Automated Web Patrol with Strider HoneyMonkeys: *Finding Web Sites That Exploit Browser Vulnerabilities*

Yi-Min Wang
Doug Beck
Xuxian Jiang
Roussi Roussev

First Version: June 4, 2005
Last Updated: July 27, 2005

Technical Report
MSR-TR-2005-72

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Automated Web Patrol with Strider HoneyMonkeys: *Finding Web Sites That Exploit Browser Vulnerabilities*

Yi-Min Wang, Doug Beck, Xuxian Jiang, and Roussi Roussev

Cybersecurity and Systems Management Research Group

Microsoft Research, Redmond, Washington

Abstract

Internet attacks that use Web servers to exploit browser vulnerabilities to install malware programs are on the rise [D04,R04,B04,S05]. Several recent reports suggested that some companies may actually be building a business model around such attacks [IF05,R05]. Expensive, manual analyses for individually discovered malicious Web sites have recently emerged [F04,G05]. In this paper, we introduce the concept of *Automated Web Patrol*, which aims at significantly reducing the cost for monitoring malicious Web sites to protect Internet users. We describe the design and implementation of the *Strider HoneyMonkey Exploit Detection System* [L05,N05], which consists of a network of monkey programs running on virtual machines with different patch levels and constantly patrolling the Web to hunt for Web sites that exploit browser vulnerabilities.

Within the first month of utilizing this new system, we identified 752 unique URLs that are operated by 287 Web sites and that can successfully exploit unpatched WinXP machines. The system automatically constructs topology graphs that capture the connections between the exploit sites based on traffic redirection, which leads to the identification of several major players who are responsible for a large number of exploit pages.

1. Introduction

Internet attacks that use a malicious, hacked, or infected Web server to exploit unpatched client-side vulnerabilities of visiting browsers are on the rise. Many attacks in the past 12 months fell into this category, including Download.Ject [D04], Bofra [R04], and Xpire.info [B04]. Such attacks allow the attackers to install malware programs without requiring any user interaction. Manual analyses of ***exploit sites*** have recently emerged [F04,G05,IF05,R05,S05,T05]. Although they often provide very useful and detailed information about which vulnerabilities are exploited and which malware programs are installed, such analysis efforts are not scalable and do not provide a comprehensive picture of the problem.

In this paper, we introduce the concept of ***Automated Web Patrol*** that involves a network of automated agents actively patrolling the Web to find malicious Web sites. We describe the design and implementation of the ***Strider HoneyMonkey Exploit Detection System*** that uses *active client honeypots* [H,HC] to perform automated Web patrol with the specific goal of finding Web sites that exploit browser vulnerabilities. The system consists of a *pipeline* of *monkey programs* running on Virtual Machines (VMs) with different patch levels in order to detect exploit sites with different capabilities.

Within the first month, we identified 752 unique URL links, operated by 287 Web sites, that can successfully exploit unpatched WinXP machines. We present a preliminary analysis of the data and suggest what can be done based on the data to improve Internet safety.

2. Automated Web Patrol with the Strider HoneyMonkey Exploit Detection System

Although the general idea of crawling the Web to look for pages of particular interest is fairly straightforward, we have found that a combination of the following key ideas are most crucial for increasing the “hit rate” and making the concept of Web patrol useful in practice:

1. **Where Do We Start?** There are 10 billion Web pages out there and most of them do not exploit browser vulnerabilities. So, it is very important to start with a list of URLs that are most likely to generate hits. The approach we took was to collect an initial list of 5000+ potentially malicious URLs by doing a Web search for Windows “hosts” files [HF] that are used to block advertisements and bad sites, and lists of known-bad Web sites that host some of the most malicious spyware programs [CWS05].

2. **How Do We Detect An Exploit?** One method of detecting a browser exploit is to study all known vulnerabilities and build signature-based detection code for each. Since this approach is fairly expensive, we decided to lower the cost of Web patrol by utilizing a *black-box approach*: we run a monkey program[1] with the Strider Flight Data Recorder (FDR) [W03] to efficiently record every single file and Registry read/write. The monkey launches a browser instance for each suspect URL and wait for a few minutes. The monkey is not set up to click on any dialog box to permit installation of any software; consequently, any executable files that get created outside the browser’s temporary folder are detected by the FDR and signal an exploit. Such a black-box approach has an important advantage: it allows the detection of known-vulnerability exploits and ***zero-day exploits*** in a uniform way, through monkeys with different patch levels. Each monkey also runs with the Strider Gatekeeper [W04] to detect any hooking of Auto-Start Extensibility Points (ASEPs) that may not involve creation of executables, and with Strider GhostBuster [W05] to detect stealth malware that hide processes and ASEP hooks.

[1] An automation-enabled program such as the Internet Explorer browser allows programmatic access to most of the operations that can be invoked by a user. A “monkey program” is a program that drives the browser in a way that mimics human user’s operation.

To ease cleanup of infected state, we run the monkey inside a VM. Upon detecting an exploit, the monkey persists its data and notifies the ***Monkey Controller***, running on the host machine, to destroy the infected VM and restart a clean VM. The restarted VM automatically launches the monkey, which then continues to visit the remaining URL list. The Monkey Controller also passes the detected ***exploit URL*** to the next monkey in the pipeline to continue investigating the strength of the exploit. When the end-of-the-pipeline monkey, running on a fully patched VM, reports a URL as an exploit, the URL is upgraded to a zero-day exploit and the malware programs that it installed are immediately investigated and passed on to the Microsoft Security Response Center.

3. **How Do We Expand And Guide The Search?** The initial list may contain only a small number of hits, so it is important to have an effective guided search in order to grow the patrolled area to increase coverage. We have found that, usually, the links displayed on an exploit page have a higher probability of being exploit pages as well because people in the exploit business like to refer to each other to increase traffic. We take advantage of this by doing Web crawling through those links to generate bigger "bad neighborhoods". More importantly, we have observed that many of the exploit pages automatically redirect visiting browsers to a number of other pages, each of which may try a different exploit or install different malware programs. We take advantage of this by tracking such redirections to enable automatic derivation of their business relationship: ***content providers*** are responsible for attracting browser traffic and selling/redirecting the traffic to ***exploit providers***, which specialize in and are responsible for actually performing the exploits to install malware.

3. Exposing and Analyzing The Exploit Sites

3.1. The Importance of Software Patching

Figure 1 shows the breakdown of the 752 Internet Explorer browser-based exploit URLs among different service-pack (SP1 or SP2) and patch levels, where "UP" stands for "UnPatched", "PP" stands for "Partially Patched", and "FP" stands for "Fully Patched". As expected, the SP1-UP number is much higher than the SP2-UP number because the former has more vulnerabilities and they have existed for a longer time.

The SP2-PP numbers are the numbers of exploit pages and sites that successfully exploited a WinXP SP2 machine partially patched up to early 2005. The fact that the number is one order of magnitude lower than the SP2-UP number demonstrates the importance of keeping software up to date. But since the number is still not negligible, enterprise corporate proxies may want to black-list these sites while the latest updates are still being tested for deployment and ISPs may want to block access to these sites to give their consumer customers more time to catch up with the updates.

	Number of Unique Exploit URLs	Number of Exploit Sites
Total	752	287
WinXP SP1 Unpatched (SP1-UP)	688	270
WinXP SP2 Unpatched (SP2-UP)	204	115
WinXP SP2 Partially Patched (SP2-PP)	17	10
WinXP SP2 Fully Patched (SP2-FP)	0	0

Figure 1. Number of Exploit URLs and sites as a Function of Patch Levels (May/June 2005 data).

The SP2-FP numbers again demonstrate the importance of keeping software up to date: none of the 752 exploit URLs was able to exploit a fully updated WinXP SP2 machine according to our May/June 2005 data. If any Web site that exploits a zero-day vulnerability ever appears and gets connected to any of these URLs, our SP2-FP HoneyMonkey will be able to quickly detect and report it to the browser and security response teams. This hopefully creates a dilemma that discourages the exploiters: most of the future exploit pages will likely get detected before they have a chance to cause large-scale infections because HoneyMonkeys browse the Web like humans and the first HoneyMonkey that gets infected can report the exploit.

3.2. Connection Topology based on Traffic Redirection

Next, we present the topology graph for each of the first three patch levels and discuss what we can learn from each graph.

3.2.1. "WinXP SP1 Unpatched" Topology

Figure 2 shows the ***URL-level*** topology graph for WinXP SP1-UP. Each rectangular node represents an individual exploit URL. Blue nodes represent Web pages that did not receive redirected traffic from any other nodes; they are most likely content providers and not major exploit providers. In contrast, red nodes represent Web pages that received redirected traffic from other exploit pages; they are most likely exploit providers if the traffic came from multiple different sites. Each gray edge represents an automatic traffic redirection. Each circle represents a ***site node*** that serves as an aggregation point for all exploit pages hosted on that site, with the site node having one blue or red edge pointing to each of the child-page rectangles. Any circle without a border is a "virtual site node" that does not correspond to an exploit URL, but is introduced purely for aggregation purposes.

The size of a circle is proportional to the number of outgoing gray edges for blue nodes and the number of incoming gray edges for red nodes. Such numbers provide a good indication of the relative popularity of the exploit sites and will be referred to as the ***connection counts***. The top exploit site in this graph has a connection count of 63; the top exploit page has a count of 29; the largest blue circle at the top has a count of

19. If available, actual amount of visit traffic to each exploit site can provide a more accurate picture of the relative popularity.

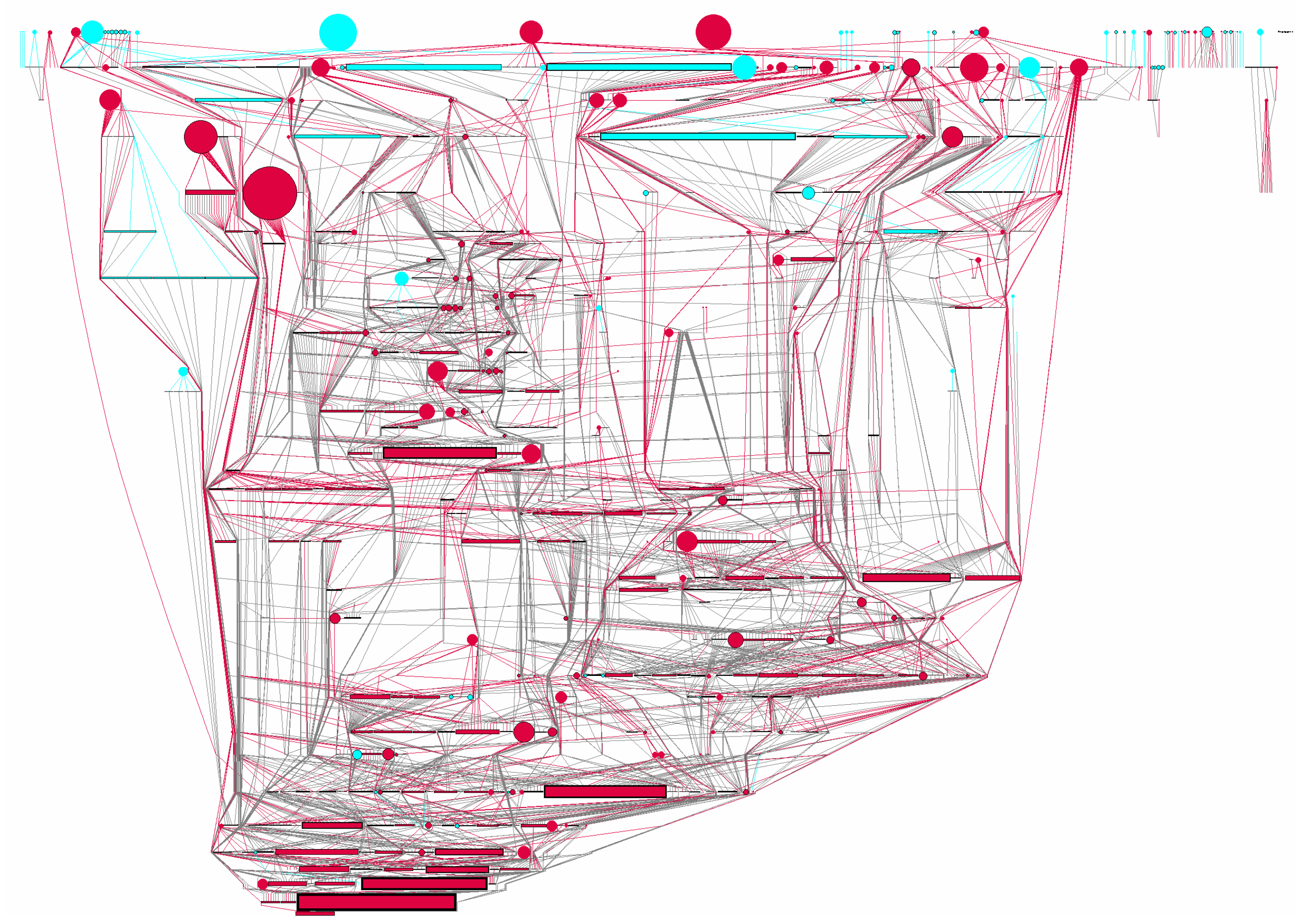


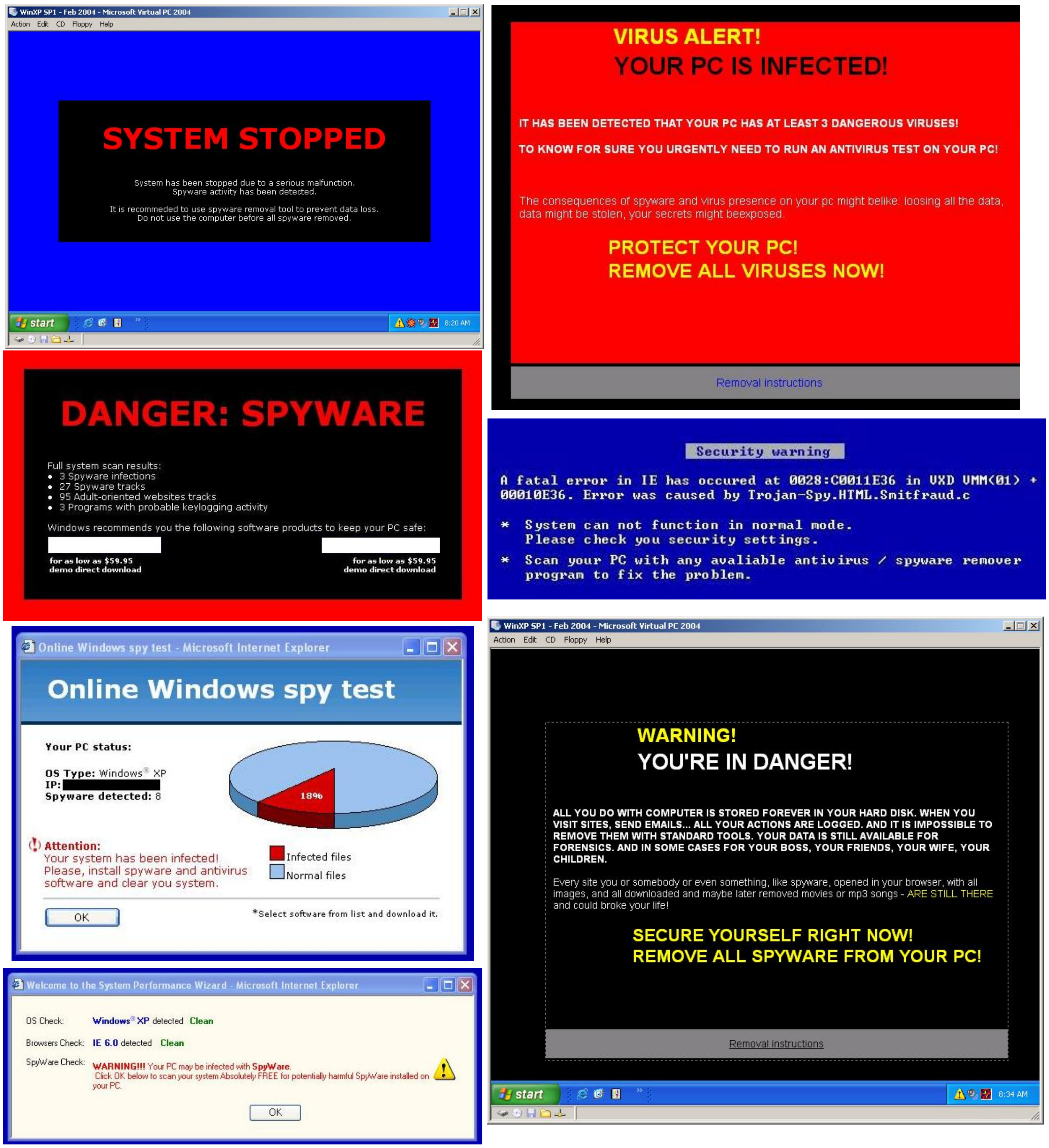
Figure 2. URL-level Topology Graph for WinXP SP1 Unpatched: 688 URLs from 270 sites

It is clear from Figure 2 that there are many major exploit providers as marked by the large red circles. We performed WHOIS lookups on the top 30 exploit providers and found that several individuals actually own more than one of these top sites and the registration addresses of these individuals span more than 10 countries. The top exploiter in our current SP1-UP network owns the three big red circles in the upper-left corner of Figure 2, which are ranked #1, #3, and #10, respectively, based on connection counts. We found that a total of 61 exploit sites have direct traffic redirection relationship with this exploiter’s three sites.

Although most of the exploit URLs in Figure 2 appear to be pornography sites that are well connected to each other, there are several visibly isolated sub-graphs. We discuss three groups that are of particular interest. The first group consists of a normal-looking shopping site that performs exploits by redirecting traffic to seven exploit URLs hosted by five advertising companies. We have verified that one of the five companies is also serving exploiting ads on a large number of popular Web pages.

The second group is a site related to screen savers. It is well known that many freeware programs bundle spyware programs in their installations; now it appears that some freeware sites are actually installing spyware through exploits. The third group consists of malicious search sites; there are more than 20 search sites in our current list of exploit URLs.

Ironically, it is not uncommon to see rogue anti-spyware vendors involved in exploit activities. Figure 3 shows screenshots of several spyware warnings following exploit-based spyware installations. Several of them try to sell to the users anti-spyware programs in order to remove spyware programs that, they themselves, have installed.


WinXP SP1 - Feb 2004 - Microsoft Virtual PC 2004
Action Edit CD Floppy Help
start
8:20 AM

SYSTEM STOPPED

System has been stopped due to a serious malfunction.
Spyware activity has been detected.

It is recommeded to use spyware removal tool to prevent data loss.
Do not use the computer before all spyware removed.

VIRUS ALERT!
YOUR PC IS INFECTED!

IT HAS BEEN DETECTED THAT YOUR PC HAS AT LEAST 3 DANGEROUS VIRUSES!

TO KNOW FOR SURE YOU URGENTLY NEED TO RUN AN ANTIVIRUS TEST ON YOUR PC!

The consequences of spyware and virus presence on your pc might belike: loosing all the data, data might be stolen, your secrets might beexposed.

PROTECT YOUR PC!
REMOVE ALL VIRUSES NOW!

Removal instructions

DANGER: SPYWARE

Full system scan results:
- 3 Spyware infections
- 27 Spyware tracks
- 95 Adult-oriented websites tracks
- 3 Programs with probable keylogging activity

Windows recommends you the following software products to keep your PC safe:

for as low as $59.95
demo direct download

for as low as $59.95
demo direct download

Security warning

A fatal error in IE has occured at 0028:C0011E36 in UXD UMM(01) + 00010E36. Error was caused by Trojan-Spy.HTML.Smitfraud.c

* System can not function in normal mode.
Please check you security settings.

* Scan your PC with any avaliable antivirus / spyware remover program to fix the problem.

Online Windows spy test

Your PC status:

OS Type: Windows® XP
IP: [redacted]
Spyware detected: 8

Attention:
Your system has been infected!
Please, install spyware and antivirus software and clear you system.

*Select software from list and download it.

Welcome to the System Performance Wizard - Microsoft Internet Explorer

OS Check:	Windows® XP detected Clean
Browsers Check:	IE 6.0 detected Clean
SpyWare Check:	WARNING!!! Your PC may be infected with SpyWare. Click OK below to scan your system Absolutely FREE for potentially harmful SpyWare installed on your PC.

OK
WinXP SP1 - Feb 2004 - Microsoft Virtual PC 2004
Action Edit CD Floppy Help

WARNING!
YOU'RE IN DANGER!

ALL YOU DO WITH COMPUTER IS STORED FOREVER IN YOUR HARD DISK. WHEN YOU VISIT SITES, SEND EMAILS... ALL YOUR ACTIONS ARE LOGGED. AND IT IS IMPOSSIBLE TO REMOVE THEM WITH STANDARD TOOLS. YOUR DATA IS STILL AVAILABLE FOR FORENSICS. AND IN SOME CASES FOR YOUR BOSS, YOUR FRIENDS, YOUR WIFE, YOUR CHILDREN.

Every site you or somebody or even something, like spyware, opened in your browser, with all images, and all downloaded and maybe later removed movies or mp3 songs - ARE STILL THERE and could broke your life!

SECURE YOURSELF RIGHT NOW!
REMOVE ALL SPYWARE FROM YOUR PC!

Removal instructions

Figure 3. Screenshots of Exploit-based Spyware Installations and Warnings

3.2.2. "WinXP SP2 Unpatched" Topology

Figure 4 illustrates the ***site-level*** topology for SP2-UP. Each node in a site-level topology graph represents a folded sub-graph consisting of a site node and all its child pages with colored incoming edges from a URL-level graph. A site-level topology graph is most useful for showing the potential business relationship among the players. The clean separation between the blue and red nodes in Figure 4 raises the possibility of disrupting the network by blocking or pursuing legal actions when appropriate against exploit providers, and monitoring content providers to detect new exploit providers that join the network. In some cases, content providers may have been hacked to do traffic redirection or exploit providers may be hosted on infected zombie machines; identifying them through the HoneyMonkey will help innocent owners clean up and reclaim their machines.

Intersecting the top 30 SP2-UP exploit sites with their SP1-UP counterpart yields 11 sites; these appear to belong to professional exploiters who are continuing to improve their capabilities in order to maintain and increase coverage. Among the remaining 19 sites on the SP2-UP list, 16 of them are known SP1-UP exploit sites but the #2, #10, and #18 sites appear to be targeting SP2 machines.

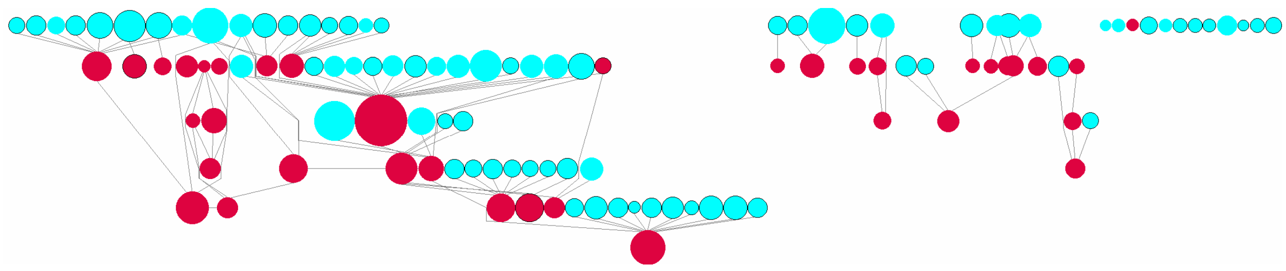


Figure 4. Site-level Topology Graph for WinXP SP2 Unpatched: 115 sites

3.2.3. "WinXP SP2 Partially Patched" Topology

Figure 5 illustrates the URL-level topology for SP2-PP, which consists of only 17 Web pages hosted by 10 sites. These are the most powerful exploit pages in terms of the number of machines they are capable of infecting and should be considered the highest priorities for investigation. It is clear from the picture that blocking the two exploit pages at the bottom of the middle sub-graph is the most effective starting point because it would disrupt more than half of this graph.

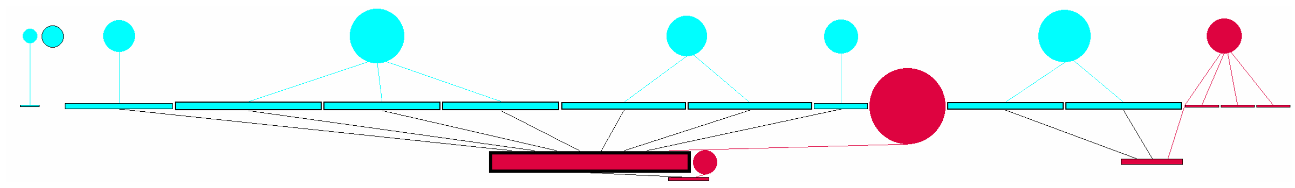


Figure 5. URL-level Topology Graph for WinXP SP2 Partially Patched: 17 URLs from 10 sites

3.3. Search Engines as an Infection Vector

Search engines are popular starting points for Web surfers. This leads to the question: *how many of the 752 exploit URLs exist in popular search engines?* Figure 6 provides the answer for three of them. An URL is counted when its name is supplied as the search string and the search results include a link to the URL. It is clear that a non-trivial percentage of the exploit URLs is returned by popular search engines. The ones that appeared in MSN Search results have since been removed, as shown in the bottom row of Figure 6.

	Number of Exploit URLs
Total	752
In Google search results	102 (13.6%)
In Yahoo search results	100 (13.3%)
In MSN Search results (June 1, 2005)	49 (6.5%)
In MSN Search results (June 10, 2005)	0 (0%)

Figure 6. Number of Exploit URLs That Generated Hits At Three Popular Search Engines

4. Summary, Future Work, and Zero-Day Exploit Detection

We have presented summary statistics and topology graphs for exploit URLs that the HoneyMonkey system discovered in the first round of Web patrol. We plan to continue monitoring all discovered exploit pages to capture new exploiters. Statistics regarding exploit sites for the latest patched vulnerabilities will be used to assess the urgency of patch deployment. Software samples gathered from the exploit URLs will be analyzed and investigated to provide concrete evidence of spyware programs that are being installed without user permission. These results will also be provided to our Enforcement Team, in order to further investigate and possibly pursue legal actions, including referrals, against those exploiters acting in contravention of the law.

We are also monitoring the top one million click-through links from a search engine to see if the exploit industry has penetrated the “good neighborhoods” of popular sites. Preliminary results reveal that contaminated Web pages that unknowingly serve ads that exploit browser vulnerabilities may be a serious concern. We are beginning to monitor links contained in spam and phishing emails because that is another way for the exploiters to lure Web users to the “bad neighborhoods” [S05]. We plan to investigate other infection vectors as well [N05,G05,S04,HD05,IW05,XSS]. In the long run, we envision multiple networks of HoneyMonkeys patrolling the Web from different corners of the world so that it is not possible for the exploiters to black-list HoneyMonkey network IP addresses and deliberately skip exploit.

Update on Zero-Day Exploit Detection: In early July 2005, the HoneyMonkey system discovered its first zero-day exploit. The javaprxy.dll vulnerability was known at that time without an available patch yet

[J105,J205], and whether it was actually being exploited was a very important piece of information. The HoneyMonkey system was able to detect the first exploit page within 2.5 hours of scanning and that was confirmed to be the first in-the-wild exploit URL reported to the Microsoft Security Response Center. Within the subsequent two weeks, we detected that over 40 of those 752 exploit URLs started to "upgrade" to the javaprxy.dll exploit. We were able to identify the three exploit sites behind them that were responsible for all the exploits and report them to the response center. Three interesting observations further demonstrate the effectiveness of the HoneyMonkey system: (1) the first detected zero-day exploit URL belongs to the top exploiter discussed right after Figure 2; (2) all three top SP1-UP exploit sites owned by this exploiter were upgraded within a short time; (3) almost half of the exploit URLs in Figure 5 were upgraded, as expected, because they appear to have the tendency of keeping their exploit capability up-to-date.

References

[B04] Xpire.info, http://www.vitalsecurity.org/xpire-splitinfinity-serverhack_malwareinstall-condensed.pdf, Nov. 2004.

[CWS05] "Webroot: CoolWebSearch Top Spyware Threat," http://www.techweb.com/showArticle.jhtml?articleID=160400314, TechWeb, March 30, 2005.

[D04] Download.Ject, http://www.microsoft.com/security/incident/download_ject.mspx, June 2004.

[F04] "Follow the Money; or, why does my computer keep getting infested with spyware?" http://www.livejournal.com/users/tacit/125748.html.

[G05] "Googkle.com installed malware by exploiting browser vulnerabilities," http://www.f-secure.com/v-descs/googkle.shtml.

[H] The Honeynet Project, http://www.honeynet.org/.

[HC] Honeyclient Development Project, http://www.honeyclient.org/.

[HD05] "Another round of DNS cache poisoning," Handlers Diary, March 30, 2005, http://isc.sans.org/.

[HF] hpHOSTS community managed hosts file, http://www.hosts-file.net/downloads.html.

[IF05] "iframeDOLLARS dot biz partnership maliciousness," http://isc.sans.org/diary.php?date=2005-05-23.

[IW05] "Scammers use Symantec, DNS holes to push adware," InfoWorld.com, March 7, 2005, http://www.infoworld.com/article/05/03/07/HNsymantecholesandadware_1.html?DESKTOP%20SECURITY.

[J105] Microsoft Security Advisory (903144) - A COM Object (Javaprxy.dll) Could Cause Internet Explorer to Unexpectedly Exit, http://www.microsoft.com/technet/security/advisory/903144.mspx.

[J205] Microsoft Security Bulletin MS05-037 - Vulnerability in JView Profiler Could Allow Remote Code Execution (903235), http://www.microsoft.com/technet/security/bulletin/ms05-037.mspx.

[L05] Robert Lemos, "Microsoft looks to "monkeys" to find Web threats," SecurityFocus News, May 17, 2005, http://www.securityfocus.com/news/11178.

[N05] Ryan Naraine, “Strider HoneyMonkey: Trawling for Windows Exploits,” eWEEK.com, May 19, 2005, http://www.eweek.com/article2/0,1759,1817822,00.asp.

[R05] “Russians use affiliate model to spread spyware,” http://www.itnews.com.au/newsstory.aspx?CIaNID=18926.

[R04] Team Register, “Bofra exploit hits our ad serving supplier,” http://www.theregister.co.uk/2004/11/21/register_adserver_attack/, November 2004.

[S04] Symantec Gateway Security Products DNS Cache Poisoning Vulnerability, http://securityresponse.symantec.com/avcenter/security/Content/2004.06.21.html.

[S05] “Michael Jackson suicide spam leads to Trojan horse,” http://www.sophos.com/virusinfo/articles/jackotrojan.html, Sophos, June 9, 2005.

[T05] Michael Ligh, “Tri-Mode Browser Exploits - MHTML, ANI, and ByteVerify,” http://www.mnin.org/write/2005_trimode.html, April 30, 2005.

[XSS] “Code insertion in Blogger comments”, March 28, 2005, http://www.securityfocus.com/archive/1/394532.

[W03] Yi-Min Wang, et al., "STRIDER: A Black-box, State-based Approach to Change and Configuration Management and Support", in *Proc. Usenix LISA*, Oct. 2003.

[W04] Yi-Min Wang, et al., "Gatekeeper: Monitoring Auto-Start Extensibility Points (ASEPs) for Spyware Management", in *Proc. Usenix LISA*, 2004

[W05] Yi-Min Wang, et al., "Detecting Stealth Software with Strider GhostBuster," to appear in *Proc. DSN*, June 2005