Automated Spyware Collection and Analysis

Andreas Stamminger, Christopher Kruegel, Giovanni Vigna[1]
and Engin Kirda[2]

[1] University of California, Santa Barbara
{as,chris,vigna}@cs.ucsb.edu
[2] Institut Eurecom, France
kirda@eurecom.fr

Abstract. Various online studies on the prevalence of spyware attest overwhelming numbers (up to 80%) of infected home computers. However, the term spyware is ambiguous and can refer to anything from plug-ins that display advertisements to software that records and leaks user input. To shed light on the true nature of the spyware problem, a recent measurement paper attempted to quantify the extent of spyware on the Internet. More precisely, the authors crawled the web and analyzed the executables that were downloaded. For this analysis, only a single anti-spyware tool was used. Unfortunately, this is a major shortcoming as the results from this single tool neither capture the actual amount of the threat, nor appropriately classify the functionality of suspicious executables in many cases.
For our analysis, we developed a fully-automated infrastructure to collect and install executables from the web. We use three different techniques to analyze these programs: an online database of spyware-related identifiers, signature-based scanners, and a behavior-based malware detection technique. We present the results of a measurement study that lasted about ten months. During this time, we crawled over 15 million URLs and downloaded 35,853 executables. Almost half of the spyware samples we found were not recognized by the tool used in previous work. Moreover, a significant fraction of the analyzed programs (more than 80%) was incorrectly classified. This underlines that our measurement results are more comprehensive and precise than those of previous approaches, allowing us to draw a more accurate picture of the spyware threat.

1 Introduction

In general, spyware is used to describe a broad class of software that is surreptitiously installed on a user's machine to intercept or take control over the interaction between the user and her computer. This broad definition includes programs that monitor a user's Internet surfing habits, but might also apply to software that redirects browser activity, commits click fraud, or downloads additional malware. Unfortunately, over time, the term spyware has become increasingly imprecise, and different companies or researchers often label the same program differently. In this paper, we use the term spyware in a more narrow

sense – as browser-based software that records privacy-sensitive information and transmits it to a third party without the user's knowledge and consent. This definition is more faithful to the "original" purpose of spyware, which is to record the activity of a user while surfing the web.

A host can become infected with spyware in various ways. For example, the spyware component might come bundled with shareware, such as a peer-to-peer client or a supposed Internet "accelerator." It is common practice that small software companies, unable to sell their products in retail, cooperate with spyware/adware distributors to fund the development of their products [1]. Most of the time, however, users have no choice to "unselect" the installation of the piggybacked nuisance without disrupting the desired software functionality.

In this paper, we are interested in the extent to which executables on the web present a spyware threat. This allows us to compare our results to a previous study [2]. For our analysis, we focus on spyware that uses Microsoft's Internet Explorer to monitor the actions of a user. Typically, this is done either by using the Browser Helper Object (BHO) interface or by acting as a browser toolbar. We feel that this focus is justified by the fact that the overwhelming majority of spyware uses a component based on one of these two technologies, a fact that is confirmed by a number of previous papers [3–6].

As mentioned above, the authors of a previous measurement study [2] attempted to assess the prevalence of spyware on the Internet. To this end, they crawled the web for executables, downloaded them, and installed the programs in an automated fashion. Then, the authors executed a single anti-spyware program, Lavasoft's Ad-Aware [7], to assess the threat level of each program.

Unfortunately, in the previous study, little attention was devoted to the fact that relying on the output and correctness of a *single* tool can significantly misrepresent the true problem, and thus, the perception of the spyware threat. Obviously, scanner-based systems cannot detect novel threats for which no signature exists. Also, such systems are often trivial to evade by using techniques such as obfuscation or code substitution. Hence, scanner-based systems may introduce false negatives, and as a result, cause the threat to be *underestimated*. However, it is also possible that a detection tool mislabels programs as being more dangerous than they actually are. Such false positives may cause an *overestimation* of the threat.

In our work, one of the aims was to show the bias that is introduced by deriving statistics from the results of a single tool. Obviously, we did not simply want to re-run the experiments with more anti-spyware tools (although we did employ a second, scanner-based application). Instead, we wanted to perform our analysis using substantially different approaches that aim to detect spyware. To this end, we checked for spyware-related identifiers in the Windows registry, using a popular, publicly-available database [8]. Moreover, we employed a behavior-based approach [3] that monitors the execution of a component in a sandbox and checks for signs of suspicious behavior. By combining multiple techniques and employing further, manual analysis in cases for which different tools disagree, we aimed to establish a level of "ground truth" for our sample set. Based on this

ground truth, we identify the weaknesses of each technique when exposed to a large set of real-world, malicious code.

In summary, the contributions of this paper are the following:

- In about ten months, we crawled over 15 million URLs on the Internet and analyzed 35,853 executables for the presence of spyware components.
- We employed three different analysis techniques and devoted additional manual effort to identify the true nature of the components that we obtained. This allows us to expose the weaknesses of individual analysis approaches.
- We compare our results to a previous study that attempted to measure the spyware threat on the Internet and critically review their results.

2 Methodology

In this section, we describe our approach to analyze the extent of spyware on the Internet. In order to keep a consistent terminology within the rest of the paper, we first define the behavior that constitutes spyware activity. Then, we explain how we crawl the web for executables and briefly discuss our approach to automatically install these programs. Finally, we describe how we identify a program as spyware,

As mentioned previously, the term spyware is overloaded. For example, it is not uncommon that a component that displays advertisements is considered spyware, even it does not read nor leak any privacy-related information. Other examples of mislabeled spyware are toolbars that provide search fields that send input to a search engine of the user's choice. Clearly, information that is entered into the search field is forwarded to the search engine. Hence, the component is not malicious, as it informs the user where the data is sent to.

Because of the ambiguous use of the spyware term, it is possible that the actual risk of downloading a spyware-infected executable is overstated. Consequently, we need a more precise discrimination between different classes of activity. As mentioned previously, we focus in our study on browser extensions (BEs) for the Microsoft Internet Explorer (from now on, we use the term browser extension to refer to both BHOs and toolbars). To make our discussion of browser extensions more precise, we propose the following taxonomy:

Benign. An extension is benign when it does not perform any function that might be undesirable from a privacy-related point of view, nor exposes the user to unwanted content.

Adware. Adware is benign software with the purpose of advertising a certain product, e.g., via pop-up windows. These components do not leak any sensitive information, though.

We also consider a toolbar as adware when it provides a search field to the user that sends the input to a particular (often, less well-known) search engine. The reason is that the toolbar promotes, or advertises, the use of a particular search engine. Of course, the user is free to use the toolbar or not.

Grayware. Grayware occasionally performs actions that send sensitive data to third parties in a way that is not completely transparent to users, especially inexperienced ones.

An important class of grayware are browser hijacker components. A browser hijacker is software that modifies the default browser behavior. Depending on the resource that is controlled, we distinguish between different types of hijackers. A *homepage hijacker* modifies the default home page that is displayed when the browser is launched. A *search hijacker* modifies the default search engine of the browser. It allows the user to enter keywords directly into the browser's address bar without the need to request the website of a search engine. Typically, the user is redirected to a less popular search engine with sponsored results. Similarly, a *error page hijacker* causes the browser to display a particular web site whenever a misspelled URL is entered into the address bar. Usually, the original URL is passed as a query parameter to this web site. While a hijacker component might appear to be a useful feature, it is also profitable for the author of the landing site. This is for two reasons: First, it increases the hit count for his site (which drives up advertising revenue) and second, it reveals popular URL misspellings to facilitate domain squatting. Since a hijacker component is not triggered for regular pages that are visited, it is not a means to capture *all* of the user's surfing activity. Also, an alert user can notice the modified browser behavior and reset it accordingly. These are the two differences that distinguish grayware from spyware.

Spyware. Spyware, as defined in this paper, serves the purpose of secretly and comprehensively collecting data about the user, such as her surfing habits or form inputs. The data collection process is invisible, and a significant amount of user data (for example, most or all of the visited URLs) are leaked to a third, untrusted party.

Malware. Some components are reported to perform actions that are typically associated with "regular" malware. An example are Trojan downloaders that run in the context of the browser when accessing malicious content on the Internet so that they can bypass personal firewalls. These components do not necessarily access private information, but perform clearly undesirable activity. For such components, we use the generic term malware.

It is possible that the same component implements functionalities that fall into different categories. For example, a spyware component could also display ads. In this case, the program is assigned to the category that captures the more malicious behavior (spyware, for this example).

2.1 Web Crawling

To find a representative amount of programs that install spyware components, we developed a fully automated system that crawls the web for potential candidates and downloads them. To this end, we make use of the Heritrix [9] web crawler, which can be easily extended and customized. For downloading interesting web

resources, we focus on binary content, such as executables or zip archives. Similar to [2], we identify such content by examining two properties for each candidate URL. If either (1) the URL's file extension is `.exe`, or (2) the "Content-type" HTTP header of the corresponding web resource is `application/octet-stream`, we download the file. We then check the first bytes of the file header and compare it with the "magic" value that denotes a Windows executable. We perform similar checks for zip, cabinet (`.cab`), and MS Installer files (`.msi`).

To determine whether Internet users with a specific field of interest are more likely to encounter spyware on the web, we defined eight categories, similar to [2]: adult, games, kids, music, desktop (office), pirate, shareware, and toolbar. For each category, we fed the Google search engine with category-specific keywords and used the fifty most relevant search results as a seed for our crawler. We consider this a reasonable approach, because these are the pages that users would most likely encounter when searching for content in the categories mentioned above. To focus our crawling to those web sites that are found by the Google search, we do a breath-first crawl only up to a depth of three links away from the seed.

2.2 Automatic Installation

To determine whether an executable contains spyware, we install it on a Windows guest system running on top of a Qemu virtual machine emulator [10]. Each executable is installed on a system that has been reverted to a known, clean state. Since we wish to analyze thousands of programs, the installation process has to be performed automatically. To this end, we had to find a way to simulate user interaction, which is typically necessary when navigating through Windows installation wizards that have a graphical user interface.

Once an executable is successfully installed, we have to determine whether a browser extension (BE) is present or not. Fortunately, this is quite straightforward. The reason is that, in order to be loaded on startup by the Internet Explorer, a BE must register its CLSID (i.e., Component ID) under a particular (directory) key in the Windows Registry. Thus, after each installation, we simply check for the presence of CLSIDs in this special directory. Note that it is difficult for a spyware to avoid setting this key, as the Internet Explorer would otherwise simply not load the BE at startup. We proceed with the subsequent analysis phase when any BE is identified.

2.3 Analysis

The purpose of the analysis phase is to determine whether a BE is malicious or not. More precisely, we attempt to classify each browser extension based on the taxonomy introduced previously. We use three different approaches to determine the type of a BE: an identifier-based mechanism, two scanner-based tools, and a behavior-based technique. They are discussed in more detail below.

Identifier-based Detection. The identifier-based detection relies on the value of the CLSID of the BE component. Interestingly, many spyware programs use

the same CLSID to register their component (possibly because the developers were lazy or use the same code base). Thus, the value of the identifier can provide some insight into the nature of the corresponding program. Moreover, also the file name of the extension component can be revealing. Of course, both identifiers can be easily modified by miscreants.

CastleCops [8] is[3] a community of security professionals that provides a free and searchable database of BHOs and toolbars. At the time of writing, it contained 41,144 entries. For each BE, the database lists various information, including the BE's CLSID and its file name. Furthermore, a classification is provided. This classification includes *X* for spyware and malware, *O* for programs that are open to debate (such as grayware and adware), and *L* for legitimate items.

To perform identifier-based detection, we use HijackThis [11], a free utility that scans a computer for installed browser extensions, reporting both the CLSIDs and path names of the identified components. Based on the information provided by HijackThis, we consult the CastleCops database. Using the classification provided by this database, we can classify the browser extension accordingly. The absence of any entry results in the BE being classified as legitimate.

Scanner-based Detection. Our scanner-based detection was based on two commercial anti-spyware products, Ad-Aware [7] and Spybot [12] – both popular and well-known spyware scanners.

Ad-Aware uses a number of threat categories to specify the precise nature of a sample. During our analysis, we encountered the following categories:

- *Adware*: Programs displaying advertising on the user's computer, without leaking sensitive information.
- *Data miner*: Programs designed to collect and transmit private user information to a third party. This behavior may be disclosed to the user through to the EULA. This is the equivalent to our spyware definition.
- *Malware*: A generic category for harmful programs, equivalent to our malware class.

To ensure that we had the newest signatures, we always updated Ad-Aware's signature database before launching a scan. To check for suspicious code, we perform a *full system scan*. Once the tool is finished, we check the report for the presence of any component that is recognized as being suspicious. If this is the case, we record the corresponding threat category.

Spybot is a spyware scanner that attempts to detect threats on the user's computer by comparing registry entries and files against a database with signatures of well-known malware samples. This tool allows to choose the threat categories for which a user wants to check. For our study, we chose those categories that we assumed to be most-closely related to spyware: hijackers, keyloggers, malware, potentially unwanted programs, and spyware. After we run a

[3] Unfortunately, CastleCops has recently ceased its operations, but was still active while we performed our analysis.

system scan, Spybot lists each detected threat, without providing any further classification.

Behavior-based Detection. To perform behavior-based detection, we build upon an analysis tool that we obtained from the authors of [3]. This tool allows the identification of unknown browser components as spyware by dynamically observing their behavior. Specifically, the tool monitors the flow of sensitive information (such as the URL that a user visits or the content of the web pages that are loaded) as it is processed by the Internet Explorer and any loaded browser extension (BHOs and toolbars). Whenever it observes any leak of sensitive information on behalf of a BE, such as submitting data to a remote server, this BE is considered spyware. For its analysis, two types of sensitive data are considered:

- URLs that the browser navigates to, and
- the contents of web pages retrieved by the browser in response to browser navigation.

Whenever sensitive (tainted) information is written out on behalf of the monitored BE, this action is recorded as suspicious. Writing out information can refer to saving data in a file, but also considers the case when data is sent over a network socket. This allows one to identify two different kinds of suspicious behavior:

- *Browser hijackers (grayware)*: As mentioned previously, hijacker components modify the default browser behavior such that certain user input is redirected to particular web sites. This behavior is detected when search terms or malformed URLs are entered into the browser address bar and then leaked by the BE.
- *Spyware*: These programs are detected when URLs are secretly leaked to an entity outside the browser (such as a remote host or a local file).

The behavior-based analysis is dynamic. Hence, it is necessary to monitor the activity of a BE while the browser is used to surf the web. To perform the dynamic analysis in an automated fashion, we had to develop an additional tool that allows us to drive the browser and simulate a user surfing the web (while monitoring the activity of browser extensions). This tool interacts with the browser in three different ways: by entering data directly into the address bar, by filling out and submitting form fields, and by following links on web pages. This variety of actions should provide for the realistic simulation of a user that browses the web. Moreover, to trigger hijacker programs, the tool enters keywords directly into the browser's address bar and intentionally requests malformed web addresses.

To simulate a browsing session, we require a list of URLs that should be visited as well as a number of keywords that we can enter into form fields. Our list of URLs included various popular search engine sites, such as `google.com`, `yahoo.com`, and `altavista.com`. During our browsing session, we surfed these

sites and entered numerous keywords with the aim to "trigger" the spyware program to leak information to a remote server or redirect the browser to a different site. We compiled our list of keywords using Google HotTrends, selecting the most popular search terms. Besides search engine sites, we also entered some of these keywords directly into the browser's address bar. To trigger BEs that hijack error pages, we also entered misspelled URLs.

3 Results

In this section, we discuss the results of our measurement study. More precisely, we show the prevalence of spyware-infected executables for a number of different "regions" on the web. Moreover, we compare the effectiveness of different detection techniques, examining their strengths and limitations. In particular, we are interested in the possible bias that Ad-Aware introduces, since this was the sole tool used in a previous attempt to quantify the extent of spyware on the Internet [2]. Finally, we compare the findings in the previous study to the results of our analysis.

Table 1. Crawler results by file type.

	Win32 exec.	**Zip archive**	**MS install**	**Cabinet**
Files	29,104 (72.5%)	10,260 (25.6%)	425 (1.1%)	335 (0.8%)

3.1 Overall Results

We crawled the web for ten months (from January 2007 until the end of October 2007), visited over 15 million URLs, and found 35,853 executables. Table 1 shows the number of binary resources that we discovered, categorized by their file type. The vast majority of downloads were Win32 executables and Zip archives. As shown in Table 2, 9.4% of all executables were installing at least one BHO or toolbar. This underlines the popularity of these techniques. Each browser extension that we obtained during the ten month crawl period was analyzed using the three approaches described in the previous section. Then, based on the (often differing) results of the individual techniques, we performed manual analysis to obtain a "ground truth" for our data set.

To obtain ground truth, we inspected those BEs for which the analysis methods reported different results. The manual analysis was carried out by launching the sample in a virtual machine, manually monitoring its network traffic as well as other system modifications (e.g., created files or registry entries). Also, we performed more extensive web surfing. The recorded behavior was then compared to various, online malware descriptions, and, based on all available information, a final classification was assigned to each sample. Moreover, especially in cases

Table 2. Overall crawler results.

URLs crawled	executables found	executables w/ BEs	unique BEs
15,111,548	35,853	3,356 (9.4%)	512

where a particular BE was more popular (i.e., part of several executables), we contacted the developers of the anti-spyware products to resolve classification errors. Although small errors are clearly possible, we believe that we have established a solid data set of benign and malicious components that can meaningfully serve as ground truth for our evaluation.

Table 3. Overall analysis results.

executables w/ non-benign BEs	unique non-benign BEs	executables w/ spy/malware BEs	unique spy/malware BEs
2,384 (6.6%)	205 (40.0%)	117 (0.3%)	22 (4.3%)

Table 3 shows the overall analysis results. It can be seen that about 6.6% of all executables contain non-benign BEs. However, most of these programs belong to the adware category, while the fraction of executables that contain malicious components (spyware and malware) is significantly less - only 0.3% or 117 executables. This clearly underlines that the spyware threat might appear much more dramatic when the analysis does not distinguish precisely between non-invasive adware and malicious spyware. A breakdown of the non-benign BEs according to our taxonomy is presented in Table 4.

3.2 Distribution of Infected Executables

In this section, we discuss in more detail the prevalence of particular, malicious browser extensions, as well as their habitat (i.e., domains and regions on the web that are primary sources for these browser extensions).

Table 5 shows those ten browser extensions that we encountered most frequently in executables. Note that, for this table, we only consider grayware, spyware, and malware extensions. The reason for not considering adware is that we want to specifically focus on the more invasive, malicious programs. It can be seen that *NewDotNet* is by far the most popular component found by our crawler, being bundled with 197 executables. Most of these executables are peer-to-peer software (e.g., Limewire, Gnutella) and download accelerators. *NewDotNet* is an error hijacker that redirects URLs that cannot be resolved via DNS to various remote hosts, such as `r404.qsrch.net`. *Webhancer* is the most popular spyware component, and it is bundled particularly often with screensavers. In our

experiments, this component secretly recorded the URLs that were visited and forwarded them to `dr2.webhancer.com`.

Table 4. Non-benign BEs, by class.

class	# BEs	times observed
adware	162 (79.0%)	1,985 (83.3%)
grayware	21 (10.2%)	282 (11.8%)
spyware	18 (8.8%)	91 (3.8%)
malware	4 (2.0%)	26 (1.1%)

Table 5. Top 10 BEs - counting only grayware, spyware, and malware.

name	class	times observed
NewDotNet	grayware	197
Webhancer	spyware	60
P2P Energy	grayware	45
TR/Agent.A	malware	24
NavExcel	grayware	21
Acez.SiteError	grayware	6
Pal.PCSpy	spyware	6
ClickSpring	spyware	5
SmartKeyLogger	spyware	5
CasinoBar	spyware	2

In the next step, we analyzed the prevalence of malicious browser extensions based on the categories of the web sites that are serving them. As mentioned in Section 2.1, for finding sites to crawl, we seeded the Google searches with keywords that were chosen from eight categories (adult, games, kids, music, desktop (office), pirate, shareware, and toolbar). The results for the prevalence of non-benign components on pages of these categories are shown in Table 6. As the numbers demonstrate, we encountered spyware in all categories.

Before analyzing the results in detail, we conjectured that most spyware would be found on shareware or freeware sites. This is not only because of the large amount of executables hosted on those sites, but also because shareware is often offered together with dubious adware to finance its development. Our results confirm the initial intuition: The *shareware* category is not only the richest source for executables in general, but also holds the largest number of executables that install a BE. Interestingly, although over 15% of the shareware applications come with a non-benign BE, the actual fraction causing a spyware or malware infection is comparatively low (0.4%). The categories of the sites where BEs are most likely misused for malicious purposes are adult, desktop (office), and games, as indicated by the highest fraction of spyware BEs (last row in Table 6).

3.3 Detection Effectiveness

This section provides a detailed comparison between the ground truth and the results delivered by each detection technique that we used for our study. This allows us to identify interesting cases in which a certain technique is particularly effective or ineffective.

Table 6. Penetration of non-benign BEs across different web categories.

	adult	**games**	**kids**	**music**	**office**	**pirate**	**share**	**toolbar**
URLs (in K)	660	536	2,375	3,573	1,089	4,589	1,791	498
domains	790	1,678	1,821	1,662	1,911	3,795	3,298	2,087
executables	1,298	3,048	3,732	3,053	3,363	6,586	11,043	3,730
executables	49	85	278	273	59	143	2,270	199
w/ BEs	(3.8%)	(2.8%)	(7.4%)	(8.9%)	(1.8%)	(2.2%)	(20.6%)	(5.3%)
executables w/	30	14	158	163	31	81	1,825	82
non-ben. BEs	(2.3%)	(0.5%)	(4.2%)	(5.3%)	(0.9%)	(1.2%)	(16.5%)	(2.2%)
domains w/	16	9	48	56	26	44	88	39
non-ben. BEs	(2.0%)	(0.5%)	(2.6%)	(3.4%)	(1.4%)	(1.2%)	(2.7%)	(1.9%)
executables w/	7	3	13	22	10	12	42	8
spy/mal. BEs	(0.5%)	(0.1%)	(0.3%)	(0.7%)	(0.3%)	(0.2%)	(0.4%)	(0.2%)
domains w/	5	3	10	16	7	8	15	7
spy/mal. BEs	(0.6%)	(0.2%)	(0.5%)	(1.0%)	(0.4%)	(0.2%)	(0.5%)	(0.3%)
BEs	17	13	201	208	32	79	232	172
non-benign	6	4	120	127	16	25	110	68
BEs	(35.3%)	(30.8%)	(59.7%)	(61.1%)	(50.0%)	(31.6%)	(47.4%)	(39.5%)
spy/malware	3	2	8	12	5	9	10	5
BEs	(17.6%)	(15.4%)	(4.0%)	(5.8%)	(15.6%)	(11.4%)	(4.3%)	(2.9%)

Identifier-based Detection. Table 7 contrasts our ground truth classification with the labeling according to CastleCops. Each table entry shows the number of unique BEs and, in parenthesis, the number of corresponding executables, based on their classification by CastleCops versus their true nature.

Table 7. Ground truth vs. CastleCops.

	-	**legitimate**	**debatable**	**ad-/spyware**
benign	62 (166)	186 (583)	57 (220)	2 (3)
adware	106 (278)	4 (10)	31 (1,641)	21 (56)
grayware	2 (2)	1 (3)	6 (52)	12 (225)
spyware	2 (4)	0 (0)	4 (15)	12 (72)
malware	0 (0)	0 (0)	0 (0)	4 (26)

When examining this table, the considerable number of debatable components reflects the general difficulty analysts face when they have to assign a certain category to a certain browser extension. Often, it is up to the user whether they consider the behavior of a component acceptable or not. Also, there are a quite large number of CLSIDs (106) used by adware BEs that we could not find in the online database. This is mainly due to *Softomate* components, discussed in the following paragraph.

In general, it can be seen that identifier-based identification works surprisingly well. Unfortunately, this kind of detection can be easily evaded, and certain spyware variants (e.g., *Win32.Stud.A*) already use randomly-generated CLSIDs.

Table 8. Ground truth vs. Ad-Aware.

	benign	**adware**	**data miner**	**malware**
benign	303 (963)	4 (9)	0 (0)	0 (0)
adware	15 (20)	14 (99)	130 (1,863)	3 (3)
grayware	8 (238)	3 (8)	10 (36)	0 (0)
spyware	4 (9)	2 (3)	7 (67)	5 (12)
malware	4 (26)	0 (0)	0 (0)	0 (0)

Scanner-based Detection. Table 8 shows our comparison with the reports provided by Ad-Aware. When we consider the similarity of our definition of *spyware* and Ad-Aware's description of a *data miner*, our results show a surprising mismatch in the number of detected samples. During our analysis, Ad-Aware (mis)labeled 130 unique adware components as data miner. All other techniques could not confirm these threats.

Closer examination of Ad-Aware's report showed that 92% of these mislabeled components are toolbars. To determine whether these components only track user data that is entered into the toolbar, we additionally performed manual testing. Some of these toolbars provide search results for paid advertisers, but only when we use the search function of the toolbar. Clearly, this is the expected behavior, and thus, should not be classified as data miner. We also contacted Lavasoft to resolve this issue. We were told that one possible cause for their classification might be the fact that the installation routine does not clearly state the purpose of an adware program, and thus, it is labeled as data miner. Additionally, they admit that some samples were misclassified.

One particular problem was caused by the *Softomate Toolbar*, which is a developer aid for creating customized Internet Explorer components. A few malicious samples are created using this tool. However, Ad-Aware tags *all* toolbars that are developed with the help of Softomate as data miner. This is unfortunate, because we observed that over 50% of all executables with browser extensions were using a component produced by Softomate. However, only a tiny fraction is recognized as malicious by all other detection techniques. Given the significant amount of adware BEs that were tagged as data miners by Ad-Aware, we recognize a significant bias that overstates the actual threat.

On the other hand, we also found four actual spyware threats not reported by Ad-Aware. Three of these threats were revealed by the behavior-based detection technique (as we show later below), and three could also be identified using Spybot. This demonstrates the limitations of signature-based detection and the possibility to underestimate the threat because of novel, malicious code instances. However, four cases are still a relatively small number. In two additional cases, a spyware threat was misclassified as adware.

Table 9 shows our comparison with Spybot. At first glance, it appears that Spybot misses a considerable amount of adware samples. On further examination, 93% of these BEs are Softomate Toolbars, a popular type of extension. As we discussed previously, we labeled these BEs as (mildly annoying) adware, but one could also argue that they are benign. Therefore, we consider this mismatch as negligible.

Table 9. Ground truth vs. Spybot.

	not detected	detected
benign	304 (965)	3 (7)
adware	131 (1,831)	31 (154)
grayware	8 (59)	13 (223)
spyware	3 (7)	15 (84)
malware	1 (2)	3 (24)

Table 10. Ground truth vs. behavior-based.

	not detected	detected
benign	300 (956)	7 (16)
adware	161 (1,984)	1 (1)
grayware	6 (8)	15 (274)
spyware	4 (13)	14 (78)
malware	4 (26)	0 (0)

Behavior-based Detection. Table 10 shows the performance of our taint analysis with respect to ground truth. As expected, those BEs leaking sensitive user information, such as URLs surfed by the user, could be found in the categories grayware and spyware. Since *benign* software and *adware* do not disclose private user information to a remote server, we cannot distinguish between these components.

A significant advantage of behavior-based, dynamic analysis is the fact that also novel threats can be identified. Thus, we would expect that the behavior-based approach can identify more spyware components than scanner-based techniques. Table 11 lists those BEs that were detected by the behavior-based analysis but missed by Ad-Aware. For seven unique extensions, we detected redirections for keywords entered directly into the browser's address bar. Two different, unique BEs leaked all surfed URLs to a remote third party.

Table 11. BEs detected by behavioral analysis but not Ad-Aware.

name	# variants	class
811_Toolbar	1	grayware
Camfrog Toolbar	1	grayware
CasinoDownloader	2	spyware
CyberDefender	1	grayware
NewDotNet	4	grayware
Offsurf Proxy	1	grayware
P2P Energy	1	grayware
Win32.Stud.A	1	spyware

Table 12. False positives raised by behavior-based detection.

name	# variants
ChildWebGuardian	3
GL-AD Popup Term.	1
PCTools Browser	1
SurfLogger	1
WhereWasI	1

The fact that Ad-Aware misses *NewDotNet* is problematic, as this component is the most popular grayware found by our crawler (accounting for 197 infected executables, as can be seen in Table 5). This introduces an imprecision into statistics that depend on Ad-Aware output. In addition to the seven grayware components, Ad-Aware also missed two spyware programs. Both programs transmit all the URLs that are surfed to a third party. More precisely, *Casino-Downloader* transmits all surfed URLs to `ad.outerinfoads.com` and various other affiliated severs. *Win32.Stud.A* is a BHO that is installed silently by a free picture viewer application. Interestingly, we observed that different CLSIDs are used every time the BHO is installed. This clearly indicates an attempt to evade identifier-based detection. This BHO records the URLs visited by the user and transmits them encrypted to `www.googlesyndikation.com`. When it detects certain keywords or URLs, it aggressively displays pop-up advertisements.

The behavioral analysis failed to recognize a few malicious components as spyware. One important reason was that several components attempted to connect to remote hosts that were no longer available. Thus, collected information could not be leaked. In other cases, the components were waiting for a particular trigger (a specific URL) that was not part of our set of visited URLs.

The behavior-based analysis considers a BE as spyware whenever it leaks tainted (sensitive) user information from the Internet Explorer process. However, there might be cases in which this operation is legitimate, giving raise to false positives. In the following, we discuss the samples that have been incorrectly labeled as spyware, although their behavior is (likely) legitimate. Table 12 provides an overview. For example, *ChildWebGuardian* tracks user surfing habits and is intended to give parental control over the sites visited by a child. Thus, it logs the list of URLs that a user visits to a local file, presenting it later to the parent for inspection.

It is interesting to note that all components that caused false positives write information (such as URLs) to the local file system only. Thus, the behavioral analysis could be modified so that a component is marked as spyware only when sensitive information is sent over the network (possibly via the file system or another process). For the analyzed components, this would *not* have caused additional false negatives.

Overall, the behavioral analysis captured the spyware threat most accurately. Together with Spybot, this technique correctly detected the largest fraction of malicious browser extensions. Moreover, it raised by far the smallest number of false positives (and this number could be further decreased, as discussed previously). Thus, when repeating our experiments without any manual analysis, the results of the behavioral technique can be used to classify unknown components. Adding tools such as Spybot can improve detection rates but also incorrectly inflates the number of spyware components due to false positives.

3.4 Comparison to Previous Work

When we compare our measurement results to findings in the previous study [2], we note certain similarities. For example, the previous study observed that be-

tween 5.5% and 13.4% of all crawled executables are spyware-infected. If we consider non-benign BEs of all categories, the fraction of infected executables we detect in our study is 6.6%. However, this number does not reflect the actual spyware threat present on the Internet. Rather than focusing on the real spyware-threat, it only provides a rough estimate of the number of programs that ship with possibly annoying, but nevertheless non-intrusive, advertising components. The reason is that only a small fraction of non-benign samples actually perform privacy-invasive operations (as shown in Table 4). A major reason for the different assessment of the threat level between our study and previous work is Ad-Aware. Ad-Aware was the only tool used in the previous study, and it mislabels a significant number of non-malicious adware programs as spyware (data miner). This leads to an overestimation of the actual number of executables that are infected with privacy-invasive components.

4 Related Work

As detailed in previous sections, our work was inspired by the measurement study presented in [2]. Similar to the methodology presented in that paper, we crawled the web for executables that were then automatically installed and analyzed. The major difference of our work is the way in which we perform our analysis. Instead of relying on a single tool, we use three different approaches to classify each executable. This allows us to derive a more precise assessment of the extent of the spyware threat on the Internet than was reported by the authors of the study in [2]. Moreover, we are able to identify the weaknesses of individual detection and analysis techniques. As a result, we can understand in which ways the results reported in the previous work might be biased.

Since malicious code is an important problem, a number of researchers have proposed techniques to analyze and detect malware. The details of the behavioral-based approach, which we used and extended in this paper to automatically identify spyware components, were previously presented in [4]. Other dynamic approaches [13, 14] to identify more general classes of malware based on their behavior often use taint propagation to detect suspicious information flows. Complementary to dynamic techniques, there are static analysis approaches [15] to identify malicious code patterns, and techniques [16] to extract network-based signatures that capture suspicious traffic flows.

5 Conclusion

In this paper, we present the results of a measurement study that attempts to quantify the extent of spyware-infected executables on the Internet. Inspired by previous work, we crawled the web for executables that were then installed and analyzed. In total, our experiment lasted around ten months. We crawled over 15 million URLs and downloaded more than 35 thousand executables. An important difference to previous work is the fact that we used three different analysis techniques. By combining the views from different vantage points, we

were able to identify the limitations of each individual technique. In particular, we found that Ad-Aware, the tool used for the previous study, significantly overestimates the severity of many samples. As a result, previous work might have overestimated the prevalence of privacy-invasive spyware. While we did find a non-negligible number of spyware-infested executables, the vast majority of non-benign browser extensions were not stealing private information but displaying annoying advertisements.

References

1. Good, N., Dhamija, R., Grossklags, J., Thaw, D., Aronowitz, S., Mulligan, D., Konstan, J.: Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware. In: Symposium On Usable Privacy and Security (SOUPS). (2005)
2. Moshchuk, A., Bragin, T., Gribble, S.D., Levy, H.M.: A Crawler-based Study of Spyware on the Web. In: Network and Distributed Systems Security Symposium (NDSS). (2006)
3. Egele, M., Kruegel, C., Kirda, E., Yin, H., Song, D.: Dynamic Spyware Analysis. In: Usenix Annual Technical Conference. (2007)
4. Kirda, E., Kruegel, C., Banks, G., Vigna, G., Kemmerer, R.: Behavior-based Spyware Detection. In: Usenix Security Symposium. (2006)
5. Wang, Y., Roussev, R., Verbowski, C., Johnson, A., Wu, M., Huang, Y., Kuo, S.: Gatekeeper: Monitoring Auto-Start Extensibility Points (ASEPs) for Spyware Management. In: Large Installation System Administration Conference. (2004)
6. Hackworth, A.: Spyware. US-CERT Publication (2005)
7. Lavasoft: Ad-Aware. `http://www.lavasoftusa.com/software/adaware`
8. Castlecops: The CLSID / BHO List / Toolbar Master List. `http://www.castlecops.com/CLSID.html`
9. Mohr, G., Stack, M., Rnitovic, I., Avery, D., Kimpton, M.: Introduction to Heritrix. In: 4th International Web Archiving Workshop. (2004)
10. Bellard, F.: QEMU, a Fast and Portable Dynamic Translator. In: Usenix Annual Technical Conference (Freenix Track). (2005)
11. Trendmicro: HijackThis. `http://www.trendsecure.com/portal/en-US/tools/security_tools/hijackthis`
12. Spybot: Spybot Search & Destroy. `http://www.safer-networking.org/`
13. Moser, A., Kruegel, C., Kirda, E.: Exploring Multiple Execution Paths for Malware Analysis. In: Symposium on Security and Privacy. (2007)
14. Yin, H., Song, D., Egele, M., Kruegel, C., Kirda, E.: Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis. In: ACM Conference on Computer and Communication Security (CCS). (2007)
15. Christodorescu, M., Jha, S., Seshia, S., Song, D., Bryant, R.: Semantics-Aware Malware Detection. In: Symposium on Security and Privacy. (2005)
16. Wang, H., Jha, S., Ganapathy, V.: NetSpy: Automatic Generation of Spyware Signatures for NIDS. In: Annual Computer Security Applications Conference (ACSAC). (2006)

Acknowledgments. This work has been supported by the Austrian Science Foundation (FWF) and by Secure Business Austria (SBA) under grants P-18764, P-18157, and P-18368, and by the European Commission through project FP7-ICT-216026-WOMBAT.