools\urls.txt", ForReading)

Set args = WScript.Arguments
Set argNames = WScript.Arguments.Unnamed

   if (args.Count <= 0) then
       Platform = ""
       WScript.Echo "Must enter a platform parameter" & vbCRLF & "FF = Firefox"
& _
       vbCRLF & "FFSEC = Firefox Secure"
       WScript.Quit()
   else
       if (args.Count > 2) then
                   Platform = args.Item(0)
```

```
                    Sector = args.Item(1)
                    FilePosition = args.Item(2)
                    SkipFLAG = 1
             else
                    Platform = args.Item(0)
                    Sector = args.Item(1)
                    FilePosition = 0
                    SkipFLAG = 0
      end if
   end if

Destination = "s:\" & Platform & "\" & Sector & " Firefox URL Access Report.txt"
if (fso.FileExists(Destination)) then
      Set URLReport = fso.OpenTextFile(Destination, ForAppending)
else
      Set URLReport = fso.CreateTextFile(Destination, False)
end if

'Set objShell = WScript.CreateObject("WScript.Shell")
Set objShell = CreateObject("WScript.Shell")

url = "about:blank"
FireFox = objShell.Run("firefox.exe", 3)
objShell.AppActivate "Mozilla Firefox"

WScript.Sleep 5000
startTime = Now()
TestDate = Date()
URLReport.WriteLine TestDate & vbNewLine & Sector & ":Initiating Experiment at "
& startTime
URLReport.WriteLine ("------------------------------------------------------------
-------------------")
URLReport.WriteLine "Count" & vbTab & "Access Time" & vbTab & vbTab & "URL
Visited"
WScript.Echo ("------------------------------------------------------------------
------------")
WScript.Echo TestDate & vbNewLine & Sector & ":Initiating Experiment at " &
startTime
WScript.Echo ("------------------------------------------------------------------
------------")
WScript.Echo "Count" & vbTab & "Access Time" & vbTab & vbTab & "URL Visited"
WScript.Echo ("------------------------------------------------------------------
------------")

Do While f.AtEndOfStream <> True

      if (SkipFLAG = 1) then

            for x = 1 to FilePosition
                  f.SkipLine
                  URLCount = URLCount + 1

            next

                  SkipFLAG = 0

      else

      url = f.Readline

      objShell.SendKeys "%n"
      WScript.Sleep 500
      objShell.SendKeys "{ESC}"
      WScript.Sleep 500
      objShell.SendKeys "^l"
      WScript.Sleep 1500
      objShell.SendKeys "^a"
      objShell.SendKeys "{DEL}"
      WScript.Sleep 1000
      objShell.SendKeys url
      objShell.SendKeys "{ENTER}"
```

```
        AccessTime = Now()
        URLReport.WriteLine(URLCount & vbTab & AccessTime & vbTab & url)
        WScript.Echo URLCount & vbTab & AccessTime & vbTab & url

        WScript.Sleep RateSpeed
        strFName = URLCount & Sector
        objShell.Run("cmd  /c  c:\tools\registry.vbs  "  &  Platform  &  "  "  &
strFName), 7, True
        objShell.Run("cmd  /c  c:\tools\collector.vbs  "  &  Platform  &  "  "  &
strFName), 7, True
        objShell.Run("cmd /c c:\tools\miner.vbs " & Platform & " " & strFName),
7, True

        URLCount = URLCount + 1

        end if
Loop
f.Close()
endTime = Now()

sTime = DateDiff("s", startTime, endTime) Mod 60
mTime = DateDiff("n", startTime, endTime) Mod 60
hTime = DateDiff("h", startTime, endTime)
URLReport.WriteLine("End of Run: " & endTime)
URLReport.WriteLine(vbNewLine & "End Time: " & endTime & vbNewLine & "Total Run
Time (h:m:s): " & hTime & ":" & mTime & ":" & sTime)
WScript.Echo ("End of Run: " & endTime)
WScript.echo (vbNewLine & "End Time: " & endTime & vbNewLine & "Total Run Time
(h:m:s): " & hTime & ":" & mTime & ":" & sTime)
URLReport.Close()

set f = nothing


WScript.Quit()
```

C. MINER.VBS

```
'**********************************************************************
'*
'* File:           miner.vbs
'* Created:        July 4, 2005
'* Version:        2.0
'*
'* Main Function:  Pass two parameters and it executes the tool and .
'*                  directs output to a text file of your choice
'*                  toolFunction [Tool Name] [File Output Name]
'* by Mark Barwinski, Naval Postgraduate School
'*
'**********************************************************************

OPTION EXPLICIT
Const AutoRunsc = "C:\tools\autorunsc.exe -a -w -s -e -d -m -p"
Const Handles = "C:\tools\handle -u"
'Const Processes = "c:\tools\pslist.exe -d -m"
Const Processes = "tasklist /SVC /FO TABLE"
Const TCP = "c:\tools\tcpvcon.exe -a"
Const NETSTAT = "netstat -ano"
Const Service = "c:\tools\psservice.exe"
Dim File, args, argNames, Platform, CurrentURL


Set args = WScript.Arguments
Set argNames = WScript.Arguments.Unnamed
Platform = args.Item(0)
CurrentURL = args.Item(1)

WScript.echo CurrentURL

File = "s:\" & Platform & "\" & CurrentURL & "AutoRuns.txt"
call Tool(AutoRunsc, File)

File = "s:\" & Platform & "\" & CurrentURL & "Handles.txt"
call Tool(Handles, File)

File = "s:\" & Platform & "\" & CurrentURL & "ProcessList.txt"
call Tool(Processes, File)

File = "s:\" & Platform & "\" & CurrentURL & "TCP.txt"
call Tool(TCP, File)

File = "s:\" & Platform & "\" & CurrentURL & "NETSTAT.txt"
call Tool(NETSTAT, File)

File = "s:\" & Platform & "\" & CurrentURL & "Service.txt"
call Tool(Service, File)

Function Tool(strName, strOutputFile)
    Dim fso, MyFile, objShell, objScriptExec, strBuff
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set MyFile = fso.CreateTextFile(strOutputFile,1,True)
    Set objShell = CreateObject("WScript.Shell")
    Set objScriptExec = objShell.Exec(strName)
    strBuff = objScriptExec.StdOut.ReadAll
  '  WScript.Echo strIpConfig
MyFile.WriteLine(strBuff)
MyFile.Close
Set objScriptExec = Nothing
End Function
```

D. REGISTRY.VBS

```
'*********************************************************************
'*
'*     File:         Registry.vbs
'*     Created:      June 23, 2005
'*     Version:      1.3
'*     Author:             Mark Barwinski, Naval Postgraduate School
'*
'*     Parameters:   FF (Firefox), FFSEC (Firefox Secure)
'*                   IE (Internet Explorer), IESEC (IE Secure)
'*                   save creates a baseline of registry keys to be stored
'*                   in the c:\tools\baseline folder
'*
'*                   May take a second parameter = CurrentURL passed by
'*                   the driver program
'*
'*     Description:  Collects various registry keys associated with
'*                   autostart of programs and services. It also collects
'*                   keys associated with BHO in IE and other security
'*                   settings.  It also enumerates files in the startup
'*                   folders for both the Administrator and the All Users
'*                   folder. Once the data is collected, it deletes all
'*                   files in the Startup folders
'*
'*                   Registry keys are read from a text file located at
'*                   c:\thesis coding\tools\Keys.txt
'*                   This location changes to c:\tools\Keys.txt in the
'*                   testbench OS
'*
'*     LOGON
'*     HKCU\SOFTWARE\Policies\Microsoft\Windows\System\Scripts\Startup
'*     HKLM\SOFTWARE\Policies\Microsoft\Windows\System\Scripts\Startup
'*     HKCU\Software\Policies\Microsoft\Windows\System\Scripts\Logon
'*     HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Logon
'*     HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
'*     HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell

'*     HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell

'*     HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell

'*     HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell

'*     HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Taskman

'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
'*     HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load

'*     HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run

'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

'*     HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

'*     HKCU\Software\Microsoft\Windows\CurrentVersion\Run
'*     HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
'*
'*
'*     C:\Documents and Settings\All Users\Start Menu\Programs\Startup

'*     C:\Documents and Settings\Administrator\Start Menu\Programs\Startup

'*
'*
'*     SERVICES
'*     HKLM\System\CurrentControlSet\Services
'*     HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices\
'*     HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce\
```

```
'*     HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices\
'*     HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce\
'*
'*
'*     EXPLORER
'*     HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components
'*     HKCU\SOFTWARE\Microsoft\Active Setup\Installed Components
'*
       HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskSchedul
er
'*
       HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoa
d
'*
       HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoa
d
'*     HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks

'*     HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved

'*     HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
'*
'*     INTERNET EXPLORER
'*     HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser    Helper
Objects
'*     HKLM\Software\Microsoft\Internet Explorer\Toolbar
'*     HKLM\Software\Microsoft\Internet Explorer\Extensions
'*     HKLM\SOFTWARE\Microsoft\Internet Explorer\Search
'*     HKCU\SOFTWARE\Microsoft\Internet Explorer\Main
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\High
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Low
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Medium
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\MedLow
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\ZoneMap\Domains
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3
'*     HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4
'*
'*     APPINIT
'*     HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls

'*
'*     WINLOGON NOTIFICATIONS
'*     HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify

'*
'*     WINSOCK PROVIDERS
'*     HKLM\System\CurrentControlSet\Services\WinSock
'*
       HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Catal
og9
'*
'*     ADDITIONAL AUTORUN KEYS
'*     HKEY_CLASSES_ROOT\exefile\shell\open\command
'*     HKEY_CLASSES_ROOT\.exe
'*     HKLM\Software\Classes\exefile\shell\open\command
'*
'*     MODIFICATION OR DEFAULT FOLDER LOCATION
'*     HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User        Shell
Folders
'*     HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User        Shell
Folders
'*
'*     ADDITIONAL INSTALLATION VECTORS
'*     HKCR\PROTOCOLS\Name-Space Handler
```

```
'*     HKCU\Control Panel\Desktop\Wallpaper
'*
'*     CONSISTENCY OF PROGRAMS EXECUTING FILES
'*     HKEY_CLASSES_ROOT\exefile\shell\open\command
'*     HKEY_CLASSES_ROOT\regfile\shell\open\command
'*     HKEY_CLASSES_ROOT\txtfile\shell\open\command
'*     HKEY_CLASSES_ROOT\inifile\shell\open\command
'*     HKEY_CLASSES_ROOT\Scrfile\shell\open\command
'*     HKEY_CLASSES_ROOT\comfile\shell\open\command
'*     HKEY_CLASSES_ROOT\batfile\shell\open\command
'*     HKEY_CLASSES_ROOT\htafile\shell\open\command
'*     HKEY_CLASSES_ROOT\piffile\shell\open\command
'*
'*
'*                 Upon completion of the collection of these keys, keys are
'*                 RESET to the baseline condition by writing files from the
'*                 c:\tools\baseline\ folder
'*
'*
'*
'*
'*     Output:             Saves  one  file  with  all  the  values  for  the
abovereferenced
'*                 keys to a folder of choice.
'*
'*     Assumptions: 1.- Browser running as Administrator
'*                 2.- Makes use of c:\tools\reg outside registry tool
'*                 3.- Location of Reset Files: c:\tools\baseline\
'*                 4.- Keys must be located in Keys.txt file
'*
'*
'*     NOTES:      Should change path to Reg tool
'*
'*
'**********************************************************************



OPTION EXPLICIT

'////////////////////////////////////////////////////
Dim CurrentURL
'////////////////////////////////////////////////////

Dim Platform, args, oFS, Target, Destination, Errors, ErrorsFile, i, argNames,
tempLoc
Dim ErrorFile
Dim  MyFile,  objShell,  objScriptExec,  strShellOutput,  NewFileName,  objFolder,
objFile, colFiles
Dim KeyNames(85)
Dim RegFileName, command, fKeys, arrayIndex
Const ForReading = 1

'Require outside of loop
Const HKEY_Shell_100 = "HKCU\Control Panel\Desktop"
Const HKEY_Shell_100_Parameters = " /v Wallpaper"
Const         HKEY_Shell_101          =          "HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon"
Const HKEY_Shell_101_Parameters = " /v Userinit"

'////////////////////////////////////////////////////
'Initialize the Array with Keys from Keys.txt file
'////////////////////////////////////////////////////
arrayIndex = 0
Set oFS = CreateObject("Scripting.FileSystemObject")
Set fKeys = oFS.OpenTextFile("c:\Tools\Keys.txt", ForReading)
Do While fKeys.AtEndOfStream <> True
       KeyNames(arrayIndex) = fKeys.Readline
       arrayIndex = arrayIndex + 1
Loop
fKeys.close()
```

```
'////////////////////////////////////////////////////
'Check to see if any arguments were passed to the script
'arguments may be: FF, FFSEC, IE, IESEC depending on the
'platform. If first argument is 'localhost' then run
' a baseline
'////////////////////////////////////////////////////
Set args = WScript.Arguments
Set argNames = WScript.Arguments.Unnamed


if (args.Count <= 0) then
Platform = ""
WScript.Echo "Must enter a platform parameter" & vbCRLF & "FF = Firefox" & _
vbCRLF & "FFSEC = Firefox Secure" & vbCRLF & "IE = Internet Explorer" & _
vbCRLF & "IESEC = Internet Explorer Secure"
WScript.Quit()
else

       Platform = args.Item(0)

       if ((args.Item(0) = "IE") or (args.Item(0) = "IESEC") or (args.Item(0) =
"FF") or (args.Item(0) = "FFSEC")) then

       Destination = "s:\" & Platform & "\"
       ErrorsFile = "s:\" & Platform & "\errors.txt"
       CurrentURL = args.Item(1)
       RegistryCollection()


       else

       Select Case args.Item(0)
              Case "localhost"
              Destination = "s:\" & Platform & "\"
              ErrorsFile = "s:\" & Platform & "\errors.txt"
              CurrentURL = "BASELINE"
              RegistryCollection()

              Case "save"
              'WScript.Echo "save"
              tempLoc= args.Item(1)
              Destination = "s:\Baseline\" & tempLoc & "\"
              ErrorsFile = "s:\Baseline\" & tempLoc & "\errors.txt"
              SaveRegistry()

              Case "restore"
              'WScript.Echo "restore"
              tempLoc= args.Item(1)
              Destination = "s:\Baseline\" & tempLoc & "\"
              ErrorsFile = "s:\Baseline\" & tempLoc & "\errors.txt"
              RestoreRegistry()
              RestoreStartupFolders()
       End Select
       end if
end if
WScript.Quit()

'////////////////////////////////////////////////////////



'////////////////////////////////////////////////////
'RegistryCollection performs all data collection and also
'it calls reset functions at the end
'////////////////////////////////////////////////////
sub RegistryCollection()

Set oFS = CreateObject("Scripting.FileSystemObject")

NewFileName = Destination & CurrentURL & "REG.txt"
Set MyFile = oFS.CreateTextFile(NewFileName,1,True)
```

```
Set objShell = CreateObject("WScript.Shell")

For i = 0 to UBound(KeyNames)
	On Error Resume Next
	MyFile.WriteLine("HKEY_" & (i+1) & vbNewLine & "-------------------------
---------")
	Set objScriptExec = objShell.Exec("c:\tools\reg query " & chr(34) &
KeyNames(i) & chr(34) & " /s")
	strShellOutput = objScriptExec.StdOut.ReadAll

	if strShellOutput = "" then
	strShellOutput = "Nothing Entered in Key: " & KeyNames(i)
	end if
	MyFile.WriteLine(strShellOutput)
	MyFile.WriteLine("-----------------------------------")
	if (Err.Number > 0) then
		WriteError()
	end if
Next


On Error Resume Next
Set objScriptExec = objShell.Exec("c:\tools\reg query " & chr(34) &
HKEY_Shell_100 & chr(34) & HKey_Shell_100_Parameters)
strShellOutput = objScriptExec.StdOut.ReadAll
	if strShellOutput = "" then
	strShellOutput = "Nothing Entered in Key: " & HKEY_Shell_100 &
HKey_Shell_100_Parameters
	end if
	MyFile.WriteLine(strShellOutput)
	if (Err.Number > 0) then
		WriteError()
	end if

On Error Resume Next
Set objScriptExec = objShell.Exec("c:\tools\reg query " & chr(34) &
HKEY_Shell_101 & chr(34) & HKey_Shell_101_Parameters)
strShellOutput = objScriptExec.StdOut.ReadAll
	if strShellOutput = "" then
	strShellOutput = "Nothing Entered in Key: " & HKEY_Shell_101 &
HKey_Shell_101_Parameters
	end if
	MyFile.WriteLine(strShellOutput)
	if (Err.Number > 0) then
		WriteError()
	end if

MyFile.WriteLine(vbNewLine                                                    &
"*************************************************************************")
MyFile.NewLine()

'ENUMERATION OF ALL USERS STARTUP FOLDER
MyFile.WriteLine(vbNewLine & vbNewLine & "Enumeration of the Users Startup
Folder in")
MyFile.WriteLine("C:\Documents and Settings\All Users\Start
Menu\Programs\Startup" & _
	vbNewLine & "-----------------------------------")
MyFile.WriteLine("File Name" & vbTab & "File Size")
MyFile.WriteLine("-----------------------------------")
Set objFolder = oFS.GetFolder("C:\Documents and Settings\All Users\Start
Menu\Programs\Startup")
Set colFiles = objFolder.Files
For Each objFile in colFiles
	MyFile.WriteLine(objFile.Name & vbTab & objFile.Size)
Next

'ENUMERATION OF ADMINISTRATOR STARTUP FOLDER
MyFile.WriteLine(vbNewLine & vbNewLine & "Enumeration of the Administrator
Startup Folder in")
MyFile.WriteLine("C:\Documents and Settings\Administrator\Start
Menu\Programs\Startup" & _
	vbNewLine & "-----------------------------------")
```

```
MyFile.WriteLine("File Name" & vbTab & "File Size")
MyFile.WriteLine("----------------------------------")
Set objFolder = oFS.GetFolder("C:\Documents and Settings\Administrator\Start
Menu\Programs\Startup")
Set colFiles = objFolder.Files
For Each objFile in colFiles
      MyFile.WriteLine(objFile.Name & vbTab & objFile.Size)
Next

'ENUMERATION OF COOKIES FOLDER
MyFile.WriteLine(vbNewLine & vbNewLine & "Enumeration of the Cookies Folder in")
MyFile.WriteLine("C:\Documents and Settings\Administrator\Cookies" & _
      vbNewLine & "-----------------------------------------------")
MyFile.WriteLine("File Name" & vbTab & vbTab & vbTab & "File Size")
MyFile.WriteLine("-----------------------------------------------")
Set objFolder = oFS.GetFolder("C:\Documents and Settings\Administrator\Cookies")
Set colFiles = objFolder.Files
For Each objFile in colFiles
      MyFile.WriteLine(objFile.Name & vbTab & vbTab & vbTab &  objFile.Size)
Next

'ENUMERATION OF FAVORITES FOLDER
MyFile.WriteLine(vbNewLine & vbNewLine & "Enumeration of the Favorites Folder
in")
MyFile.WriteLine("C:\Documents and Settings\Administrator\Favorites" & _
      vbNewLine & "----------------------------------")
MyFile.WriteLine("File Name" & vbTab & "File Size")
MyFile.WriteLine("----------------------------------")
Set         objFolder         =         oFS.GetFolder("C:\Documents         and
Settings\Administrator\Favorites")
Set colFiles = objFolder.Files
For Each objFile in colFiles
      MyFile.WriteLine(objFile.Name & vbTab & objFile.Size)
Next



RestoreRegistry()
RestoreStartupFolders()
end sub


'////////////////////////////////////////////////////
'Writes any errors associated with query of a registry key to the errors.txt
file
'////////////////////////////////////////////////////
sub WriteError()
      Set Errors = oFS.OpenTextFile(ErrorsFile, 8, "True")
      WScript.Echo  Cstr(Err.Number)  &  vbCRLF  &  Err.Source  &  vbCRLF  &
Err.Description
                  Errors.WriteLine(Now() & vbTab & "Error: Could not query
Registry for " & Platform & " when accessing " & CurrentURL & vbCRLF _
                        & "Error Number: " & Cstr(Err.Number) & vbTab &
"Error Source: " & Err.Source & vbTab & "Error Description: " _
                        & Err.Description & vbNewLine)
      Err.Clear()
      Errors.Close()
end sub


'////////////////////////////////////////////////////
'Restores the registry settings back to the baseline
'////////////////////////////////////////////////////
sub RestoreRegistry()
Set objShell = CreateObject("WScript.Shell")
For i = 0 to UBound(KeyNames)
      command = "cmd /c c:\tools\reg restore " & chr(34) & KeyNames(i) &
chr(34) & " " & Destination & "RegFileName" & CStr((i+1)) & ".hiv"
      objShell.Run(command), 7, False
Next
```

```
objShell.Run("cmd /c c:\tools\reg restore " & chr(34) & HKEY_Shell_100 & chr(34)
& " " & Destination & "RegFileName" & "100" & ".hiv"), 7, False
objShell.Run("cmd /c c:\tools\reg restore " & chr(34) & HKEY_Shell_101 & chr(34)
& " " & Destination & "RegFileName" & "101" & ".hiv"), 7, False

DomainErrase()

end sub


'////////////////////////////////////////////////////
'Create Baseline by Saving Registry
'////////////////////////////////////////////////////
sub SaveRegistry()
Set objShell = CreateObject("WScript.Shell")
For i = 0 to UBound(KeyNames)
      command = "cmd /c c:\tools\reg save " & chr(34) & KeyNames(i) & chr(34) &
" " & Destination & "RegFileName" & CStr((i+1)) & ".hiv"
      objShell.Run(command), 7, False
Next
objShell.Run("cmd /c c:\tools\reg save " & chr(34) & HKEY_Shell_100 & chr(34) &
" " & Destination & "RegFileName" & "100" & ".hiv"), 7, False
objShell.Run("cmd /c c:\tools\reg save " & chr(34) & HKEY_Shell_101 & chr(34) &
" " & Destination & "RegFileName" & "101" & ".hiv"), 7, False

end sub


'////////////////////////////////////////////////////
'Restore all Startup folders by deleting all EXE and LNK files
'////////////////////////////////////////////////////
sub RestoreStartupFolders()
Set objShell = CreateObject("WScript.Shell")
command = "cmd /c del /Q " & chr(34) & "C:\Documents and Settings\All
Users\Start Menu\Programs\Startup\*.*" & chr(34)
objShell.Run(command), 7, False
command = "cmd /c del /Q " & chr(34) & "C:\Documents and
Settings\Administrator\Start Menu\Programs\Startup\*.*" & chr(34)
end sub


sub DomainErrase()
Const HKEY_LOCAL_MACHINE = &H80000001
dim strComputer, Subkey, strKeyPath, objReg, arrSubKeys
strComputer = "

Set objReg=GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & _
    strComputer & "\root\default:StdRegProv")

strKeyPath         =         "SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\ZoneMap\Domains"
objReg.EnumKey HKEY_LOCAL_MACHINE,strKeyPath, arrSubKeys
if (isArray(arrSubKeys) = -1) then
      for each Subkey in arrSubKeys
            objReg.DeleteKey  HKEY_LOCAL_MACHINE,strKeyPath  &  "\"  &  Subkey  &
"\www"
            objReg.DeleteKey HKEY_LOCAL_MACHINE,strKeyPath & "\" & Subkey
      next
end if

strKeyPath         =         "SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\P3P\History"
objReg.EnumKey HKEY_LOCAL_MACHINE,strKeyPath, arrSubKeys
if (isArray(arrSubKeys) = -1) then
      for each Subkey in arrSubKeys
            objReg.DeleteKey HKEY_LOCAL_MACHINE,strKeyPath & "\" & Subkey
      next
end if
end sub
```

E. COLLECTOR.VBS

```
'*********************************************************************
'*
'*     File:         Collector.vbs
'*     Created:      July 4, 2005
'*     Version:      2.1
'*     Author:             Mark Barwinski, Naval Postgraduate School
'*
'*     Parameters:   FF (Firefox), FFSEC (Firefox Secure)
'*                   IE (Internet Explorer), IESEC (Internet Explorer Secure)
'*
'*     Description:  Collects hosts file located at
'*                   c:\windows\system32\drivers\etc
'*
'*                   Collects Favorites for IE located at
'*                   c:\document and settings\administrator\favorites
'*
'*                   DISABLED
'*                   Collects Cookies for IE located at
'*                   c:\document and settings\administrator\cookies
'*
'*                   ** If Firefox then it collects the following:
'*                   Collects hosts file located at
'*                   c:\windows\system32\drivers\etc
'*
'*                   DISABLED
'*                   Collects Cookies for Firefox located at
'*                   c:\document and settings\administrator\application data\_
'*                         \mozilla\firefox\profiles\xxxxx.default\cookies.txt
'*
'*                   Collects Bookmarks
'*                   c:\document and settings\administrator\application data\_
'*
       \mozilla\firefox\profiles\xxxxx.default\bookmarks.html
'*
'*                   Collects user preferences
'*                   c:\document and settings\administrator\application data\_
'*                         \mozilla\firefox\profiles\xxxxx.default\prefs.js
'*
'*
'*                   Additionally, resets host file, cookies, and favorites for
IE
'*                   and Pref, hosts, bookmarks, and cookies files for IE
'*                   after each url visited
'*
'*
'*
'*
'*
'*     Output:             Saves files with the URL name appended to them for
'*                   identification
'*
'*     Assumptions:  Browser running as Administrator
'*
'*
'*********************************************************************


OPTION EXPLICIT
   Dim Platform, args, Target, Destination, Errors, ErrorsFile, i, argNames
   Dim  Hosts,  IECookies,  IECookies2,  FFCookies,  IEFavorites,  FFBookmarks,
FFPreferences
   Dim NewFileName, objShell, oFS
   Dim  colUserEnvVars,  BaselineHosts,  IEBaselineCookies,  IEBaselineFavorites,
FFBaselineCookies, FFBaselineBookmarks, FFBaselinePreferences
   Dim  objFolder,  objFSO,  colFiles,  objFile,  File1,  File2,  objScriptExec,
strShellOutput, FileReport, NegativeReport
   Dim profileNum
```

```
   Const ForReading = 1
   Const ForAppending = 8

   Set objShell = CreateObject("WScript.Shell")
   Set colUserEnvVars = objShell.Environment("Process")
   Set objFSO = CreateObject("Scripting.FileSystemObject")
   Set oFS = CreateObject("Scripting.FileSystemObject")

   On Error Resume Next
   profileNum = getProfileFolder()
   On Error Goto 0

   Hosts = "C:\Windows\System32\Drivers\Etc\hosts"
   BaselineHosts = "C:\tools\baseline\hosts"
   IEBaselineCookies = "c:\tools\baseline\Cookies\"
   IEBaselineFavorites = "c:\tools\baseline\Favorites"
   IECookies = colUserEnvVars("HOMEPATH") & "\Cookies\"
   IEFavorites = colUserEnvVars("HOMEPATH") & "\Favorites\"
   FFCookies       =       colUserEnvVars("HOMEPATH")       &       "\Application
Data\Mozilla\firefox\profiles\" & profileNum & "\cookies.txt"
   FFBookmarks      =      colUserEnvVars("HOMEPATH")      &      "\Application
Data\Mozilla\firefox\profiles\" & profileNum & "\bookmarks.html"
   FFPreferences      =      colUserEnvVars("HOMEPATH")      &      "\Application
Data\Mozilla\firefox\profiles\" & profileNum & "\prefs.js"
   FFBaselineCookies = "c:\tools\baseline\Cookies.txt"
   FFBaselineBookmarks = "c:\tools\baseline\Bookmarks.html"
   FFBaselinePreferences = "c:\tools\baseline\prefs.js"


   '////////////////////////////////////////////////////
   Dim CurrentURL
   '////////////////////////////////////////////////////



   '////////////////////////////////////////////////////
   'Check to see if any arguments were passed to the script
   'arguments may be: FF, FFSEC, IE, IESEC depending on the
   'platform
   '////////////////////////////////////////////////////

Set args = WScript.Arguments
   Set argNames = WScript.Arguments.Unnamed

   if (args.Count <= 0) then
       Platform = ""
       WScript.Echo "Must enter a platform parameter" & vbCRLF & "FF = Firefox"
& _
       vbCRLF & "FFSEC = Firefox Secure" & vbCRLF & "IE = Internet Explorer" & _
       vbCRLF & "IESEC = Internet Explorer Secure"
       WScript.Quit()
   else
       Platform = args.Item(0)
       CurrentURL = args.Item(1)
       if ((args.Item(0) = "IE") or (args.Item(0) = "IESEC")) then
              CollectIE()
       else
              if ((args.Item(0) = "FF") or (args.Item(0) = "FFSEC")) then
                     CollectFirefox()
              end if
       end if
   end if

   '////////////////////////////////////////////////////
   'CollectIE
   '////////////////////////////////////////////////////
   sub CollectIE()

   Destination = "s:\" & Platform & "\"
   ErrorsFile = "s:\" & Platform & "\errors.txt"
   NegativeReport = Destination & CurrentURL & "Negative DIFF Report.txt"
   '////////////////////////////////////////////////////
```

```
   DIFF Hosts, BaselineHosts

   ResetIE_Files()
   end sub
   '////////////////////////////////////////////////////

   '////////////////////////////////////////////////////
   'CollectFirefox
   '////////////////////////////////////////////////////
   sub CollectFirefox()

   Destination = "s:\" & Platform & "\"
   ErrorsFile = "s:\" & Platform & "\errors.txt"
   NegativeReport = Destination & CurrentURL & "Negative DIFF Report.txt"
   '////////////////////////////////////////////////////
   DIFF Hosts, BaselineHosts
   DIFF FFBookmarks, FFBaselineBookmarks
   DIFF FFPreferences, FFBaselinePreferences
   ResetFF_Files()

   end sub
   '////////////////////////////////////////////////////

   '////////////////////////////////////////////////////
   'Get Profile Folder Name
   'Returns the name of any default profile for Firefox
   '////////////////////////////////////////////////////
   function getProfileFolder()
       dim   inputstring,   re,   demofolder,   subfol,   folcoll,   pFolderName,
FirefoxProfile
       dim objMatch, folderPath
       Set re = new regexp

       re.Pattern = "\w{8}\.default"
       re.IgnoreCase = true
       re.Global = true
       folderPath     =     colUserEnvVars("HOMEPATH")     &     "\Application
Data\Mozilla\Firefox\Profiles"
       Set demofolder = objFSO.getFolder(folderPath)
       Set folcoll = demofolder.SubFolders
       for each subfol in folcoll
              inputstring = inputstring & " " & subfol.Name

       next
       Set pFolderName = re.Execute(inputstring)

       for each objMatch in pFolderName
              FirefoxProfile = objMatch.Value
       Next

       getProfileFolder = FirefoxProfile
   end function


'////////////////////////////////////////////////////
'Reset IE Files
'////////////////////////////////////////////////////
sub ResetIE_Files()

  Set Target = objFSO.GetFile("C:\tools\baseline\hosts")
  Target.Copy("C:\Windows\System32\Drivers\Etc\hosts")


  Set objFolder = objFSO.GetFolder(IECookies)
  Set colFiles = objFolder.Files
  For Each objFile in colFiles
       if objFile.Type = "Text Document" then
              objFSO.DeleteFile(objFile.Path), True
       end if
  Next

  objFSO.DeleteFile(IEFavorites & "\*.*"), true
```

```
  objFSO.DeleteFolder(IEFavorites & "\*"), true

end sub


'////////////////////////////////////////////////////
'Reset FF Files
'////////////////////////////////////////////////////
sub ResetFF_Files()

Set Target = objFSO.GetFile("C:\tools\baseline\hosts")
Target.Copy("C:\Windows\System32\Drivers\Etc\hosts")

objFSO.CopyFile     "c:\tools\baseline\Bookmarks.html",     "c:\Documents     And
settings\Administrator\Application Data\Mozilla\firefox\profiles\" & profileNum
& "\bookmarks.html"
objFSO.CopyFile      "c:\tools\baseline\cookies.txt",      "c:\Documents      And
settings\Administrator\Application Data\Mozilla\firefox\profiles\" & profileNum
& "\cookies.txt"
objFSO.CopyFile       "c:\tools\baseline\prefs.js",       "c:\Documents       And
settings\Administrator\Application Data\Mozilla\firefox\profiles\" & profileNum
& "\prefs.js"

end sub

Sub DIFF(File1, File2)
       File1 = Chr(34) & File1 & Chr(34)
       File2 = Chr(34) & File2 & Chr(34)
       Set objScriptExec = objShell.Exec("c:\tools\diff -s " & File1 & " " &
File2)
       strShellOutput = objScriptExec.StdOut.ReadAll
       If Len(strShellOutput) = 0 then

              if (oFS.FileExists(NegativeReport)) then
                     Set     FileReport     =     oFS.OpenTextFile(NegativeReport,
ForAppending)
              else
                     Set FileReport = oFS.CreateTextFile(NegativeReport, False)
              end if

              FileReport.WriteLine("Unchanged : " & File2)
       else
              if InStr(strShellOutput, "identical") > 0 then
                     if (oFS.FileExists(NegativeReport)) then
                            Set   FileReport   =   oFS.OpenTextFile(NegativeReport,
ForAppending)
                     else
                            Set  FileReport  =  oFS.CreateTextFile(NegativeReport,
False)
                     end if
              FileReport.WriteLine("Unchanged : " & File2)
              else
                     NewFileName = Destination & CurrentURL & "Browser_DATA.txt"
                     if (oFS.FileExists(NewFileName )) then
                            Set   FileReport   =   oFS.OpenTextFile(NewFileName   ,
ForAppending)
                     else
                            Set  FileReport  =  oFS.CreateTextFile(NewFileName  ,
False)
                     end if
                     FileReport.WriteLine(strShellOutput)

              end if
       end if
       FileReport.Close()
end Sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B – REGISTRY KEYS

This appendix provides a list of the registry keys queried by script registry.vbs. These registry keys were read after visiting each URL in a sector.

```
HKCU\SOFTWARE\Policies\Microsoft\Windows\System\Scripts\Startup
HKLM\SOFTWARE\Policies\Microsoft\Windows\System\Scripts\Startup
HKCU\Software\Policies\Microsoft\Windows\System\Scripts\Logon
HKLM\Software\Policies\Microsoft\Windows\System\Scripts\Logon
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell
HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Shell
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Taskman
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components
HKCU\SOFTWARE\Microsoft\Active Setup\Installed Components
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects
HKCU\SOFTWARE\Microsoft\Internet Explorer\Main
HKLM\SOFTWARE\Microsoft\Internet Explorer\Main
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\P3P\History
HKLM\Software\Microsoft\Internet Explorer\Extensions
HKLM\SOFTWARE\Microsoft\Internet Explorer\Search
HKLM\SOFTWARE\Microsoft\Internet Explorer\SearchUrl
HKLM\Software\Microsoft\Internet Explorer\Toolbar
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\High
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Low
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Medium
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\MedLow
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\Domains
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Appinit_Dlls
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify
HKLM\System\CurrentControlSet\Services\WinSock
HKLM\System\CurrentControlSet\Services\WinSock2\Parameters\Protocol_Catalog9
HKEY_CLASSES_ROOT\exefile\shell\open\command
```

```
HKEY_CLASSES_ROOT\.exe
HKLM\Software\Classes\exefile\shell\open\command
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders
HKCR\PROTOCOLS\Name-Space Handler
HKEY_CLASSES_ROOT\exefile\shell\open\command
HKEY_CLASSES_ROOT\regfile\shell\open\command
HKEY_CLASSES_ROOT\txtfile\shell\open\command
HKEY_CLASSES_ROOT\inifile\shell\open\command
HKEY_CLASSES_ROOT\Scrfile\shell\open\command
HKEY_CLASSES_ROOT\comfile\shell\open\command
HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\htafile\shell\open\command
HKEY_CLASSES_ROOT\piffile\shell\open\command
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\P3P\History
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\ZonesMap\Domains
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\High
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Low
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\Medium
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\TemplatePolicies\MedLow
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\Internet Explorer\Main
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\Internet Explorer\Toolbar
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\Internet Explorer\Extensions
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\Internet Explorer\Search
HKEY_USERS\S-1-5-21-329068152-1004336348-839522115-
1003\Software\Microsoft\Windows\Internet Explorer\SearchUrl
```

APPENDIX C – OSIRIS REPORTS

This appendix provides reports generated by Osiris Host Integrity System. Reports for the three most infected test beds are provided – bank, real estate, and online travel-related web sites. The reports show an entry code, test bed name (e.g., [IE]), a [cmp] or [new] for the type of event, the full path of the changed or new file, and the date and time associated with the file.

A. BANKING SECTOR – IE

compare time: Mon Jul 11 10:04:20 2005
host: IE
scan config: ()
log file: 4
base database: 2
compare database: 3

[211][IE][cmp][c:\windows\system32\drivers\PROCEXP.SYS][mtime][Wed Jul 06 15:29:20 2005][Sun Jul 10 21:18:40 2005]

[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\pohrazvr.exe]

[223][IE][cmp][mod_ports][TCP:0.0.0.0:1025][TCP:1025;exe=C:\WINDOWS\System32\svchost.exe;pid=964;local=0.0.0.0;remote=0.0.0.0][TCP:1025;exe=C:\WINDOWS\System32\svchost.exe;pid=984;local=0.0.0.0;remote=0.0.0.0]

[223][IE][cmp][mod_ports][TCP:0.0.0.0:135][TCP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=872;local=0.0.0.0;remote=0.0.0.0][TCP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=892;local=0.0.0.0;remote=0.0.0.0]

[223][IE][cmp][mod_ports][TCP:0.0.0.0:2265][TCP:2265;exe=C:\WINDOWS\System32\osirisd.exe;pid=1408;local=0.0.0.0;remote=0.0.0.0][TCP:2265;exe=C:\WINDOWS\System32\osirisd.exe;pid=1412;local=0.0.0.0;remote=0.0.0.0]

[223][IE][cmp][mod_ports][TCP:0.0.0.0:5000][TCP:5000;exe=C:\WINDOWS\System32\svchost.exe;pid=1172;local=0.0.0.0;remote=0.0.0.0][TCP:5000;exe=C:\WINDOWS\System32\svchost.exe;pid=1144;local=0.0.0.0;remote=0.0.0.0]

[223][IE][cmp][mod_ports][UDP:1026][UDP:1026;exe=C:\WINDOWS\System32\svchost.exe;pid=1128][UDP:1026;exe=C:\WINDOWS\System32\svchost.exe;pid=1088]

[223][IE][cmp][mod_ports][UDP:1027][UDP:1027;exe=C:\WINDOWS\System32\svchost.exe;pid=964][UDP:1027;exe=C:\WINDOWS\System32\svchost.exe;pid=984]

[223][IE][cmp][mod_ports][UDP:123][UDP:123;exe=C:\WINDOWS\System32\svchost.exe;pid=964][UDP:123;exe=C:\WINDOWS\System32\svchost.exe;pid=984]

[223][IE][cmp][mod_ports][UDP:135][UDP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=872][UDP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=892]

[223][IE][cmp][mod_ports][UDP:1900][UDP:1900;exe=C:\WINDOWS\System32\svchost.exe;pid=1172][UDP:1900;exe=C:\WINDOWS\System32\svchost.exe;pid=1144]

[223][IE][cmp][mod_ports][UDP:500][UDP:500;exe=C:\WINDOWS\system32\lsass.exe;pid=696][UDP:500;exe=C:\WINDOWS\system32\lsass.exe;pid=680]

[223][IE][cmp][mod_kmods][service:RasAuto][service:RasAuto;dname:Remote Access Auto Connection Manager;status:stopped][service:RasAuto;dname:Remote Access Auto Connection Manager;status:running]

[223][IE][cmp][mod_kmods][service:RasMan][service:RasMan;dname:Remote Access Connection Manager;status:stopped][service:RasMan;dname:Remote Access Connection Manager;status:running]

[223][IE][cmp][mod_kmods][service:TapiSrv][service:TapiSrv;dname:Telephony;status:stopped][service:TapiSrv;dname:Telephony;status:running]

[221][IE][new][mod_ports][TCP:0.0.0.0:1041][TCP:1041;exe=SYSTEM;pid=4;local=0.0.0.0;remote=0.0.0.0]

Change Statistics:
---
checksums: 1
SUID files: 0
root-owned files: 3
file permissions: 0
new: 18
missing: 2
total differences: 45

B. REAL ESTATE SECTOR - IE

compare time: Sun Jul 24 16:33:50 2005
host: IE
scan config: ()
log file: 7
base database: 5
compare database: 6
[202][IE][missing][C:\Documents and Settings\Administrator\Favorites\Desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\cfout.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\djtopr1150.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\jkill.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\umqltg4cl_.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\0TUZODQB\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\2LNGD472\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\2LNGD472\install\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\2LNGD472\somegirls\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\2LNGD472\www.hotels\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\3V1F7XWK\WebRebates\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\3V1F7XWK\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\3V1F7XWK\thumb\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\4P6BKXUR\PRScript\1331\135.dll]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\4P6BKXUR\www.cheaptickets\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AB2VELEN\Fab\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AB2VELEN\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AB2VELEN\intonantion\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AB2VELEN\js_code\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AB2VELEN\nem220\1331\135.dll]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\GHM3WTYV\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\K1MRK9QR\wsem303\1331\135.dll]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\K33NMO55\actalert\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\K33NMO55\desktop.ini]

[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\ODM7CXIB\blah\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\ODM7CXIB\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\OHQBC9AV\ads\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\OHQBC9AV\cheaptickets\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\OHQBC9AV\www.cheaptickets\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\OPYNS1MR\desktop.ini]
[203][IE][new][C:\Program Files\180searchassistant\salm.exe]
[203][IE][new][C:\Program Files\180searchassistant\salmhook.dll]
[203][IE][new][C:\Program Files\Internet Optimizer\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\install.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\optimize.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\install.exe]
[203][IE][new][C:\Program Files\Media Gateway\Info.txt]
[203][IE][new][C:\Program Files\Media Gateway\MediaGateway.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\38ts9dio.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\40754594.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\66708108.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\9c172bf2.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\9sva9xma.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\Uninstall.EXE]
[203][IE][new][C:\Program Files\ProSiteFinder\gx3x3ujv.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinder.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinderh.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\wbk6f8lb.DLL]
[203][IE][new][C:\Program Files\Web_Rebates\README.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\log.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_u.htm]

[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates0.exe]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates1.exe]
[203][IE][new][C:\Program Files\Web_Rebates\disp1150.exe]
[203][IE][new][c:\windows\7if8uge6.exe]
[203][IE][new][c:\windows\Downloaded Program Files\ClientAX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\MediaGatewayX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\clientax.inf]
[203][IE][new][c:\windows\fwnkjsb.exe]
[203][IE][new][c:\windows\nem220.dll]
[203][IE][new][c:\windows\system32\atl71.dll]
[203][IE][new][c:\windows\system32\bn9v8uqv.exe]
[203][IE][new][c:\windows\system32\drivers\PROCEXP.SYS]
[203][IE][new][c:\windows\system32\fka3ngn0.dll]
[203][IE][new][c:\windows\system32\irkuap4b.exe]
[203][IE][new][c:\windows\wsem303.dll]

Change Statistics:
---
checksums: 0
SUID files: 0
root-owned files: 0
file permissions: 0
new: 96
missing: 1
total differences: 97

C. ONLINE TRAVEL SECTOR – IE

compare time: Sun Jul 24 17:28:33 2005
host: IE
scan config: ()
log file: 7
base database: 5
compare database: 6
[202][IE][missing][C:\Documents and Settings\Administrator\Favorites\Desktop.ini]
[211][IE][cmp][C:\Documents and Settings\Administrator\Local Settings\desktop.ini][mtime][Tue Jul 19 21:02:50 2005][Tue Jul 19 20:52:05 2005]
[211][IE][cmp][C:\Documents and Settings\Administrator\ntuser.ini][mtime][Tue Jul 19 20:54:44 2005][Tue Jul 19 20:50:10 2005]
[211][IE][cmp][C:\Documents and Settings\LocalService\Local Settings\desktop.ini][mtime][Tue Jul 19 21:01:05 2005][Tue Jul 19 20:51:47 2005]
[211][IE][cmp][C:\Documents and Settings\NetworkService\Local Settings\desktop.ini][mtime][Tue Jul 19 21:01:05 2005][Tue Jul 19 20:51:47 2005]
[211][IE][cmp][C:\Program Files\VMware\VMware Tools\tools.conf][mtime][Tue Jul 19 21:02:53 2005][Tue Jul 19 20:52:11 2005]
[211][IE][cmp][C:\Tools][mtime][Tue Jul 19 20:23:11 2005][Tue Jul 19 20:57:59 2005]
[202][IE][missing][C:\Tools\OnlineTravel.txt]
[204][IE][cmp][C:\Tools\urls.txt][checksum][15348a26d047966e277bc13546b82c1b][12046cd73549257e3f61d834175ba5ff]
[215][IE][cmp][C:\Tools\urls.txt][bytes][11200][10635]
[211][IE][cmp][C:\Tools\urls.txt][mtime][Tue Jul 19 20:16:36 2005][Tue Jul 19 20:18:17 2005]
[203][IE][new][C:\Documents and Settings\Administrator\Application Data\Macromedia\Flash Player\macromedia.com\support\flashplayer\sys\#i.cmpnet.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\cfout.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\djtopr1150.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\jkill.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temp\umqltg4cl_.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\89Q7ST27\a\1331\135.com]

[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\A956J6TO\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\A956J6TO\fl_26063_eng\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\A956J6TO\install\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\A956J6TO\sidecar_national\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\AP7418R2\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\H0CVLTWX\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\JBLFJPGS\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\K5UZ4LE3\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\K5UZ4LE3\wsem303\1331\135.dll]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\O7ZRY05P\a\1331\135.com]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\O7ZRY05P\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\PZV7XPOA\WebRebates\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\PZV7XPOA\desktop.ini]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\PZV7XPOA\nem220\1331\135.dll]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\PZV7XPOA\rorbutton\1331\135.txt]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\S3LZYYV1\actalert\1331\135.exe]
[203][IE][new][C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\S3LZYYV1\desktop.ini]
[203][IE][new][C:\Program Files\180searchassistant\salm.exe]
[203][IE][new][C:\Program Files\180searchassistant\salmhook.dll]
[203][IE][new][C:\Program Files\Internet Optimizer\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\install.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\optimize.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\actalert.exe]
[203][IE][new][C:\Program Files\Internet Optimizer\update\install.exe]
[203][IE][new][C:\Program Files\Media Gateway\Info.txt]
[203][IE][new][C:\Program Files\Media Gateway\MediaGateway.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\40754594.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\5b6fayfg.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\66708108.txt]
[203][IE][new][C:\Program Files\ProSiteFinder\8wfvt0n5.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.dll]
[203][IE][new][C:\Program Files\ProSiteFinder\ProSiteFinder1\prositefinder1.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\Uninstall.EXE]
[203][IE][new][C:\Program Files\ProSiteFinder\fsn9m6kq.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\ldhznpoc.DLL]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinder.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\prositefinderh.exe]
[203][IE][new][C:\Program Files\ProSiteFinder\saqegvlp.DLL]
[203][IE][new][C:\Program Files\Web_Rebates\README.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ub.htm]

[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Html\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\log.txt]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\popo1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\pref1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\scri1150a.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_r.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rb.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_rbh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_u.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ub.htm]
[203][IE][new][C:\Program Files\Web_Rebates\Sy1150\Tp1150\spec1150a_ubh.htm]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates0.exe]
[203][IE][new][C:\Program Files\Web_Rebates\WebRebates1.exe]
[203][IE][new][C:\Program Files\Web_Rebates\disp1150.exe]
[203][IE][new][C:\Tools\RealEstate.txt]
[203][IE][new][c:\windows\Downloaded Program Files\ClientAX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\MediaGatewayX.dll]
[203][IE][new][c:\windows\Downloaded Program Files\clientax.inf]
[203][IE][new][c:\windows\ci05p91g.exe]
[203][IE][new][c:\windows\gnwdsrut.exe]
[203][IE][new][c:\windows\nem220.dll]
[203][IE][new][c:\windows\system32\6v4mf229.dll]
[203][IE][new][c:\windows\system32\atl71.dll]
[203][IE][new][c:\windows\system32\drivers\PROCEXP.SYS]
[203][IE][new][c:\windows\system32\fqnkc7um.exe]
[203][IE][new][c:\windows\system32\qfmrgpis.exe]
[203][IE][new][c:\windows\wsem303.dll]
[223][IE][cmp][mod_ports][TCP:0.0.0.0:1025][TCP:1025;exe=C:\WINDOWS\System32\svchost.exe;pid=952;local=0.0.0.0;remote=0.0.0.0][TCP:1025;exe=C:\WINDOWS\System32\svchost.exe;pid=956;local=0.0.0.0;remote=0.0.0.0]
[223][IE][cmp][mod_ports][TCP:0.0.0.0:135][TCP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=860;local=0.0.0.0;remote=0.0.0.0][TCP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=864;local=0.0.0.0;remote=0.0.0.0]
[223][IE][cmp][mod_ports][TCP:0.0.0.0:2265][TCP:2265;exe=C:\WINDOWS\System32\osirisd.exe;pid=1372;local=0.0.0.0;remote=0.0.0.0][TCP:2265;exe=C:\WINDOWS\System32\osirisd.exe;pid=1384;local=0.0.0.0;remote=0.0.0.0]
[223][IE][cmp][mod_ports][TCP:0.0.0.0:5000][TCP:5000;exe=C:\WINDOWS\System32\svchost.exe;pid=1100;local=0.0.0.0;remote=0.0.0.0][TCP:5000;exe=C:\WINDOWS\System32\svchost.exe;pid=1156;local=0.0.0.0;remote=0.0.0.0]
[223][IE][cmp][mod_ports][UDP:1026][UDP:1026;exe=C:\WINDOWS\System32\svchost.exe;pid=1040][UDP:1026;exe=C:\WINDOWS\System32\svchost.exe;pid=1112]
[223][IE][cmp][mod_ports][UDP:1027][UDP:1027;exe=C:\WINDOWS\System32\svchost.exe;pid=952][UDP:1027;exe=C:\WINDOWS\System32\svchost.exe;pid=956]
[223][IE][cmp][mod_ports][UDP:123][UDP:123;exe=C:\WINDOWS\System32\svchost.exe;pid=952][UDP:123;exe=C:\WINDOWS\System32\svchost.exe;pid=956]
[223][IE][cmp][mod_ports][UDP:135][UDP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=860][UDP:135;exe=C:\WINDOWS\system32\svchost.exe;pid=864]
[223][IE][cmp][mod_ports][UDP:1900][UDP:1900;exe=C:\WINDOWS\System32\svchost.exe;pid=1100][UDP:1900;exe=C:\WINDOWS\System32\svchost.exe;pid=1156]

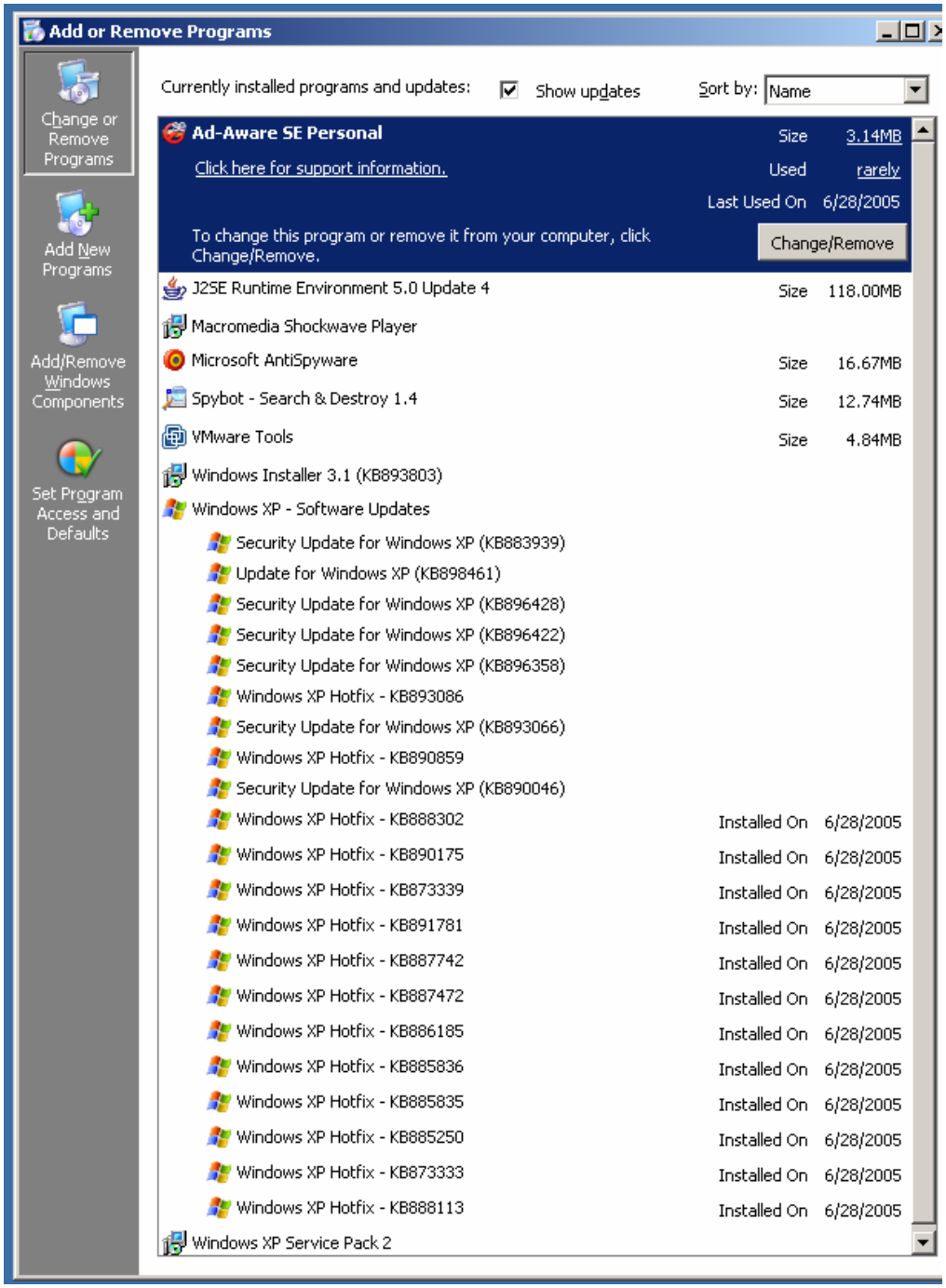
[223][IE][cmp][mod_ports][UDP:500][UDP:500;exe=C:\WINDOWS\system32\lsass.exe;pid=648][UDP:500
;exe=C:\WINDOWS\system32\lsass.exe;pid=700]
[221][IE][new][mod_ports][TCP:0.0.0.0:2400][TCP:2400;exe=C:\Program
Files\Web_Rebates\WebRebates0.exe;pid=5556;local=0.0.0.0;remote=0.0.0.0]
[221][IE][new][mod_ports][UDP:1046][UDP:1046;exe=C:\WINDOWS\System32\svchost.exe;pid=956]
[221][IE][new][mod_ports][UDP:1067][UDP:1067;exe=C:\WINDOWS\System32\svchost.exe;pid=1112]
[221][IE][new][mod_ports][UDP:1145][UDP:1145;exe=C:\Program
Files\180searchassistant\salm.exe;pid=1976]
[221][IE][new][mod_ports][UDP:1265][UDP:1265;exe=C:\WINDOWS\System32\svchost.exe;pid=1112]
[221][IE][new][mod_ports][UDP:1404][UDP:1404;exe=C:\WINDOWS\System32\svchost.exe;pid=1112]

Change Statistics:
---
checksums: 1
SUID files: 0
root-owned files: 2
file permissions: 0
new: 97
missing: 2
total differences: 118

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D – PATCH LEVEL

This appendix provides a list of the software and patches installed in the IESEC test bed.

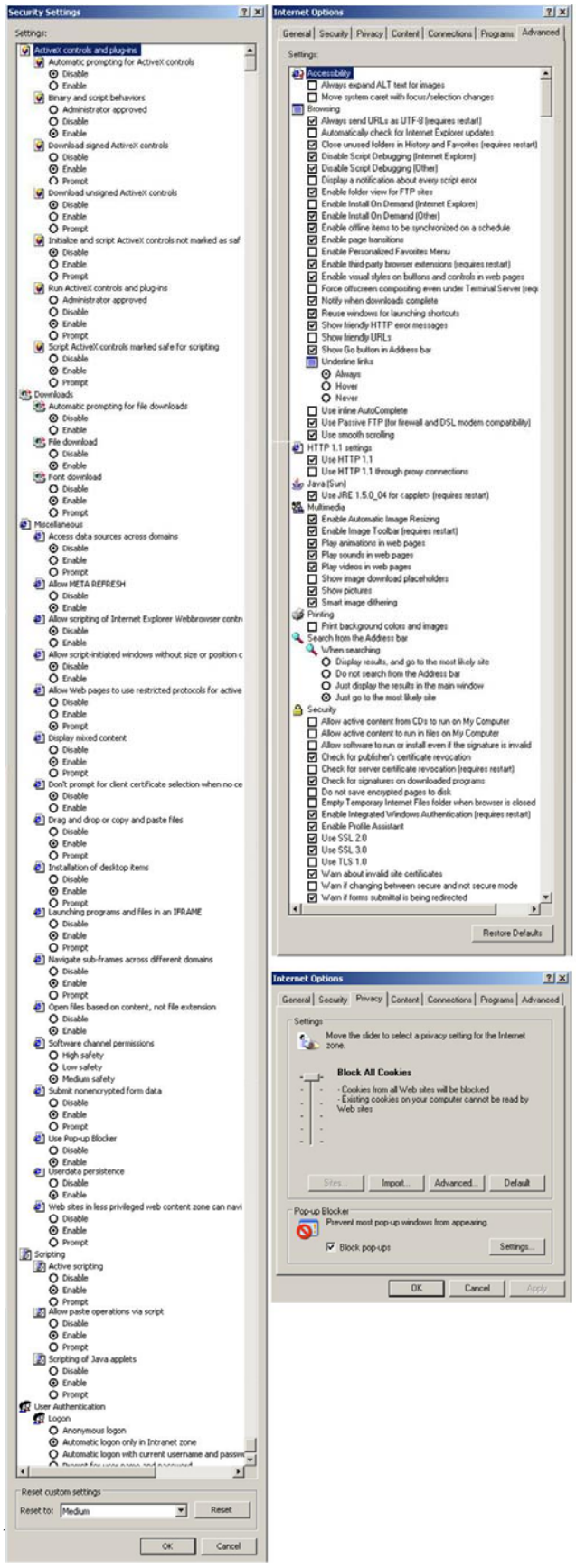


THIS PAGE INTENTIONALLY LEFT BLANK

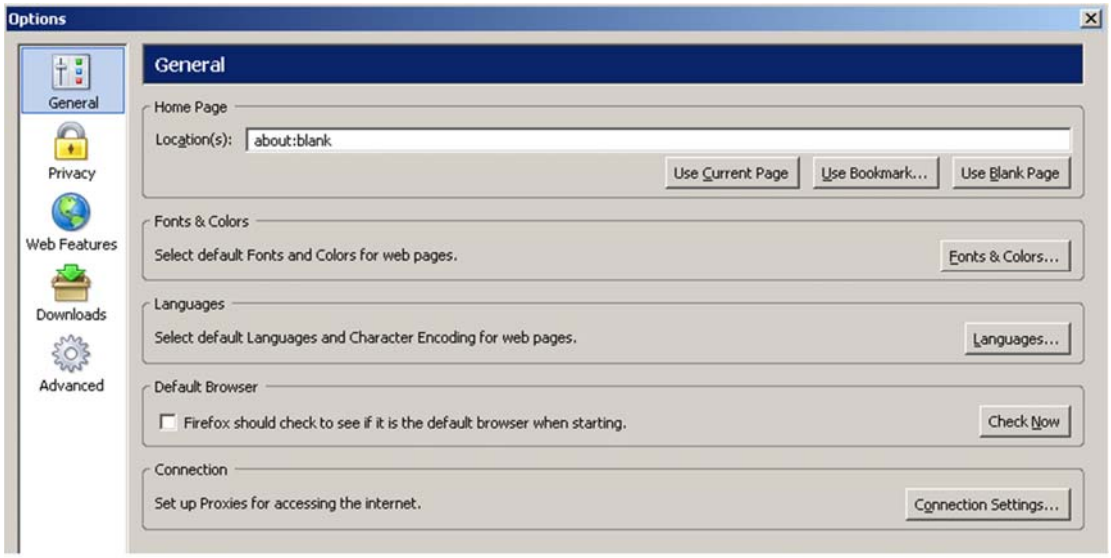
APPENDIX E – BROWSER SETTINGS

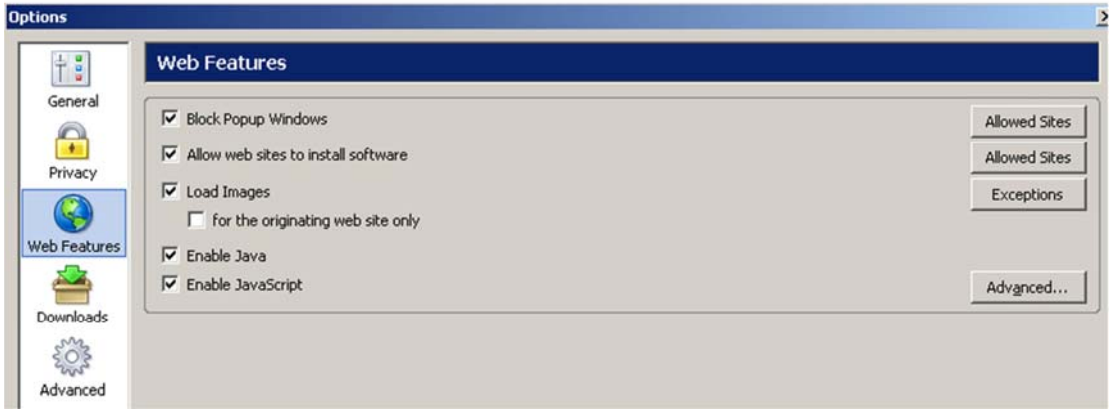
This appendix shows pertinent browser settings for the IESEC and FFSEC test beds. For Microsoft's Internet Explorer, the Security, Advanced, and Privacy tabs are shown. For Mozilla's Firefox, General, Web Features, and Advanced tabs are shown. Additionally, the content of the Prefs.js file is provided. This file contains Firefox user preferences.

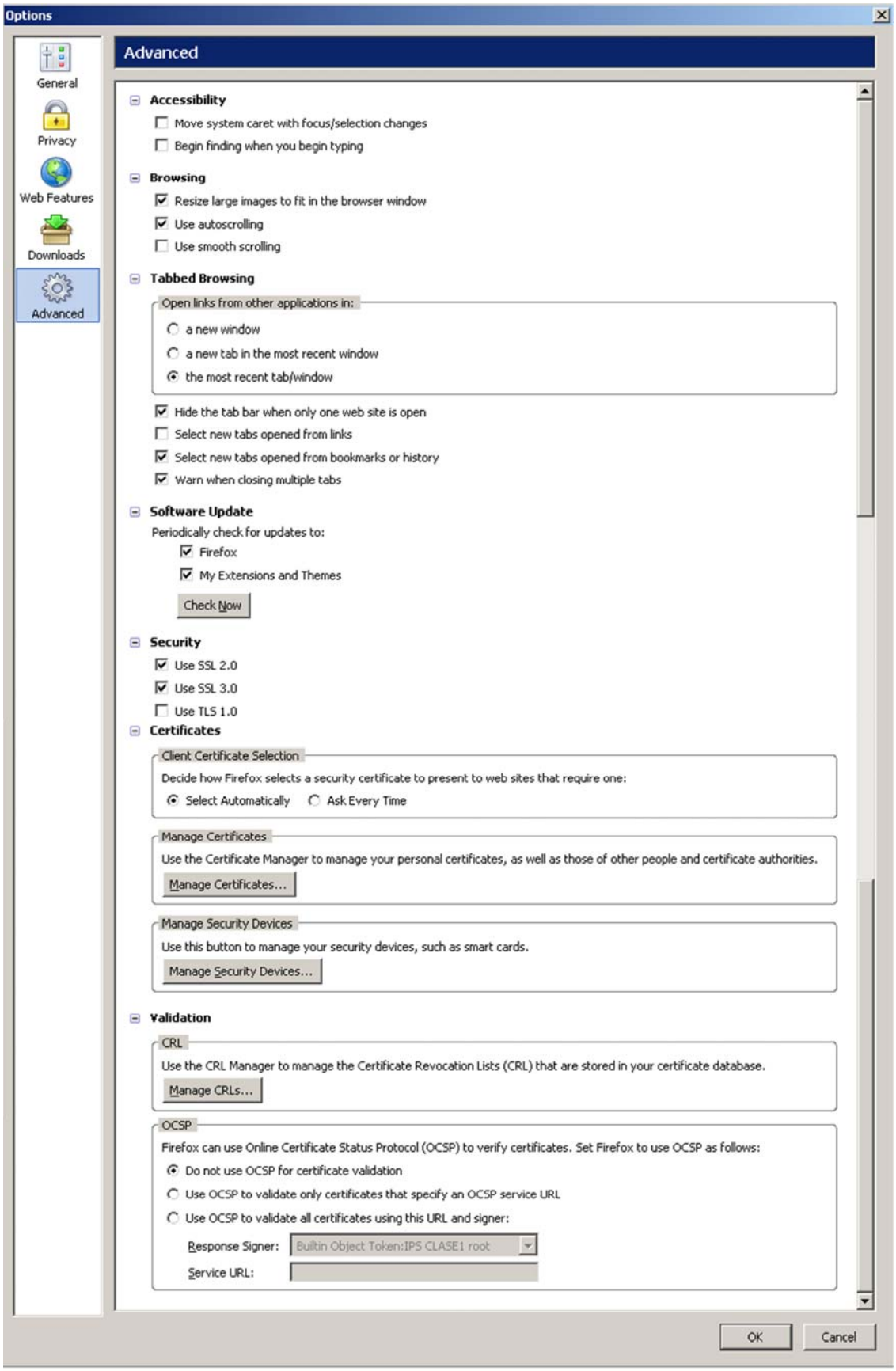
A. IE SETTINGS



B. FIREFOX SETTINGS






Options
General
Privacy
Web Features
Downloads
Advanced
Advanced
Accessibility
Move system caret with focus/selection changes
Begin finding when you begin typing
Browsing
Resize large images to fit in the browser window
Use autoscrolling
Use smooth scrolling
Tabbed Browsing
Open links from other applications in:
a new window
a new tab in the most recent window
the most recent tab/window
Hide the tab bar when only one web site is open
Select new tabs opened from links
Select new tabs opened from bookmarks or history
Warn when closing multiple tabs
Software Update
Periodically check for updates to:
Firefox
My Extensions and Themes
Check Now
Security
Use SSL 2.0
Use SSL 3.0
Use TLS 1.0
Certificates
Client Certificate Selection
Decide how Firefox selects a security certificate to present to web sites that require one:
Select Automatically
Ask Every Time
Manage Certificates
Use the Certificate Manager to manage your personal certificates, as well as those of other people and certificate authorities.
Manage Certificates...
Manage Security Devices
Use this button to manage your security devices, such as smart cards.
Manage Security Devices...
Validation
CRL
Use the CRL Manager to manage the Certificate Revocation Lists (CRL) that are stored in your certificate database.
Manage CRLs...
OCSP
Firefox can use Online Certificate Status Protocol (OCSP) to verify certificates. Set Firefox to use OCSP as follows:
Do not use OCSP for certificate validation
Use OCSP to validate only certificates that specify an OCSP service URL
Use OCSP to validate all certificates using this URL and signer:
Response Signer:
Builtin Object Token:IPS CLASE1 root
Service URL:
OK
Cancel

C. PREF.JS

```
# Mozilla User Preferences

/* Do not edit this file.
 *
 * If you make changes to this file while the browser is running,
 * the changes will be overwritten when the browser exits.
 *
 * To make a manual change to preferences, you can visit the URL about:config
 * For more information, see http://www.mozilla.org/unix/customizing.html#prefs
 */

user_pref("browser.anchor_color", "#0000FF");
user_pref("browser.display.background_color", "#C0C0C0");
user_pref("browser.display.use_system_colors", true);
user_pref("browser.download.defaultFolder", "C:\\FFDownloads");
user_pref("browser.download.dir", "C:\\FFDownloads");
user_pref("browser.download.folderList", 2);
user_pref("browser.download.manager.showAlertOnComplete", false);
user_pref("browser.history_expire_days", 20);
user_pref("browser.preferences.lastpanel", 1);
user_pref("browser.search.selectedEngine", "Google");
user_pref("browser.shell.checkDefaultBrowser", false);
user_pref("browser.startup.homepage", "about:blank");
user_pref("browser.startup.homepage_override.mstone", "rv:1.7.8");
user_pref("browser.visited_color", "#800080");
user_pref("extensions.disabledObsolete", true);
user_pref("extensions.lastAppVersion", "1.0");
user_pref("intl.charsetmenu.browser.cache", "ISO-8859-1, UTF-8");
user_pref("network.cookie.prefsMigrated", true);
user_pref("network.http.proxy.version", "1.0");
user_pref("security.OCSP.URL", "");
user_pref("security.OCSP.signingCA", "Builtin Object Token:IPS CLASE1 root");
user_pref("security.enable_tls", false);
user_pref("security.warn_entering_secure", false);
user_pref("security.warn_leaving_secure", false);
user_pref("xpinstall.whitelist.add", "");
user_pref("xpinstall.whitelist.add.103", "");
```

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Hugo A. Badillo
   NSA
   Fort Meade, MD

4. George Bieber
   OSD
   Washington, DC

5. RADM Joseph Burns
   Fort George Meade, MD

6. John Campbell
   National Security Agency
   Fort Meade, MD

7. Deborah Cooper
   DC Associates, LLC
   Roslyn, VA

8. CDR Daniel L. Currie
   PMW 161
   San Diego, CA

9. Louise Davidson
   National Geospatial Agency
   Bethesda, MD

10. Vincent J. DiMaria
    National Security Agency
    Fort Meade, MD

11. LCDR James Downey
NAVSEA
Washington, DC

12. Dr. Diana Gant
National Science Foundation

13. Jennifer Guild
SPAWAR
Charleston, SC

14. Richard Hale
DISA
Falls Church, VA

15. LCDR Scott D. Heller
SPAWAR
San Diego, CA

16. Wiley Jones
OSD
Washington, DC

17. Russell Jones
N641
Arlington, VA

18. David Ladd
Microsoft Corporation
Redmond, WA

19. Dr. Carl Landwehr
National Science Foundation
Arlington, VA

20. Steve LaFountain
NSA
Fort Meade, MD

21. Dr. Greg Larson
IDA
Alexandria, VA

22. Penny Lehtola
NSA
Fort Meade, MD

23. Ernest Lucier
Federal Aviation Administration
Washington, DC

24. CAPT Deborah McGhee
Headquarters U.S. Navy
Arlington, VA

25. Dr. Vic Maconachy
NSA
Fort Meade, MD

26. Doug Maughan
Department of Homeland Security
Washington, DC

27. Dr. John Monastra
Aerospace Corporation
Chantilly, VA

28. John Mildner
SPAWAR
Charleston, SC

29. Jim Roberts
Central Intelligence Agency
Reston, VA

30. Charles Sherupski
Sherassoc
Round Hill, VA

31. Dr. Ralph Wachter
ONR
Arlington, VA

32. David Wirth
N641
Arlington, VA

33. Daniel Wolf
NSA
Fort Meade, MD

34. Jim Yerovi
NRO
Chantilly, VA

35. CAPT Robert Zellmann
CNO Staff N614
Arlington, VA

36. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA

37. Timothy E. Levin
Naval Postgraduate School
Monterey, CA

38. Mark Barwinski
Civilian, Naval Postgraduate School
Monterey, CA