Measurement and Analysis of Spyware in a University Environment

Stefan Saroiu, Steven D. Gribble, and Henry M. Levy
Department of Computer Science & Engineering
University of Washington
{tzoompy,gribble,levy}@cs.washington.edu

Abstract

Over the past few years, a relatively new computing phenomenon has gained momentum: the spread of "spyware." Though most people are aware of spyware, the research community has spent little effort to understand its nature, how widespread it is, and the risks it presents. This paper is a first attempt to do so.

We first discuss background material on spyware, including the various types of spyware programs, their methods of transmission, and their run-time behavior. By examining four widespread programs (Gator, Cydoor, SaveNow, and eZula), we present a detailed analysis of their behavior, from which we derive signatures that can be used to detect their presence on remote computers through passive network monitoring. Using these signatures, we quantify the spread of these programs among hosts within the University of Washington by analyzing a week-long trace of network activity. This trace was gathered from August 26th to September 1st, 2003.

From this trace, we show that: (1) these four programs affect approximately 5.1% of active hosts on campus, (2) many computers that contain spyware have more than one spyware program running on them concurrently, and (3) 69% of organizations within the university contain at least one host running spyware. We conclude by discussing security implications of spyware and specific vulnerabilities we found within versions of two of these spyware programs.

1 Introduction

Over the past few years, a relatively new computing phenomenon has gained momentum: the spread of spyware. Although there is no precise definition, the term "spyware" is commonly used to refer to software that, from a user's perspective, gathers information about a computer's use and relays that information back to a third party. This data collection occurs sometimes with, but often without, the knowing consent of the user. In this paper, we use the term spyware in conformity with this common usage.[1] Spyware may also appropriate resources of the computer that it infects [15] or alter the functions of existing applications on the affected computer to the benefit of a third party [12].

Spyware poses several risks. The most conspicuous is compromising a user's privacy by transmitting information about that user's behavior. However, spyware can also detract from the usability and stability of a user's computing environment, and it has the potential to introduce new security vulnerabilities to the infected host. Because spyware is widespread, such vulnerabilities would put millions of computers at risk. In Section 5, we demonstrate vulnerabilities within versions of two widely deployed spyware programs, and we discuss the potential impact of such flaws.

Though most people are aware of spyware, the research community has to date spent little effort understanding the nature and extent of the spyware problem. This paper is an initial attempt to do so. First, we give an overview of spyware in general, in which we discuss the various kinds of spyware programs, their behavior, how they typically infect computers, and the proliferation of new varieties of spyware programs. Next, we examine four particularly widespread spyware programs (Gator, Cydoor, SaveNow, and eZula), and we present a detailed description of their behavior. Our examination was limited to software versions released between August 2003 and the January 2004; as such, our observations and results might not hold for other versions.

Based on our examination, we derive network signatures that can be used to detect the presence of these programs on remote computers by monitoring network traffic. With these signatures, we gather a week-long trace of network traffic exchanged between the University of Washington (a large public university) and the Internet,

[1] Deciding whether a particular program should be called spyware or not can be both difficult and delicate. In practice, there is a continuous spectrum of program behavior that spans from malicious and invasive to fully legitimate. In this paper, we use the term spyware very broadly, and in general apply the term as might be commensurate with the experience of an unsophisticated user. However, we are careful to describe the precise behavior of individual programs discussed in this paper.

from August 26th to September 1st, 2003. We perform a quantitative study of spyware based on this trace, characterizing the spread of the four spyware programs within the university.

Though hundreds of spyware programs exist, our findings show that these four programs alone affect approximately 5.1% of active university hosts, and that these hosts often have more than one spyware program running. Additionally, we find that a majority of organizations within the university contain at least one spyware-infected host, suggesting that existing organization-specific security policies and mechanisms (such as perimeter firewalls) are not effective at preventing spyware installation. Even though our measurements are gathered at only one site, and hence may not be representative of the Internet at large, we believe our results confirm that spyware is a significant problem.

The rest of this paper is organized as follows. In Section 2 we set the context of our study with a brief discussion on the general characteristics of spyware. In Section 3 we narrow our focus to four prevalent spyware programs, giving a detailed description of their behavior. Section 4 presents quantitative results based on our week-long network trace. We discuss implications of our results in Section 5, we present related work in Section 6, and we conclude in Section 7.

2 A Brief Spyware Primer

Spyware exists because information has value. For example, information gathered about the demographics and behavior of Internet users has value to advertisers, the ability to show advertisements correlated with user behavior has value to product vendors, and gathering keystrokes or introducing backdoor vulnerabilities on a host has value to attackers. As long as this value exists, there will be incentive to create spyware programs to capitalize on it.

People are typically exposed to spyware as a result of their behavior. Users may install popular software packages that contain embedded spyware, Web sites may prompt users to install Web browser extensions that contain spyware, and Web browsers retain 'cookies' to track user behavior across collections of cooperating Web sites. The constant growth in the number of Internet users and the increasing amount of time users spend on the Internet have served to amplify users' exposure to spyware.

Spyware succeeds because today's desktop operating systems make spyware simple to build and install. Operating systems and applications are designed to be extensible, and as a result, there are numerous interfaces for interposing on events and interacting with other programs. Operating systems also tend to hide information about background activities to shield users from unwanted complexity. The combination of these two properties makes it difficult to prevent spyware programs from gathering the information they want, or for the user to detect when such information is being harvested or transmitted. As is often the case, there is a tension between usability and security, and to date market pressures appear to favor usability.

2.1 Classes of Spyware

There are many different kinds of spyware. Borrowing from the terminology used in SpyBot S&D [17], a free spyware removal tool, we define the following classes:

- **Cookies and Web bugs:** Cookies are small pieces of state stored on individual clients' Web browsers on behalf of Web servers. Cookies can only be retrieved by the Web site that initially stored them. However, because many sites use the same advertisement provider, these providers can potentially track the behavior of users across many Web sites. Web bugs – invisible images embedded on pages – are related to cookies in that advertisement networks often contract with Web sites to place such bugs on their pages. Cookies and Web bugs are purely passive forms of spyware; they contain no code of their own, relying instead on existing Web browser functions.
- **Browser hijackers:** Hijackers attempt to change a user's Web browser settings to modify their start page, search functionality, or other browser settings. Hijackers, which predominantly affect Windows operating systems, may use one of several mechanisms to achieve their goal: installing a browser extension (called a "browser helper object," or BHO), modifying Windows registry entries, or directly modifying or replacing browser preference files.
- **Keyloggers:** Keyloggers were originally designed to record all keystrokes of users in order to find passwords, credit card numbers, and other sensitive information. Keyloggers have expanded in scope, capturing logs of Websites visited, instant messaging sessions, windows opened, and programs executed.
- **Tracks:** A "track" is a generic name for information recorded by an operating system or application about actions the user has performed. Examples of tracks include recently visited Website lists maintained by most browsers and lists of recently opened files and programs maintained by most operating systems. Although a track is typically innocuous on its own, tracks can be mined by malicious programs.

- **Malware:** Malware refers to a variety of malicious software, including viruses, worms, trojan horses, and automatic phone dialers (which attempt to dial modems to connect to expensive services).

- **Spybots:** Spybots are the prototypical example of "spyware." A spybot monitors a user's behavior, collecting logs of activity and transmitting them to third parties. Examples of collected information include fields typed in Web forms, lists of email addresses to be harvested as spam targets, and lists of visited URLs. A spybot may be installed as a browser helper object, it may exist as a DLL on the host computer, or it may run as a separate process launched whenever the host OS boots.

- **Adware:** "Adware," a more benign variety of spybot, is software that displays advertisements tuned to the user's current activity, potentially reporting aggregate or anonymized browsing behavior to a third party.

Many instances of spyware have the ability to *self-update*, or download new versions of themselves automatically. Self-updating allows spyware authors to introduce new functions over time, but it also may be used to evade anti-spyware tools, by avoiding specific signatures contained within the tools' signature databases.

2.2 The Diversity and Extent of Spyware

Our measurements in Section 4 provide quantitative data on the spread of spyware within an organization. We can also obtain some insight into the extent of the spyware problem by considering other sources of data. One such source of data is the set of spyware signatures that anti-spyware tools have accumulated over time. These signatures are used to compare files and registry entries on a given computer against a list of known spyware programs.

SpyBot S&D [17] is a popular shareware spyware removal tool for Windows-based operating systems. As of January 27, 2004, SpyBot's database contains entries describing 790 different spyware instances. Table 1 breaks down these entries across the previously defined categories. While SpyBot S&D's database is almost certainly incomplete, it demonstrates that there is a substantial number of spyware programs in existence today.

Many spyware infections occur because of spyware programs that are piggybacked on popular software packages. Given this, another interesting source of data to consider is popular shareware and freeware programs. C|Net's *http://download.com/* Website provides free access to over 30,000 freeware and shareware software titles, as well as download statistics about these titles. As a simple experiment, we downloaded the top ten most popular software titles (as of August 2003) and used SpyBot S&D to test each program for spyware.

spyware category	cookies and web bugs	browser hijackers	key-loggers	tracks	malware	spybots
# of DB entries	34	153	62	231	168	142

Table 1. Number of entries in SpyBot S&D's database. The database contains 790 total spyware instances as of January 27, 2004. There is significant diversity in spyware, as these instances are spread across all categories.

Together, these ten titles account for over 872 million reported downloads from the C|Net site. They include three peer-to-peer file-sharing clients, three instant messaging clients, a file compression utility, a download manager, and two anti-spyware tools. Of these ten titles, we found that spyware is packaged with four of them: the software ranked #1, #4, #9, and #10 (Kazaa, iMesh, Morpheus, and Download Accelerator Plus, respectively). These programs have been downloaded over 470 million times. The most popular program (Kazaa Media Desktop) by itself has been downloaded over 265 million times and contains several different types of spyware.[2] Assuming C|Net's data is correct, hundreds of millions of users have been exposed to spyware from this source alone.

To help understand whether the bundling of spyware in free software is a recent phenomenon, we examined several versions of Kazaa Media Desktop released over the past two years. Table 2 shows our results. Twelve different spyware programs have been bundled with Kazaa, and every version of Kazaa released has included at least two different spyware programs. Spyware in free software is not a recent phenomenon—it has been occurring for several years.

Although neither the SpyBot S&D database metrics nor the *http://download.com/* statistics are precise indicators of the extent of the spyware problem, they do reveal that the problem is significant in scope. In the next section of this paper, we narrow our focus to four specific spyware programs. In Section 4, we use our findings to measure the extent to which these four spyware programs have infected hosts at the University of Washington.

3 Gator, Cydoor, SaveNow and eZula

An exhaustive measurement of all types and instances of spyware is well beyond the scope of one paper. Instead, we selected four specific Windows-based programs to examine in detail: Gator, Cydoor, SaveNow,

[2] Kazaa is now distributed in two versions: a free version that contains spyware, and a paid version without spyware.

version	1.3.3	1.4	1.5	1.6	1.7	2.0	2.1	2.1.1	2.6
released	12/01	01/02	02/02	04/02	05/02	09/02	02/03	05/03	11/03
Gator									X
SaveNow	X	X	X	X	X	X	X	X	
Cydoor	X	X	X	X	X	X			X
BDE	X	X	X	X	X	X			
VX2	X	X							
New.net	X	X	X	X	X	X			
OnFlow	X	X						X	
D/L-Ware					X	X	X		
CmnName	X	X	X	X	X	X			X
PromulGate						X			
DirecTVIcon			X	X					
MySearch									X

Table 2. Spyware bundled with Kazaa. This table shows the 10 different programs that were bundled with Kazaa at various points in time, for software versions released between December 2001 and November 2003.

and eZula. We chose these four programs out of the hundreds of possibilities available to us for several reasons. First, anecdotal evidence suggests that these are among the most widely spread instances of spyware. Second, we were successful in deriving signatures that allowed us to detect them remotely with high confidence by sniffing network traffic. Finally, all four are "spybot" or "adware" class programs, according to the classification in the previous section. Because such programs are typically packaged with popular free software, it is particularly easy for an unwitting user to unknowingly install them. For each of the four programs, we give an overview of how they function and what kinds of information they collect.

These four spyware programs each send and retrieve information from remote servers using the HTTP protocol. Because of this, we were able to derive signatures that detect and identify spyware programs operating on remote computers using passive network monitoring. Our signatures are based on two components: lists of servers that each spyware program could potentially communicate with, and HTTP signatures that distinguish spyware activity from human-generated Web browsing activity to those servers. For us to classify a Web request as originating from a particular spyware program, the web request must go to a server associated with that program, and the request must match the HTTP signature associated with that program.

To construct the list of servers with which the spyware programs communicate, we identified all external servers whose DNS name belongs to one of the spyware companies' domain names, or whose IP address belongs to an address prefix allocated to the spyware company according to the ARIN and RIPE registries. Our lists of DNS names associated with spyware servers have 44 entries for Gator, 18 for Cydoor, 12 for SaveNow and 2 for eZula. Our lists of IP address prefixes associated with spyware servers have 4 prefixes for Gator, 9 for Cydoor, 2 for SaveNow, and 1 for eZula. Appendix A presents the server lists and HTTP signatures we used for these programs in full detail.

3.1 Gator

Gator is adware that collects and transmits information about a user's Web activity. Its goal is to gather demographic information and generate a profile of the user's interests for targeted advertisements. Gator may log and transmit URLs that the user visits, partially identifying information such as the user's first name and zip code, and information about the configuration and installed software on the user's machine. Gator also tracks the sites that a user visits, so that it can display its targeted ads at the moment that specific words appear on the user's screen. Gator is also known as OfferCompanion, Trickler, or GAIN.

Gator can be installed on a user's computer in several ways. When a user installs one of several free software programs produced by Claria Corporation (the company that produces Gator), such as a free calendar application or a time synchronization client, the application installs Gator as well. Several peer-to-peer file-sharing clients, such as iMesh [8], Grokster [7], or Kazaa [9], are bundled with Gator. When visited, some Web sites will pop up advertisements on the client's browser that prompt the user to download software that contains Gator. Gator can run either as a DLL linked with the free software that carries it, or within a process of its own launched from an executable called *gain.exe* or *cmesys.exe*. Gator is capable of self-updating.

A rudimentary mechanism to "de-fang" spyware is to remap the DNS names of the spyware servers by adding entries to the client's *hosts.txt* file. By doing so, communication between the spyware client and server is disrupted. However, we observed that Gator inspects the *hosts.txt* file every time the client's computer is rebooted, and comments out any entries that refer to the gator.com domain. Additionally, Gator caches the IP addresses of gator.com DNS names, making it immune to further changes to *hosts.txt*.

3.2 Cydoor

Cydoor displays targeted pop-up advertisements whose contents are dictated by the user's browsing history. When a user is connected to the Internet, the Cydoor client prefetches advertisements from the Cydoor servers. These advertisements are displayed whenever the user runs an application that contains Cydoor, whether the user is online or offline. In addition, Cydoor collects information about certain Web sites that a user visits and periodically uploads this data to its central servers. When a user first installs a program that contains Cydoor, the user is prompted to fill out a demographic questionnaire, the contents of which is transmitted to the Cydoor servers.

Cydoor Technologies (the company that produces Cydoor software) offers a freely downloadable Software

Development Kit (SDK) that can be used to embed the Cydoor DLL in any Windows program, potentially generating advertisement revenue for the program's author. Removing the Cydoor DLL can cause the program that contains it to break.

3.3 SaveNow

SaveNow monitors the Web browsing habits of a user and triggers the display of advertisements when the user appears to be shopping for certain products. While SaveNow does not appear to transmit information about the user's behavior, it does use collected information to target its advertisements. SaveNow will periodically contact external servers in order to update its cached advertisements and its triggers, and to update the executable image itself (*save.exe*). Today's most popular peer-to-peer file-sharing application, Kazaa, is bundled with SaveNow.

3.4 eZula

eZula attaches itself to a client's Web browser and modifies incoming HTML to create links to advertisers from specific keywords. When a client is infected with eZula, these artificial links are displayed and highlighted within rendered HTML. It has been reported that eZula can modify existing HTML links to redirect them to its own advertisers [21], but we have not observed this ourselves. eZula is also known as TopText, ContextPro or HotText. eZula is bundled with several popular file-sharing applications (such as Kazaa and LimeWire), and it can also be downloaded as a standalone tool. eZula runs as a separate process (*ezulamain.exe*) and it includes the ability to self-update.

3.5 Summary

Gator, Cydoor, SaveNow, and eZula vary significantly in functionality, infection mechanism, and the degree of risk they represent to affected users. Through a manual examination of these four programs, we characterized how they operate and we derived HTTP signatures (presented in Appendix A) that can be used to remotely detect infected hosts using passive network monitoring. In the next section of this paper, we use these signatures to measure the activity of spyware-infected hosts within the University of Washington.

4 Measurement and Analysis of Spyware

In this section, we present measurements and analysis of spyware activity at the University of Washington, a large public university with over 60,000 faculty, students and staff, gathered using a week-long passive network trace. We have two main goals: (1) to understand how widespread spyware is within the university, both at the granularity of individual clients and at the granularity of organizations (such as academic departments), and (2) to gain some insight into what kinds of user behavior are correlated with spyware.

	WWW	Gator	Cydoor	SaveNow	eZula
HTTP transactions	120,593,877	489,934	33,122	4,645	5,096
# of clients	31,303	1,077	399	406	63
# of servers contacted	989,794	67	22	3	2
# of orgs. observed	239	154	72	116	40
total bytes transferred	0.95 TB	0.80 GB	149 MB	2.4 MB	37.2 MB
average requests/min	11,964	44.8	3.29	0.46	0.51

Table 3. Trace statistics. Our trace was collected over a week-long period starting on August 26, 2003. "Organizations" refer to groups such as the Department of Physics.

4.1 Methodology

The University of Washington connects to its Internet Service Providers via two border routers. These two routers are connected to four gigabit Ethernet switches, each of which connects to one of four campus backbone links. The switches mirror both incoming and outgoing packets to our passive monitoring host over dedicated gigabit links. Peak bandwidth exchanged between the university and its ISP can reach approximately 800 Mb/s, though the average bandwidth we observed during the trace was 238 Mb/s. The campus contains between 40,000 and 50,000 hosts. Over the period of our trace, we observed 34,983 university IP that exchanged HTTP traffic with external hosts.

The monitoring host reconstructs TCP and HTTP streams from the mirrored packets and produces a log of HTTP activity. Both HTTP requests and HTTP responses are reconstructed; we use heuristics to reconstruct pipelined HTTP requests on persistent connections. All sensitive information, including IP addresses and URLs, is anonymized using keyed one-way hashing before being written to a log. Rather than recording the entire HTTP transaction in the log, our software extracts relavant features (such as source and destination IP addresses, URLs, and transfer lengths) from each request and records them in the log. Hardware counters on the mirroring switches reported 0.000616% packet drops, and the network interface card of the monitoring host showed no packet drops during our measurement interval. Software counters within the kernel packet filter of the monitoring host also reported no packet drops.

In order to preserve locality in anonymized IP addresses, we anonymize each octet of campus IP addresses separately. However, we also zeroed out the last two bits in the last octet of each campus IP address to make the anonymization more secure. We do record the organizational membership (such as individual academic departments and campus dormitories) for each anonymized IP address in the log. Throughout this paper, we identify clients and servers by their IP addresses; this has limitations which we will discuss below.

Our monitoring software classifies each HTTP request before anonymizing and logging it to disk. Any HTTP request to port 80, 8000, or 8080 is classified as WWW traffic. We use our previously derived HTTP signatures to identify traffic from Gator, Cydoor, SaveNow, and eZula within the set of all WWW requests. Finally, as a point of comparison, we identify Kazaa file-sharing transfers by looking for Kazaa-specific headers within all HTTP requests, regardless of the port at which they are directed.

Although our tracing software records all HTTP requests and responses flowing both in and out of our university, the data presented in this paper only considers HTTP requests generated from clients inside the university and the corresponding HTTP responses generated by servers outside the university. Our week-long trace was initiated on August 26, 2003. This trace period corresponds to summer break within campus, so we observed less traffic than when classes are in full session. Table 3 shows a summary of our trace statistics.

4.1.1 Assumptions and Limitations

Our methodology has several inherent limitations. Because we destroy two bits in each IP address during anonymization, we cannot uniquely identify an individual client by IP address alone: each anonymized IP address that appears in the trace log could correspond to four actual IP addresses. However, while collecting the trace, our software maintained counters of the correct number of unique non-anonymized IP addresses in each of the traffic categories in Table 3. Using these counters, we calculated the ratio of the correct number of non-anonymized IP addresses observed by our software to the number of anonymized clients appearing in our log for each traffic category. Whenever needed, we use these "IP address calibration ratios" to back-infer the correct number of IP addresses in a population subset. These ratios are 1.58, 1.05, 1.0, 1.0, and 1.0 for WWW, Gator, Cydoor, SaveNow, and eZula, respectively. All IP-address based population statistics presented in this paper are calibrated using this method.

Because DHCP is used to assign IP addresses in portions of our campus, our methodology of identifying clients by IP address is problematic, as over a sufficiently long time scale, many clients may share the same IP address and an individual client may use several different IP addresses. This "DHCP effect" has been noted in previous studies [3, 11]. To minimize the effects of DHCP, we chose to restrict our trace to a short period of time: one week. Additionally, we excluded the university's dial-up modem pools from our trace, since DHCP issues are particularly problematic for this subset of the university. As we will describe in Section 4.2.1, we were able to calculate the "true" number of Gator clients within the trace, regardless of the anonymization and DHCP issues, using a unique identifier that Gator happens to provide in some of its request headers.

Of the 1,027 active (anonymized) Gator IP addresses observed, we were able to observe Gator identifiers sent from 914 of them. The remainder of the Gator IP addresses did not exchange messages that happened to contain an identifier. From these 914 IP addresses, we counted 872 unique Gator identifiers. The "true" number of Gator clients (872) was therefore inflated by anonymization and DHCP effects to 914, a factor of 1.05. In the rest of this paper, if we quote a population size that is derived from counting Gator identifiers (as opposed to IP addresses), we will explicitly say so.

Another potential problem is that our spyware HTTP signatures could potentially miss some spyware traffic. The signatures distinguish normal user-generated HTTP requests to spyware companies' servers from spyware-generated requests using HTTP patterns. If our signatures happen to miss rarely occurring patterns, then we will underestimate the amount of spyware traffic and potentially the number of spyware-infected clients. Similarly, our spyware signatures filter out traffic based on our list of spyware servers; if this list is incomplete, then we will fail to detect additional spyware traffic. Given that we are only detecting the presence of four specific spyware programs out of the many hundreds that exist, our reported numbers should be considered as a conservative lower bound on the true impact of spyware within the university.

4.2 A Client View of Spyware

We begin by looking at how spyware has affected individual clients within the university. More specifically, we quantify the number of clients with spyware, the rate at which new installations occur, and correlations between various kinds of network activity and susceptibility to spyware installation.

4.2.1 The Spread of Spyware

Over the course of the week, 31,303 internal Web clients accessed 989,794 external Web servers (Table 3). A significant fraction of them, 3.4% (1,077 clients), had Gator installed, 1.3% had Cydoor installed, 1.3% had

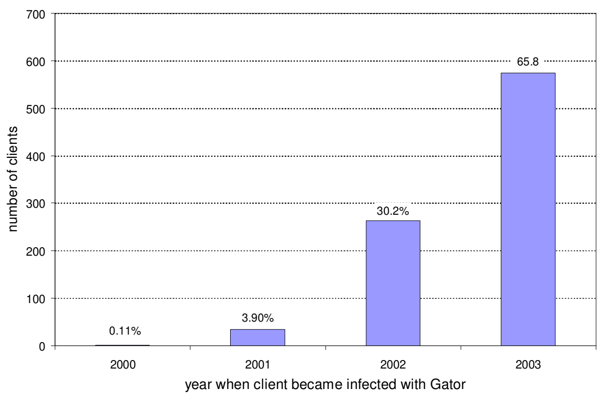

Figure 1. Year of infection for Gator clients. This graph shows the number of currently infected clients that became infected during each of the past three years.

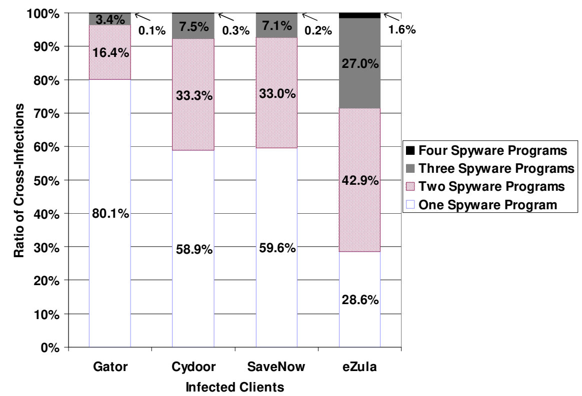

Figure 2. Multiple spyware infections among clients. For each of Gator, Cydoor, SaveNow, and eZula, this chart shows the fraction of infected clients that are infected with exactly one spyware program, two spyware programs, three spyware programs, and all four spyware programs.

SaveNow installed, and 0.2% had eZula installed. *In total, we found that 1,587 clients (5.1%) were infected with one or more spyware programs.*[3]

Although the fraction of active clients that are infected with these four spyware programs may appear to be small, we believe this number is disturbingly high, especially considering that we only measured four out of the hundreds of spyware programs that exist. For example, if a remote exploit exists in Gator, 3.4% of active clients in our university would be susceptible.

During our analysis, we discovered that when Gator is first installed on a client, it performs a series of distinct HTTP requests to "register" the Gator client with the Gator infrastructure. By monitoring these requests, we were able to measure the rate of new Gator installations within the university. Over the course of a week, we detected 52 new installations. Given this slow rate of infection and the fact that we detected 1,077 Gator clients, we hypothesize that many current Gator clients had Gator installed several months or years in the past.

Another fortuitous discovery allowed us to confirm this hypothesis. Many of the messages sent by a Gator client to Gator's central servers carry a timestamp that specifies the precise date of the initial installation. We confirmed that this timestamp survives Gator self-updates. We were able to discover the initial installation date for 872 out of the 1,077 Gator clients; the remaining clients never exchanged a message containing the timestamp. We also used this timestamp to uniquely identify Gator clients within our trace, as mentioned previously in Section 4.1.1.

Figure 1 shows the year of installation for these 872 clients. Over half (65.8%) of clients were installed in 2003, and approximately one third (30.2%) were installed in 2002. Gator has been present within our university for over three years, and one client that was infected in 2000 still remains infected today. Note that these numbers only indicate the number of Gator installations that are still observable today: they do *not* indicate the total number of Gator installations that happened each year. It is possible that other Gator clients exist but either left the university or removed Gator.

4.2.2 Modems Vs. Non-Modems

It is reasonable to expect that spyware will affect personally-owned computers more than university-owned computers, since people have greater freedom to install software on their own machine. To explore whether this is true, we measured the number of infections within the university modem pool to compare against the previously reported statistics for (non-modem) university hosts. Many people dial into the university modem pool from their personal machines. Since DHCP issues are especially problematic for modem pools, we focused on Gator, relying on Gator timestamps rather than IP addresses as unique identifiers within the modem pool.

As previously mentioned, we observed 872 Gator installations within 31,303 non-modem pool university hosts, counting using Gator timestamps. We observed 942 Gator installations in our modem pool (20 of which also appeared in the non-modem pool hosts) within the 12,435 accounts that logged into the modem pool at least once during our trace. Based on these timestamp statistics, 2.8% of non-modem pool hosts were infected with Gator, whereas 7.6% of modem-pool hosts were infected with Gator. Spyware does appear to be more prevalent on personally-owned computers, but it also has a significant presence on university-owned computers.

[3] Note that we could not measure spyware installations on computers that were inactive during our tracing period, so this number is conservative.

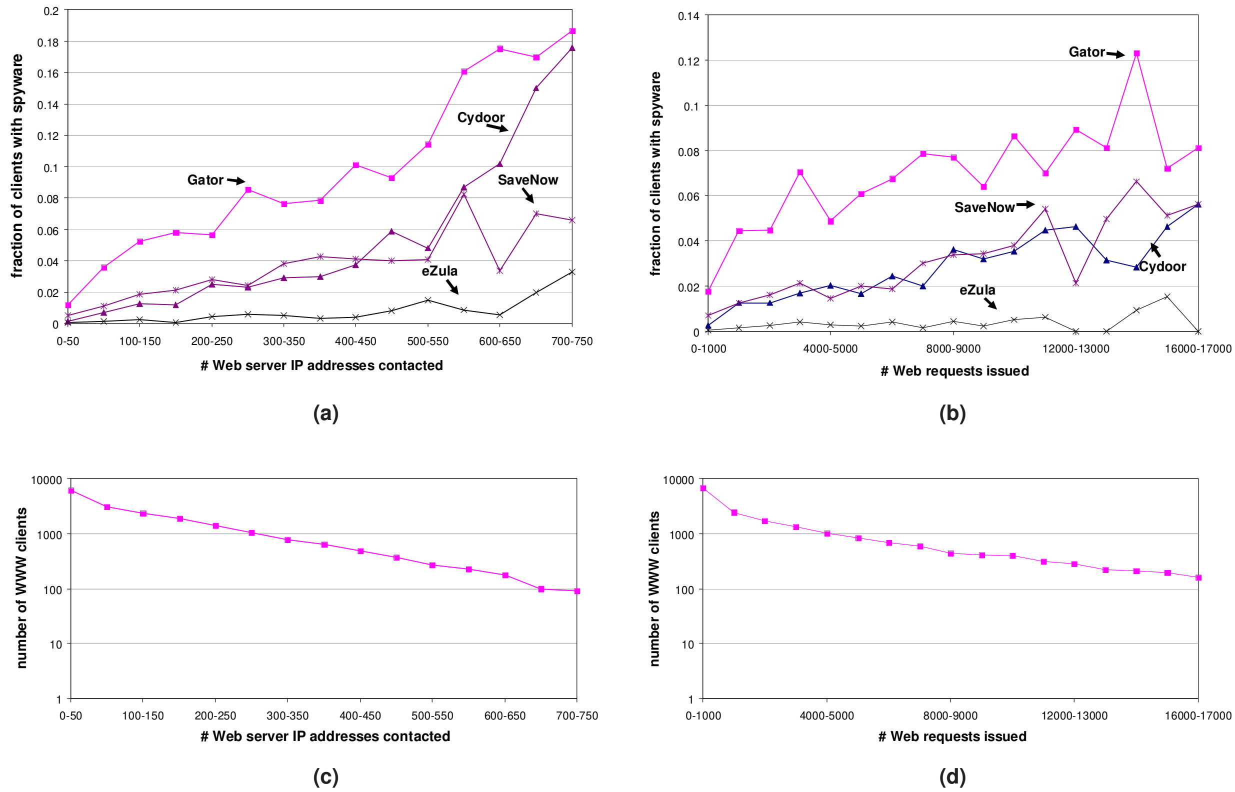

Figure 3. Spyware infections as a function of Web activity. (a) The fraction of WWW clients infected with at least one spyware program as a function of the number of external Web server IP addresses contacted during the week-long trace. (b) The fraction of WWW clients infected with spyware as a function of the number of Web requests issued during the trace. (c) The number of WWW clients (infected or not) as a function of external Web server IP addresses contacted. For example, the point (200-250,1397) shows that there were 1397 Web clients that contacted at least 200 Web servers, but less than 250 Web servers. (d) The number of WWW clients (infected or not) as a function of Web requests issued.

4.2.3 Cross-Infection Rates

The data presented above showed that there were 1,945 spyware infections within the traced (non-modem) population, but only 1,587 computers were infected with spyware. Therefore, many clients must be infected with more than one spyware program. Figure 2 shows, for each of the four spyware programs we detected, what fraction of clients were also infected with other spyware programs. For example, consider the set of clients with Gator. Of these, 80.1% contain only Gator, 16.4% of them contain Gator and one other spyware program, 3.4% of them contain Gator and two other spyware programs, and 0.1% contain all four spyware programs.

In contrast, just 28.6% of eZula clients are infected with only eZula. This suggests that whatever causes eZula infections also causes infections of other spyware programs. Our data shows that many clients infected with spyware are infected with more than one kind of spyware.

4.2.4 Correlating Spyware Infections with Client Behavior

There are many activities that may increase a client's potential exposure to spyware. For example, visiting a large number of Web servers increases a client's likelihood of encountering a spyware-infested Website. As another example, downloading and installing executables off of the Internet may cause a client to unwittingly install spyware. Similarly, installing and running peer-to-peer file-sharing software leads to spyware infections, since file-sharing software often contains bundled spyware. Our trace enables us to examine the correlation between these activities and the fraction of clients with spyware. It is important to note that our results identify correlation, but not necessarily causation.

Ideally, we would restrict our analysis to the behavior of clients at the approximate time when they install spyware. However, since only 52 new Gator installations occurred during the trace, we did not have an adequately large sample of client behavior at the time of new installations. Instead, we considered the behavior of

	number of clients	fraction with Gator	fraction with Cydoor	fraction with SaveNow	fraction with eZula	fraction with any spyware
no EXEs	20,630	0.875%	0.528%	0.776%	0.048%	1.953%
≥1 EXEs	10,673	8.41% [9.6x no EXEs]	2.72% [5.2x no EXEs]	2.31% [3.0x no EXEs]	0.497% [10x no EXEs]	11.1% [5.7x no EXEs]
≥10 EXEs	2,560	10.5% [12x no EXEs]	5.08% [9.6x no EXEs]	3.09% [4.0x no EXEs]	0.94% [20x no EXEs]	14.7% [7.5x no EXEs]
all	31,303	3.44%	1.28%	1.30%	0.20%	5.07%

Table 4. Spyware infections as a function of downloaded executables. Like Web activity, the number of downloaded executables seems to be correlated with the fraction of clients infected with spyware. Clients that downloaded no executables during the trace had a lower fraction of spyware infections than clients that downloaded multiple executables.

all infected clients, regardless of when the infection took place. This means that our correlations compare activity during the traced time period to infections that may have occurred weeks, months, or years in the past.

Web activity: In Figures 3(a) and (c), we cluster clients into sets according to the number of different Web server IP addresses they contact during our week-long trace. For each cluster shown on the X-axis, we plot the fraction of infected Web clients (Figure 3a) and the total number of Web clients, infected or not (Figure 3c). Figures 3(b) and (d) show similar graphs, except that we cluster clients according to the number of Web requests they issue during the week.

These graphs demonstrate that there is a higher incidence of spyware infections within the clusters of clients with more Web activity. The set of clients that communicated with between 100 and 150 Web servers had a 5.2% Gator infection rate; the set of clients that communicated with 600-650 servers had a 17.5% Gator infection rate. Similarly, the set of clients that issued fewer than 1000 Web requests within the week-long trace had a 1.8% Gator infection rate; those that issued between 12,000 and 13,000 requests had an 8.9% Gator infection rate.

Downloading executables: Downloading executable code from the Internet may expose a client to spyware. Because of this, we expected to see a greater incidence of spyware infections among clients that download many executables.

Table 4 shows the fraction of infected clients as a function of the number of executables they downloaded during our week-long trace. For example, clients that downloaded no executables had a 0.875% Gator infection rate, clients that downloaded 1 or more executables had a 8.41% Gator infection rate, and clients that downloaded 10 or more executables had a 10.5% Gator infection rate. Downloading executables appears to be correlated with higher spyware infection rates.

Using peer-to-peer file-sharing programs: Peer-to-peer file-sharing programs such as Kazaa often ship with bundled spyware. Analysis of our trace revealed that 38% of clients that issued at least one Kazaa request were infected by spyware: 17% of such clients contain Gator, 28.2% contain Cydoor, 8.1% contain SaveNow, and 1.7% contain eZula. These percentages are 5x to 22x times greater than the corresponding infection ratios of Web clients (as reported in Table 3), confirming the intuition that using file-sharing software exposes clients to spyware. However, Kazaa is not the only way clients are exposed to spyware: 62% of clients infected with spyware issued no Kazaa requests during our trace.

4.3 Spyware Bypasses Today's Security Infrastructure

Our university consists of several hundred individual organizations, including academic departments, dormitories, sporting facilities, medical clinics, and many others. Though the core networking infrastructure of the university is centrally managed, each organization is responsible for managing its own end systems and enforcing its own security policies. Some organizations have perimeter firewalls, while others do not. Many organizations centrally manage desktop configurations and are vigilant about installing security patches and anti-virus software updates, while others provide little or no support and have no explicit security policy. Each organization within the university therefore can be considered to be its own independent trust domain, with its own set of defenses against threats and intrusions.

Our network monitor is able to classify network traffic according to these organizational boundaries. Using this classification, we calculated the fraction of organizations that contain one or more spyware-infected hosts. The results are discouraging: 69% of organizations have at least one host infected with at least one variety of spyware. 64% of organizations have Gator infections, 30% have Cydoor, 49% SaveNow, and 17% eZula. Spyware has managed to penetrate most organizations' boundaries, regardless of their security policies. Perimeter protection mechanisms such as firewalls are not helpful, since most spyware infections occur with the cooperation of internal users, whether this cooperating is willingly or unwittingly given.

If a spyware infection leads to a compromise (whether because of a vulnerability in the spyware itself or because of a deliberate backdoor), then an attacker will gain control of a machine inside the organization's trust boundary. As one way of gaining further insight into this issue, we gathered a list of the top one hundred most popular Web servers within the university, ranked according to the number of requests served. Next, we identified which of those Web servers have a Gator client resident on the same /24 subnet (i.e., the IP addresses of the

Web server and the Gator client have the same first three octets). Forty-seven of these Web servers share a subnet with a Gator client, as do two of the top ten. Though some Web servers are isolated from potentially susceptible hosts, many are not.

4.4 Summary

Using passive network monitoring, we have demonstrated that spyware is widespread within the university. More specifically, our results show that:

- at least 5.1% of hosts within the university have been infected with spyware;
- some infected hosts have remained infected for several years;
- many infected hosts contain multiple kinds of spyware;
- the subsets of the university population that use the Web heavily, use peer-to-peer file-sharing software, or download many executable programs tend to have a greater fraction of infected hosts;
- spyware infections span most of the organizations within the university.

The "spyware problem" is significant in scope. In the following section, we discuss security implications of spyware, and we attempt to extrapolate our university-local results to the Internet at large.

5 Discussion

The previous section of the paper demonstrated that spyware is widespread in our university. Spyware can be an inconvenience to infected users, but we argue that it also has significant local and global security implications. As we have shown, spyware exists within most organizations in our university, and therefore has penetrated organizations' security mechanisms. If a widespread spyware program has a vulnerability, then attackers might be able to compromise a significant fraction of machines and penetrate most organizations in the university.

In this section, we describe vulnerabilities that we found in Gator and eZula. Although exploiting them requires the attacker to be able to eavesdrop on and spoof network traffic, it serves to demonstrate that spyware programs do have security weaknesses in practice. After describing the vulnerability, we conservatively estimate how many spyware infections exist within the Internet at large by back-projecting from our university-local results.

5.1 Vulnerabilities in Gator and eZula

Gator and eZula installations consist of both code (e.g., DLLs) and data (e.g., a database of keywords or URLs). Both programs contain self-update mechanisms which allow them to download updates to code or data from a central Website. Upon examination, we found that both Gator and eZula suffered from a simple vulnerability in how they install data file updates.

To update data files, Gator and eZula download compressed archives from their central Websites. The archives are retrieved from URLs that include fully qualified domain names, and therefore the programs issue DNS requests to determine the IP addresses of the Web servers to contact. After downloading a compressed archive, Gator and eZula decompress it and extract the archived files into the local filesystem.

Unfortunately, neither program verifies the authenticity or integrity of a downloaded archive before extracting files from it. Given this, if an attacker can hijack the download TCP connection or spoof *gator.com* or *ezula.com* DNS responses, the attacker can cause a victim to download and install an archive of his choosing. By constructing an archive that contains files with absolute or relative paths in their names, the attacker can place a file in a targeted place within the victim's filesystem. For example, the attacker could place an executable in the "Startup" directory of a user's account by constructing an archive that contains a filename including a path to that directory. While this vulnerability is more difficult to exploit than a buffer overrun vulnerability, it is evidence that spyware programs can and in some cases do contain security flaws. We are not alone in finding such problems: at least one other vulnerability has previously been found in Gator [4].

We implemented and successfully mounted an attack by sending spoofed DNS responses to *gator.com* and *ezula.com* DNS requests coming from infected clients. Our spoofed responses trick the spyware programs into downloading and installing updates that we supply from a local Web server, instead of downloading updates from the intended servers. We verified that we could insert arbitrary executables in our updates, leaving open the possibility of running malicious code or installing backdoors. Though we restricted our testing to victim machines we control, our attack could in practice affect arbitrary machines whose network traffic we can monitor and spoof.

We reported the security vulnerabilities we discovered to the companies that produce Gator and eZula. Claria Corporation (who produces Gator) created an updated version of their software in which the flaw was repaired. To the best of our knowledge, at the time this paper was written, eZula had not yet addressed the vulnerability.

5.2 Estimating the Spread of Spyware on the Internet

Our results only measure the number of spyware clients within our university. Though we expect these numbers to be representative of many organizations besides ours, we wanted to find a way to estimate the extent of the spyware problem on the Internet at large. To do so, we rely on two facts: Kazaa file-sharing software contains embedded spyware (Table 2), and at least 38% of active Kazaa peers within our university are infected with spyware (Section 4.2.4). Using these facts, it seems reasonable to use the presence of Kazaa as an indicator of the presence of spyware.

We have found three different ways to estimate the number of Kazaa installations on the Internet.

- Several Websites maintain counters of the number of active Kazaa users at any given time [16]; these sites generally report that the Kazaa network consists of around 4 million concurrent clients at most times. Using our 38% infection rate, we estimate that there are 1.5 million spyware-infected hosts active on the Kazaa network alone.

- Our measurement infrastructure allows us to identify external Kazaa peers that exchange content with university peers. Using a previously gathered trace, we counted 6,811,743 external Kazaa IP addresses over a 7 month period. This number is likely to be a lower bound on the number of actual Kazaa peers, since only a subset of global Kazaa peers ever contact our university, but using it, we estimate that there are at least 2.6 million spyware-infected Kazaa hosts. Furthermore, these external Kazaa IP addresses spanned over 397,000 external /24 subnets.

- At a different university, a similar study [18] captured 9 million distinct external Kazaa IP addresses interacting with internal hosts. Using this as an estimate, there are at least 3.4 million Kazaa hosts infected with spyware.

All of these estimates confirm that the spyware problem is of significant scope in the Internet at large.

6 Related Work

While we know of no other academic studies on spyware, several commercial efforts have attempted to characterize spyware [17], implement spyware detection removal tools using host-based signatures [1, 17], and estimate the spread of spyware within a customer population [14].

Several previous studies quantified the extent of related Internet security threats, such as self-propagating worms. In 2001, Moore et al. [11] found that more than 359,000 computers became infected with the Code-Red worm in less than 14 hours. In 2003, Moore et al. [10] found over 75,000 hosts infected by the Slammer worm. In this paper, we have demonstrated that spyware affects a similarly large number of hosts in the Internet, and that the existence of vulnerabilities within it makes spyware a potential threat of comparable size and scope.

Intrusion detection systems (IDSs) are a commonly used tool for the prevention and detection of Internet security threats. These systems attempt to identify known attacks, either by monitoring network activity in the case of network-based IDSs [13], or by monitoring host activity in the case of host-based IDSs [5]. The techniques developed for intrusion detection systems may be applicable to the problem of identifying spyware infections. The fact that we were able to derive signatures for passively detecting spyware traffic suggests that this problem is tractable.

A related problem to detecting infections is preventing damage from infections. Many code isolation and sandboxing techniques are potentially applicable, including virtual machines [20, 19], resource containers [2], or system-call sandboxes [6].

7 Conclusions

This paper demonstrates that spyware infections are widespread among hosts in the University of Washington. Our results show that the "spyware problem" is of large scope, and as a result, spyware has significant local and global security implications for today's Internet.

After presenting background material on spyware, we analyzed four specific spyware programs (Gator, Cydoor, SaveNow, and eZula), describing how they function and deriving network signatures that can be used to detect infected remote hosts. Using these signatures, we gathered a week-long trace of network activity at our university, and we used this trace to quantify the spread of spyware on campus.

Our results show that spyware infects at least 5.1% of active hosts on campus, and that many computers tend to have more than one spyware program running on them. We also show that 69% of organizations within the university (e.g., academic departments) contain spyware hosts, suggesting that security practices on campus are not effective at preventing spyware infections.

A vulnerability in a widespread spyware program would potentially put a large number of hosts within the university and in the Internet at risk. We discovered and described a specific vulnerability in Gator and eZula: the potential for spyware to cause substantial security problems is real.

7.1 Acknowledgements

We would like to thank Eric Schwimmer, from whom we initially learned of the prevalence of spyware in peer-to-peer file sharing software. We would also like to thank David Richardson, Ville Aikas, Art Dong and the other members of the Computing and Communications organization at the University of Washington for their support. We are grateful for the guidance of Vern Paxson, our shephard, and our anonymous reviewers. We thank Richard Dunn, Krishna Gummadi, David Wetherall, and John Zahorjan for their helpful comments on this research. This research was supported in part by the National Science Foundation under Grants ITR-0121341 and CCR-0085670, and by a gift from Intel Corporation.

References

[1] Ad-Aware. `http://www.lavasoftusa.com/software/adaware/`.

[2] G. Banga, P. Druschel, and J. Mogul. Resource containers: a new facility for resource management in server systems. In *Proceedings of the 3rd Symposium on Operating System Design and Implementation (OSDI 1999)*, New Orleans, LA, February 1999.

[3] R. Bhagwan, S. Savage, and G. Voelker. Understanding availability. In *Proceedings of the 2nd International Workshop on Peer-to-peer Systems*, Berkeley, CA, December 2002.

[4] EyeOnSecurity. `http://eyeonsecurity.org/advisories/Gator/`.

[5] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for unix processes. In *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy*, Oakland, CA, May 1996.

[6] I. Goldberg, D. Wagner, R. Thomas, and E. A. Brewer. A secure environment for untrusted helper applications: confining the wily hacker. In *Proceedings of the Sixth USENIX Security Symposium*, San Jose, CA, July 1996.

[7] Grokster. `http://www.grokster.com`.

[8] iMesh. `http://www.imesh.com`.

[9] Kazaa. `http://www.kazaa.com`.

[10] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer worm. *IEEE Security and Privacy*, 1(4):33–39, July 2003.

[11] D. Moore, C. Shannon, and J. Brown. Code-red: a case study on the spread and victims of an internet worm,. In *Proceedings of the 2002 ACM SIGCOMM/USENIX Internet Measurement Workshop*, Marseille, France, November 2002.

[12] S. Olsen. Software replaces banner ads on top sites. C|Net News.Com article, August 2001.

[13] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[14] PC Pitstop. PC Pitstop spyware statistics. `http://www.pcpitstop.com/research/spyware.asp`, January 2003.

[15] J. Schartz. "Acquitted man says virus put pornography on computer". New York Times article, August 2003.

[16] Slyck. `http://www.slyck.com`.

[17] SpyBot S&D. `http://security.kolla.de`.

[18] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 2002 USENIX Security Symposium*, San Francisco, CA, August 2002.

[19] J. Sugerman, G. Venkitachalam, and B. Lim. Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor. In *Proceedings of the 2001 Annual USENIX Technical Conference*, Boston, MA, USA, June 2001.

[20] A. Whitaker, M. Shaw, and S. D. Gribble. Scale and Performance in the Denali Isolation Kernel. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, December 2002.

[21] J. Wilson. How TopText works. `http://scumware.com/wm2.html`.

A Spyware Signatures

To identify spyware traffic using our passive network monitor, we derived network signatures for each spyware program we wanted to identify. We classify a web request as originating from spyware if the request matches both of the following signature components:

- **Server list:** the web request must originate from a client within the University of Washington, and be directed at a server in a "server list." For each spyware program, we constructed a list of IP addresses and DNS names associated with servers known to be operated by the spyware program's company.
- **HTTP signature:** the web request must contain some HTTP signature that the spyware program includes in its requests, but which web browsers are unlikely to generate. This signature may be the value of the HTTP "User-Agent" field, or it may be a URL pattern.

In this appendix, we list the signatures we derived for Gator, Cydoor, SaveNow, and eZula.

A.1 Gator

Server list:

```
autoupdate.balance.gator.com
bannerserver.balance.gator.com
bannerserver.gator.com beasley.gator.com
```

```
bg.gator.com content.balance.gator.com
coupons.gator.com dns-a.gator.com
dns-cw.gator.com dns.gator.com
dns2.gator.com gator.com gator29.gator.com
gatorcme.gator.com gi.balance.gator.com
gi.gator.com gs.balance.gator.com
gs.gator.com gw-rwc.gator.com
images.gator.com jeeves.balance.gator.com
map.gator.com outsidedns.gator.com
patchserver.balance.gator.com
pricecomparison.gator.com rs.gator.com
search.balance.gator.com search.gator.com
scriptserver.gator.com ss.balance.gator.com
ss.gator.com ssbackup.balance.gator.com
surveys.balance.gator.com trickle.gator.com
trickle.balance.gator.com ts.gator.com
updateserver.gator.com wb.gator.com
web.balance.gator.com webpdp.gator.com
webpdp.balance.gator.com www.gator.com
xmlsearch.balance.gator.com xmlsearch.gator.com
```

```
63.197.87.0/24 64.94.89.0/24 64.152.73.0/24
66.35.229.0/24
```

HTTP signature: The Gator program uses the custom User-Agent HTTP header "Gator/x.xx", where "x.xx" is the version number of the Gator client.

A.2 Cydoor

Server list:

```
www.bns2.net www.bns1.net
www.rgs1.net www.rgs2.net
www.cms1.net www.cms2.net
cydoor.com www.cydoor.com
globix.alteon.cydoor.com
globix.alteon2.cydoor.com
jcms.cydoor.com jbns.cydoor.com
jbn2.cydoor.com jbns2.cydoor.com
jbnss.cydoor.com jmbns.cydoor.com
jmcms.cydoor.com sprint.alteon1.cydoor.com
```

```
63.170.89.0/24 209.10.17.128/25 209.73.225.0/24
209.11.66.0/24 209.11.84.130/32 209.11.84.135/32
209.11.84.137/32 209.11.84.138/32
209.11.84.139/32
```

HTTP signature: To detect Cydoor, we use specific keywords in the URL of the HTTP requests. In particular, we identify as Cydoor traffic any request to a server in the server list that contains a URL whose prefix is "/bns" or "/scripts," or a URL containing the string "javasite." The "bns" keyword refers to requests for pop-up advertisements. The "scripts" and "javasite" keywords refer to scripts that are used to collect information from users (note that such information is obfuscated but not encrypted).

A.3 SaveNow

Server list:

```
app.whenu.com chromium.whenu.com
iron.whenu.com lead.whenu.com
mercury.whenu.com oxygen.whenu.com
tin.whenu.com titanium.whenu.com
web.whenu.com whenushop.whenu.com
helium.whenu.com zinc.whenu.com
```

```
209.11.45.128/27 209.73.202.0/27
```

HTTP signature: Similar to Cydoor, we use specific keywords in the URLs within requests to SaveNow servers to detect SaveNow client activity. Specifically, we identify as SaveNow traffic any URL whose prefix is "/offer," "/heartbeat," or "/about." The "offer" keyword is usually followed by a list of parameters denoting a Website visited or a keyword entered by user within a form. These are presumably used for determining which advertisement should be displayed to the client. The "heartbeat" keyword is also followed by a list of parameters indicating a name of a program, a "partner code" and an "id code." We assume that this is a heartbeat mechanism for SaveNow that identifies which program is running SaveNow. The "about" keyword also includes a list of strings that also appear to be used to select advertisements that are displayed to the client.

A.4 eZula

Server list:

```
app.ezula.com ezula.com
```

```
208.185.211.64/26
```

HTTP signature: We use the HTTP User-Agent field to identify eZula spyware traffic sent to eZula servers. The eZula spyware program uses three specific User-Agent fields: "eZula," "mez," or "Wise," depending on the specific program version.