

Analysing Privacy-Invasive Software Using Computer Forensic Methods

Martin Boldt and Bengt Carlsson

School of Engineering, Blekinge Institute of Technology, SE-372 25 Ronneby, Sweden
{martin.boldt, bengt.carlsson}@bth.se

Abstract. User privacy is widely affected by the occurrence of privacy-invasive software (PIS) on the Internet. We present a computer forensic investigation method for detecting and analysing PIS. In an experiment we use this method to evaluate both the evolution of PIS and associated countermeasures, over a four year period. Background information on both PIS and countermeasure techniques are also presented, followed by discussions on legal disputes between developers of PIS and vendors of countermeasures.

1 Introduction

Technology has revolutionized the way we collect and process information. With the help of information technology it is possible to accumulate huge data quantities for later use. The fact that information (such as user interests) creates value to advertisers has given rise to a parasitic market, focusing on information theft [20]. Software vendors take advantage of these achievements based on questionable commercial incentives when creating and distributing questionable software. Throughout this paper we group such software together under the term *privacy-invasive software* (PIS). Our use of the concept of *privacy* lies within Warren and Brandies original definition, “the right to be let alone.” [41]. Since this paper target user privacy in the context of software programs, we focus on the following three parts:

- software that covertly sneaks into systems, or
- deceives users about their business, or
- exists without any control from users.

Users’ privacy are trespassed by PIS that covertly collect privacy-invasive information, present unsolicited contents, or secretly exchange requested contents with sponsored information. Such software covertly sneak into systems and hide deep inside the core, out of reach from user control. By also excluding normal program removal routines, usually provided by the operating system, such software assure future prosperity. Locating and removing PIS are therefore associated with great cost, which is further increased since widely deployed protection mechanisms, such as anti-virus tools, do not adequately address these threats [4]. Earlier work has analysed the behaviour and impact that PIS have on users’ computers, with regard to performance, privacy and security [4][6][24]. In this paper however, we investigate the use of *computer forensic* tools

and methods when locating PIS. In an experiment we evaluate the accuracy of a leading PIS protection tool by comparing it with the computer forensic method. This paper also touch upon the evolution of PIS and the legal tussles between developers of PIS and related countermeasures.

In the next section we present some background on privacy-invasive software followed by a review of available countermeasures. Section 4 introduces *computer forensics* into our setting and in the next section it is incorporated into a method to detect privacy-invasive software. We proceed in Section 6 by describing our experiment. In Section 7 the results from the experiment are presented followed by a discussion on our findings. We end this paper with some conclusions.

2 Background

Adware and spyware are the two most dominating types of PIS that are not adequately addressed by anti-virus programs [4]. Adware displays advertisements and commercial offers on users' systems while spyware covertly collect and then transmit privacy-invasive information to third parties [2]. Further definitions of various PIS types are discussed in [3][12][29]. The term spyware is often misleadingly used in a much broader sense that includes various other forms of PIS. However, throughout this paper we use the term spyware in its more narrow sense which is presented above.

Distribution of adware intensified as soon as marketers realized that Internet advertising could generate notable income from advertisers [12]. Online advertising is a \$6.9-billion-a-year market where adware is one of the fastest growing segments [27]. Adware vendors found that targeted Internet advertising performed significantly better than standard banner ads and that advertisers were prepared to increase their payment for this service [39]. This started the transformation from adware to a combination of adware and spyware [23][37]. Hereafter, other forms of PIS emerged such as hijackers that gain revenues by replacing Web content with commercial messages, e.g. sponsored links. To reach out to customers, PIS vendors began to distribute their software through Web sites and by bundling it together with otherwise free software, such as file-sharing tools. Bundling products with PIS allow software vendors to gain revenues from the PIS vendors (who get their income from advertisers) each time their product is installed by a user. Such revenues range from cents up to \$0.25, or more, per installation [12]. Some of these file-sharing tools have been downloaded well over 350 million times¹ which produce significant income for involved parties. These prerequisites has rendered in a steady increase over the last years in both the spread of PIS as well as in the sophistication that such programs use for accomplishing their goals [42].

Competition increases when more and more vendors of PIS try to benefit from this situation. To strengthen their positions they constantly need to identify new relative advantages over their competitors. In the search for such competitive advantages, PIS vendors turn to unethical or even illegal business strategies, e.g. stealing personal information for faster financial gains. The increase in PIS related problems has continued over

1. Kazaa has been downloaded more than 350 million times from one single source, Download.com [16].

the last years and various signs point at a steady future increase [17][33][36][42]. This increase also include the most malicious types of PIS. These programs now make use of computer virus self-preservation techniques, such as run-time encryption of the binary program to avoid detection. Additionally, these PIS also use virus-like strategies when attacking and gaining access to new systems [42].

Privacy-invasive software could integrate themselves into systems either by utilizing available software vulnerabilities, or by deceiving the user into installing them, i.e. to target and deceive users to install, what they think is a useful piece of software [21][32]. So, even in a context where software vulnerabilities are being exterminated and where accurate and sophisticated protection mechanisms exist, systems would still be susceptible to PIS. Techniques that allow for users to make informed decisions in advance on whether to install a certain software or not could mitigate this problem. One such approach, based on certification of “privacy friendly software”, has been developed by TRUSTe [40]. However, until such certifications are being commonly used we will have to adopt to the fact that only visiting the wrong site on the Internet could be equivalent with PIS infection [4][30]. Once a single PIS component has gained access to a system this piece of software could be used as a gateway for additional PIS to be installed [24].

However, during recent years users tend to be more aware of the threat posed by PIS [36]. This could be a result of both the widespread and the increase in media coverage of PIS. Even though user awareness grows does not mean they take appropriate actions to address the threat [46]. Statistics from sources such as Download.com (a leading software distribution site) strengthens this view. The top ten downloaded software from Download.com include three file-sharing tools (iMesh, LimeWire, Morpheus), known to be bundled with questionable software such as spyware, together with three spyware removal tools (Ad-Aware, Spyware doctor, Spybot). Users install file-sharing software, get problems with PIS and try to repair as much as possible of the damage caused by using spyware removal tools. Unfortunately the accuracy of these removal tools are far from exact which leaves the user with trespassed systems containing unsolicited harmful software, that result in a reduction of performance, stability, privacy and ultimately the security [6].

3 Countermeasures

In an attempt to stop, or at least mitigate, the PIS hazard a whole new group of software, called *anti-spyware*, or spyware removal tools, has emerged [22]. Companies producing such tools are constantly one step behind the authors of PIS. As anti-spyware tools evolve they create an arms-race between anti-spyware vendors and developers of PIS. A somewhat similar situation is the ever ongoing struggle between anti-virus companies and virus distributors, refined detection mechanisms have to fight more and more sophisticated viruses. This problem, addressed within a spyware domain is further intensified by the previously mentioned arms-race, caused by vendors of PIS endless search for competitive advantages over each other. As a result, anti-spyware vendors face three major problems to solve.

4.

1. The need to identify new and previously unknown types of PIS. This should be done in an environment of highly dynamic and evolving variety of PIS.
2. After successfully identifying a PIS component, any proper anti-spyware tool should remove the component and thereby bring the system closer to a previously uninfected state.
3. The anti-spyware tools' ability to safeguard user data and system components during the removal phase, i.e. to keep and protect legitimate files.

Of the three problems above, the first one is most important since both problem 2 and 3 are depending on it. The remaining part of this work will focus on the problem concerned with identifying PIS, and not on the removal aspects.

The first problem cannot fully be solved in an automated manner using only software countermeasures [11][13][14]. Since software cannot fully solve this problem alone, human interaction need to be involved to identify previously unknown forms of PIS. Anti-spyware tools set to inhibit this rapid advance of PIS use the four techniques listed below to fulfil this task. All four techniques could be implemented at various locations of an infrastructure, e.g. on network routers, servers, or local workstations [5][25][43].

1. Manual identification
2. Signature based identification
3. Heuristic identification
4. Automated Internet investigation

In the *manual method*, system changes are manually looked upon and analysed by an investigator. By tracking system changes it is possible to detect both previously known and unknown PIS, since they always leave traces on infected systems, e.g. in the form of executable instructions. However, this complete identification scenario is associated with a high cost, as it is considerably time-demanding since it relies on techniques such as static analysis and reverse engineering of the binary program [18][32].

Signature based identification relies on a database holding signatures of known PIS. A signature captures unique properties of PIS, and could be thought of as a fingerprint. By comparing items in a system with the signatures in the database it is possible to identify already known PIS. However, as soon as a new PIS emerge, anti-spyware vendors need to find it, produce a signature associated with the new threat, and finally distribute the new signature to the users. This method is widely used, despite the delay in protection; since it is possible to create software that automate the detection process.

Heuristic identification could be fully automated. It relies on lists of prohibited software behaviours. The method emulates the execution of a binary program and compare each of its instructions with the list of prohibited behaviour before the program is really executed. For each match against the list of prohibited behaviour a score is added to the program. When all instructions in the program has been evaluated the total scores are summarized. If the total amount exceeds a certain threshold the program is considered to be a PIS. This method is capable of detecting previously unknown PIS as long as the list of prohibited behaviours match the behaviours of the new threat, i.e. the list is up-to-date. However, this method presents the user with various degrees of false-positives

(false alarms) depending on the threshold value. Increasing the threshold value to cope with high false-positive levels will result in increased false-negatives (undetected PIS) levels. Getting the right balance between false-positives and false-negatives is hard and is a significant drawback for this method. These drawbacks result in that users need to adjust both the threshold level and what software behaviours that should be prohibited to suit their computing environment.

The *automated Internet investigation* method is only used by anti-spyware vendors and not by the users directly. Once an anti-spyware vendor identifies a previously unknown PIS with this model a corresponding signature is created and distributed to the user's signature database. By using this method anti-spyware vendors decrease the delay in time associated with signature creation and distribution. Since this method is created by commercial interests most information about its inner workings are protected as part of the IT-business strategy. However, from the sparse information available on the Web we can conclude that the method relies on vast numbers of test computers, holding virtual systems, which automatically surf the Web in search for sites containing suspicious programs [43]. By using pattern recognition on the data received from each site visited, both known and previously unknown PIS can be identified.

An emerging trend is that PIS developers sue anti-spyware vendors for defamation and ruined business strategy, by classifying and treating their product as PIS. Some of these cases have escaped captivity to public attention and several ended up in court [38]. The most recent case involves the online marketing company "180Solutions" that sued firewall company "Zone Labs" for classifying their advertising client as spyware [47]. We believe that vendors of countermeasure tools need to be more accurate in their classification of PIS in the future, and that their decisions need to be based on solid evidence that hold for use in court. We also believe that those anti-virus vendors not addressing PIS are at less risk, since developers of malicious software, such as viruses or worms, will not sue the company because their actions are without a doubt illicit.

To separate vendors of legal marketing tools from developers of PIS a general agreement on what should be considered to be fair business practices, need to be established between developers of PIS and countermeasures [7][31]. At least until such an agreement is reached, any cautious anti-spyware vendor should keep trustworthy evidence to back up their PIS classification decisions. Using a method designed to deliver such solid evidence will be of paramount importance for every company that classifies and treats software as privacy-invasive.

4 Computer Forensic Methods

Individuals and companies rely on computers in their daily work and for doing personal duties such as online banking errands. Criminals take advantage of this fact by using computers when committing crimes. To investigate such crimes, law enforcement agencies rely on *computer forensics* [10]. It is the process used when investigating crimes involving evidence in various digital forms which is destined for use in a court [8][9]. A central part of computer forensics is that every event in the life cycle of evidence may never alter the evidence itself. Additionally, throughout the evidence life cycle every action must be well documented so that the court in the end can estimate the

amount of reliability to put in it. Main steps in computer forensic investigations involve, identification and collection of evidence, data harvesting, data reduction, reorganizing and search of data, analysis of data and finally reporting. These steps constitute a formalized process that help investigators reach conclusions that are repeatable, based on evidence, and as free as possible from errors. Anti-spyware vendors could benefit from this if PIS developers sue the vendor for ruining their business strategy when removing their tool [38]. If clear and stringent evidence together with proper handling of the evidence, could be presented to a court, it would assist the anti-spyware vendor in reaching a favourable outcome in the case.

One important principle in forensic science is *Locard's exchange principle* [26] which determines that anyone or anything, entering a crime scene takes something of the scene with them, and leaves something of them behind as they leave. This principle could also be applied in most computer settings, involving for instance PIS infections since these types of software leave tracks in both file-system and network communication. In Section 5 we discuss this in more detail.

To aid computer forensic investigators in the investigation process there exist both public domain and commercial tools. These tools allow investigators to analyse copies of whole systems, i.e. the investigator can see everything stored in a file-system. In our investigation we used a commercial tool called *Forensic Tool Kit* (FTK) which is developed by AccessData [1]. FTK has been thoroughly tested not to alter the evidence that is being investigated.

5 Investigation

In previous investigations of PIS we used a manual investigation method that is based on system state preservation [6][24][45]. By preserving the state of a system, together with complementing information (such as network traffic), it is later possible to retrieve a specific system state for analysis. During both the planning and execution of our experiment we had two main goals concerning the laboratory environment:

1. Preserve identical hardware and software configurations during all investigation steps.
2. Use default software configurations and all available security updates.

To preserve bit-wise identical system states we rely on standard BSD Unix components. This allow us to serialize a whole system into a bit-wise identical *clone file*. Such a clone file is a snapshot of a system at a specific time. From such a clone file it is later possible to restore a system and its state for analysis. Initially a snapshot of a "clean" system is created, this is regarded as the *baseline*. Such a baseline only includes the operating system and the tools used for experiment measurements. Next, an action of some kind is executed which result in infection of PIS. Such actions could be for instance, surfing to certain Web sites or installing a program bundled with PIS. Immediately after this action is performed another snapshot is taken. Depending on the experiment, additional snapshots could be created at certain intervals. Using these snapshots allows investigators to track system-changes between the points in time when the snapshots were taken. For instance, to identify any system-changes that were introduced during the installation of software A, we need to conceal all system parts in the post-installation

snapshot that are identical with the baseline. In some sense we remove the baseline from the post-install snapshot. Now, only system changes that occurred during installation of software A remains.

Our method detects any system deviation that has occurred between two points in time. Simultaneous data collection and analysis is avoided since the method has a clear separation between collection and analysis of data. The method force investigators to collect data only once, and later take the time needed to analyse this data. The level of detail in the data captured is very high which results in extensive data quantities that need to be handled. We address this problem by automating much of the structuring and refinement steps through custom-made software. However, this method cannot be fully automated since steps involving for instance data recognition and reduction rely on the skills of the investigator. Since the method cannot be fully automated it is considerably more resource demanding than automated signature based anti-spyware tools. But we believe that computer forensic tools could reduce this problem to an acceptable level.

To evaluate our investigation method we conducted an experiment set to analyse the accuracy of an anti-spyware tool in identifying PIS, bundled with three *peer-to-peer* (P2P) file-sharing tools over a four year period [19][28]. We choose to investigate an anti-spyware tool called Ad-Aware since it was the most downloaded anti-spyware tool from Download.com in October 2005, with more than 175 million downloads. The experiment used 13 identical computers holding four versions of the three most downloaded P2P file-sharing tools according to Download.com, i.e iMesh, LimeWire, and Kazaa, together with one reference machine without any file-sharing tool installed. All three file-sharing tools are widely deployed, each with between 68 and 390 million downloads [16]. The versions of the three file-sharing tools were all from 2002 until 2005, and claimed to be free from any forms of spyware. Since all of the investigated file sharing tools were developed for the Windows platform our experiment were executed in a Windows 2000 environment. Windows XP could not be used since it was incompatible with earlier versions of LimeWire. Even though file-sharing is not restricted to the Microsoft Windows platform most problems concerning PIS are [34].

In the beginning of the experiment each of the 13 computers were identical and the system state was stored with a baseline snapshot. However, system deviations began as soon the various file-sharing tools were installed. Directly after the installation process was completed a new system snapshot was created for each system. After this the systems were left to execute continuously for 72 hours. During this time all computers were left uninterrupted, except for an automated Web surfing program that was set to simulate a user visiting a number of company Web sites, such as Amazon and Apple. This was done in an attempt to trigger any dormant PIS lurking in the system. In the end of the 72 hour execution new snapshots were taken for each system. As a final step we installed and executed six versions (from 2000 until 2005) of an anti-spyware tool called Ad-Aware on each of the 13 computers. The result of these Ad-Aware executions was stored for later analysis.

To analyse the data gathered from the experiment we mainly used FTK, which offers several techniques useful for an investigator, such as pre-indexation of data, and a known file filter. Pre-indexation is a technique that indexes all data once, when the evidence is loaded. Later, during data harvesting, this result in instant search results from

all data in the investigated system. The known file filter is a technique based on cryptographic hash values that allows FTK to recognize and label files as, e.g. non tampered system files which could be concealed to the investigator. FTK also includes ways to inspect and label files based on various properties, e.g. encryption, text, binary, or image. This allows for an investigator to highlight all encrypted files through one button.

To automate tasks when sorting and filtering data we also used custom-made programs written in C and as shell scripts. PIS components identified by Ad-Aware were checked against the actual system which allowed us to identify numerous false-positives, reported by Ad-Aware. On some occasions different versions of Ad-Aware reported a single PIS by several names. We choose to report all such PIS with the latest used name. Further, we investigated all added or modified programs and components, except for the file-sharing executables. To identify if any of the files missed by Ad-Aware should be considered PIS we used static analysis based on file properties such as filename, hash value, identification tags, and strings located inside the binary program [18]. This information was then checked against two resources for classification [15][35].

6 Results

In Table 1 the total number of PIS, cookies, register keys and other components is measured as the difference between the clean system and a system “infected” by different versions of Kazaa, iMesh and LimeWire. Different versions including their release date of Ad-Aware, an anti-spyware programme, are used for the examination. The shadowed part shows the added components found by a present or future version of Ad-Aware, i.e. the actual protection against a certain version of the P2P programs.

Table 1: Total number of added components for three P2P-programs (iMesh, LimeWire and KaZaa) measured by six different versions (3.5 to SE1.06) of Ad-Aware between 2002 and 2005.

Ad-Aware	3.5 aug-00	5.5 jun-01	5.7 mar-02	6.0 mar-03	1.05 sep-04	1.06 nov-05
2002	8	59	183	278	912	638
2003	6	24	15	18	222	232
2004	11	34	38	34	218	221
2005	0	2	5	4	142	128

In general, present versions of Ad-Aware finds more components than older. 2002, 2003 and 2004 versions of the P2P-programs show a 3, 12 and 6 times increase respectively of added components. For 2002 there is a second 2-3 times increase when future versions of Ad-Aware measures the added components. Ad-Aware 2005 reported fewer components on average than the 2004 version of the program.

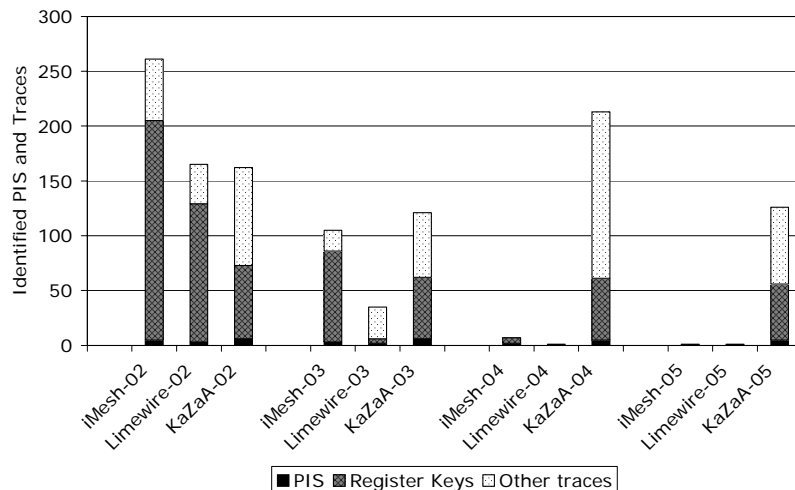


Figure 1. Number of bundled PIS programs, registry keys, and suspicious files/folders for iMesh, LimeWire and Kazaa reported by Ad-Aware over a four year period.

Figure 1 presents the amount of PIS programs, registry keys, and other traces that are being injected into a system when a P2P-program is installed. Register keys are not complete programs but are used by PIS during execution. The most dominating group of traces consist of register keys followed by not specified traces, e.g. suspicious files and folders. The actual number of executable PIS is greatly inferior compared to other traces. Kazaa 2002-2005 shows a large number of added components each year. iMesh shows a peak 2002 with progressive decreasing the years before and after, whereas LimeWire had very few added components outside the years 2002 and 2003.

Table 2: Number of PIS in three P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions (3.5 to SE1.06) of Ad-Aware and our manual forensic method (FTK). Numbers in brackets indicate traces of PIS that misleadingly was reported by Ad-Aware as fully functioning PIS.

Ad-Aware	3.5 aug-00	5.5 jun-01	5.7 mar-02	6.0 mar-03	1.05 sep-04	1.06 nov-05	FTK AdAw	FTK New
2002	1	3	3 (2)	8 (1)	8 (2)	7 (2)	11	4
2003	2	3	2 (1)	2 (2)	5 (3)	5 (3)	7	3
2004	2	3	2 (1)	2 (1)	4 (1)	4 (1)	5	3
2005	0	0	(1)	(1)	2 (1)	3 (1)	3	3

In Table 2 all exclusive PIS programs found in Kazaa, iMesh and LimeWire are counted for different versions of Ad-Aware. Ad-Aware misleadingly reported some

traces such as registry keys as fully functioning PIS. These false positives are presented as the numbers inside brackets in Table 2. The second column from the right presents the number of PIS found by either the manual forensic method or at least one version of Ad-Aware. PIS components detected by the manual method but missed by Ad-Aware are presented in the last column.

Most PIS programs were found in the 2002 version of the P2P programs with a total of 15 different programs. 11 of these programs were reported by Ad-Aware, but different versions reported a variable number. Ad-Aware prior to 2002 reported less PIS and later versions reported more, however not all of them. Besides not reporting all PIS, Ad-Aware contrarily also reported, in all three different, PIS which instead were only traces thereof, and therefore wrongly classified as functioning PIS. Our manual method found four additional PIS never reported by any version of Ad-Aware.

For the forthcoming years a similar interpretation of Table 2 shows that the number of PIS declines, especially for iMesh and LimeWire, but the number of unreported PIS programs are still about the same as for 2002.

Table 3: Total number of undiscovered PIS programs in three P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions (3.5 to SE1.06) of Ad-Aware.

Ad-Aware	3.5 aug-00	5.5 jun-01	5.7 mar-02	6.0 mar-03	1.05 sep-04	1.06 nov-05
2002	14	12	11	7	4	4
2003	8	7	7	7	3	3
2004	6	5	5	5	3	3
2005	6	6	6	4	3	3

In Table 3 the earlier results of PIS found by Ad-Aware and the manual forensic method are presented as the failure numbers of Ad-Aware. This is the best possible result using all known versions of Ad-Aware, some PIS may in later versions be reclassified as harmless files. More recent versions of Ad-Aware (grey shadowed in Table 3) found a larger number of PIS than older versions. Sometimes, as for the P2P-tools from 2002, a delay exists in finding new PIS, i.e. later versions of Ad-Aware reported more PIS programs and traces. This delay lasted for the forthcoming two years.

Table 4 shows the 25 different PIS present in Kazaa, LimeWire and iMesh. In all 19 behaved as adware, 14 as spyware, 13 as hijackers that alter Web content, and two were able to independently download new programs.

Ad-Aware was able to find 18 out of 25 programs, or about 70% covering of PIS, but did not exclusively detect a certain PIS behaviour. Approximately 80% of all adware, 70% of all hijackers, 60% of all spyware, and 50% of the downloaders were detected by Ad-Aware.

Table 4: Classification (adware, spyware, hijacker or downloader) of found PIS programs. In the host column K refer to Kazaa, L to LimeWire and I to iMesh. An X in the Ad-Aware column indicates that at least one of the investigated Ad-Aware versions found the PIS program.

Name	Host	Adware	Spyware	Hijack	Download	Ad-Aware
AltnetBDE	K	X	X			X
BestOffers	K	X	X			X
BonziBuddy	L	X	X	X		X
Bullguard	K	X				X
Claria	I,K	X	X	X		X
CommonName	I	X		X		X
Cydoor	I,K,L	X		X		X
DownloadWare	K				X	X
eZula	I,L	X		X		X
FavoriteMan	I		X		X	
HotBar	K	X	X	X		
Instafinder	K			X		X
MarketScore	I		X			
MediaLoads	K	X	X	X		
MyWay Speedbar	I,K			X		
Need2Find	K	X	X	X		
NewDotNet	I,K	X	X	X		X
Nodopops	K	X	X			
PerfectNav	K	X		X		X
PromulGate	K	X				X
RX Toolbar	K	X	X			X
ShopAtHome	I		X			X
Stop-Sign AV	I	X				X
TopMoxie	L	X		X		X
WhenU	I,K	X	X			X

7 Discussion

Forensic tools imply more manual work sorting out wrongly classified executable files than traditional signature based tools, but guarantee complete PIS detection. A forensic analysis method effectively excludes the false positives found in this investigation. By using a forensic tool it was possible to find all undetected or unreported program files otherwise missed, i.e. all installed executable files.

Unlike viruses, PIS programs exist in a grey area between being legal (business facilitators) and being illegal, i.e. behave and/or being regarded as malicious software. Normally, a virus is rapidly identified, does not cause any classification problem, and once included in the anti-virus database it remains there. Ad-Aware, the investigated anti-spyware tool, was unsuccessful with respect to all three anti-virus qualities above, i.e. PIS was slowly identified, caused classification problems, and was sometimes excluded.

The first quality, speed of identification, compromises PIS that is not reported by Ad-Aware for a certain version of the file sharing program. This could be due to that some PIS is not yet classified as PIS, i.e. they are detected but is not included into the signature database, or that PIS successfully conceal themselves from anti-spyware tools. As was shown in Table 3 the failure numbers of Ad-Aware decreased over the time showing a gradual incorporation of new PIS into its database. It took one to two years for Ad-Aware to incorporate missing PIS in the database and there were still undetected programs. Compared to anti-virus programs this is too long time and with a remaining unacceptable failure number.

The second quality, classification consistency, suffer from the presence of false negatives and positives. Reclassification, unreported and undetected files may all be false negatives, i.e. PIS found during the forensic analysis but not reported or ignored by the anti-spyware tool. Ad-Aware found about 70% of all PIS and did not show any trend to exclusively favouring the detection of a certain behaviour. Also, a lack of a deepened context analysis may influence the amount of false positives, i.e. warnings, generated by the anti-spyware tool that do not pose any risk at all. Ad-Aware did not distinguish between traces of PIS and executable programs.

The third quality, stability, was violated because executable program files, formerly by Ad-Aware classified as PIS, was later excluded. Three such programs, behaving as adware, spyware or hijackers were found. There were no obvious reason for reclassifying these programs because of more harmless actions. Instead there are different business considerations for anti-spyware tools compared to anti-virus tools, such as legal aspects of excluding third-part material.

In all, the 2005 version of Ad-Aware found 15 PIS out of 25 for the 2002-2005 versions of the three P2P tools. Also, later versions of P2P tools contained fewer PIS than older versions. So, the decrease in the number of PIS is probably not the result of more efficient countermeasures, but refined business strategies. Either a company tries to exclude its marketing program from the anti-spyware databases or choose another kind of marketing. Both strategies are found in the 2005 versions where iMesh and LimeWire excluded all PIS and Kazaa contained a bigger rate of undetected files.

We believe the failure from anti-spyware tools to deal with the three qualities above rely on both obsolete identification techniques, but also on the lack of a general agreement on what should be considered as privacy-invasive behaviour of software. Without such an agreement it is a more arbitrary task to distinguish privacy-invasive software from legitimate software than separating malicious software, such as virus and worms, from legitimate software. Another drawback for anti-spyware vendors is that the tools they classify as PIS are developed by companies that are ready to take legal actions if

needed. This is a scenario most anti-virus vendors do not have to worry about when classifying and treating for instance a worm as malicious software. Both vendors of anti-spyware tools and marketing companies need to commonly establish where to draw the border between PIS and legitimate business facilitators. If such an agreement could be reached, both legitimate marketing companies and vendors of anti-spyware tools will benefit. Legitimate marketing vendors no longer need to be affected by decreased revenues since their advertising clients were wrongly classified as PIS, and anti-spyware companies face a lower risk of being sued by indignant marketing vendors. Additionally, every deceitful software developer creating PIS would be treated, rightfully, by the anti-spyware vendors.

If a general separation between privacy-invasive and legitimate software could be established it would be possible to certify software as “privacy friendly”. Complementing such a certification with a short description on e.g. software behaviour, transmitted data, and removal routines it would be possible for users to make informed decisions on whether or not to install a certain software. Such a service would provide users with an important tool that allow them to increase the amount of control they have over their systems and their digital privacy, on both home computers and mobile devices.

To deal with the inaccurate results from anti-spyware tools we believe a combined approach using both a signature database and system snapshot technique would be fruitful. Such a tool could automatically identify every newly added or modified executable program on the system, and send an alert to the user.

8 Conclusions

In an ongoing arms race between PIS and anti-spyware vendors identifying, removing and keeping/protecting legitimate files are major problems to solve. Four techniques for identifying PIS are discussed; manual, signature based, heuristic and automated Internet identification. The identification task is further complicated by the necessity to consider legal aspects which is a major distinction between anti-spyware and anti-virus tools.

The article compared different versions of three file-sharing programs by means of included PIS. The effectiveness of comparable versions of an anti-spyware program was correlated against a manual method using a forensic tool comparing a “clean” system with the system infected by added components from the file sharing tools.

The investigated anti-spyware program failed to report all PIS programs, marked earlier discovered PIS as ordinary programs, or wrongly classified traces of PIS as functioning PIS. There was also a palpably reduction of PIS programs included in later versions of two out of three file sharing programs. The manual forensic method managed to find all added executable files and to sort out traces of PIS.

Unlike viruses, PIS programs exist in a grey area between being business facilitators and being regarded as malicious software. Compared to the more established anti-virus approach, the investigated anti-spyware tool suffered from three quality attributes; rapid identification, classification consistency and conformity violations. We believe the failure from anti-spyware tools to deal with the three qualities above rely on both

obsolete identification techniques, but also on the lack of a general agreement on what should be considered as privacy-invasive behaviour of software.

An up-to-date version of Ad-Aware finds more components than an older, because of an upgraded database. Unlike a virus database, PIS may be reclassified or incorporated years later, i.e. it is much easier for a PIS to escape detection. In the future a combined anti-spyware tool, targeting already known PIS with its signature database, and a forensic tool, finding all added or modified executable files, may bring users closer to an acceptable digital environment free from systematic privacy invasions. To reach such a goal it is also of most importance to develop routines that allow users to make informed decisions prior to the software installation process, on whether to install a certain software or not.

9 References

- [1] AccessData Corporation, <http://www.accessdata.com>, 2006-01-03.
- [2] W. Ames, "Understanding Spyware: Risk and Response", in *IEEE Computer Society - IT Professional*, Vol. 6, Issue 5, 2004.
- [3] Anti-Spyware Coalitions, <http://www.antispywarecoalition.org>, 2006-01-03.
- [4] K. P. Arnett and M. B. Schmidt, "Busting the Ghost in the Machine", in *Communications of the ACM*, Vol. 48, Issue 8, 2005.
- [5] Blue Coat Systems - Spyware Interceptor, <http://www.bluecoat.com/products/interceptor/>, 2006-01-03.
- [6] M. Boldt, B. Carlsson, and A. Jacobsson, "Exploring Spyware Effects", in *Proceedings of the Eighth Nordic Workshop on Secure IT Systems*, Helsinki Finland, 2004.
- [7] J. Bruce, "Defining Rules for Acceptable Adware", in *the Fifteenth Virus Bulletin International Conference (VB2005)*, Dublin Ireland, 2005.
- [8] B. Carrier, "File System Forensic Analysis", Addison-Wesley Professional, Upper Saddle River, NJ, 2005.
- [9] H. Carvey, "Windows Forensics and Incident Recovery", Addison-Wesley, Upper Saddle River, NJ, 2005.
- [10] E. Casey, "Digital Evidence and Computer Crime: Forensic Science and the Internet", Academic Press, London UK, 2004.
- [11] D. M. Chess and S. R. White, "An Undetectable Computer Virus", in *Virus Bulletin Conference*, Orlando FL, 2000.
- [12] E. Chien, "Techniques of Adware and Spyware", in *Fifteenth Virus Bulletin International Conference (VB2005)*, Ireland, 2005.
- [13] F. Cohen, "Computational Aspects of Computer Viruses", in *Computers & Security*, Vol. 8, Issue 4, San-Fransisco CA, 1989.
- [14] F. Cohen, "Computer Viruses - Theory and Experiments", in *IFIP-Sec 84*, Toronto Canada, 1984.
- [15] Computer Associates Spyware Information Center, <http://www3.ca.com/securityadvisor/pest/>, 2006-01-03.
- [16] Download.com, <http://www.download.com>, 2006-01-03.

- [17] “Emerging Internet Threats Survey 2003”, commissioned by Websense International Ltd., February, 2003. http://netpartners.com/company/news/research/Emerging_Threats_2003_EMEA.pdf, 2006-01-03.
- [18] D. Farmer, and W. Venema, “*Forensic Discovery*”, Addison-Wesley, Upper Saddle River NJ, 2004.
- [19] A. Froemmel, “Dangers And Containment Of P2P Utilities On A Corporate Network”, SANS Reading Room, SANS Institute, 2003. http://www.giac.org/certified_professionals/practicals/gsec/2828.php, 2006-01-03.
- [20] S. Görling, “An Introduction to the Parasite Economy”, in EICAR 2004, Luxemburg, 2004.
- [21] G. Hoglund, and G. McGraw, “*Exploiting Software - How To Break Code*”, Addison-Wesley, Boston MA, 2004.
- [22] L. Hunter, “*Stopping Spyware*”, Addison-Wesley, Upper Saddle River, NJ, 2005.
- [23] A. Jacobsson, “Exploring Privacy Risks in Information Networks”, in *Blekinge Institute of Technology Licentiate Thesis Series No. 2004:11*, Sweden, 2004.
- [24] A. Jacobsson, M. Boldt, and B. Carlsson, “Privacy-Invasive Software in File-Sharing Tools”, in *Proceedings of the 18th IFIP World Computer Congress*, Toulouse France, 2004.
- [25] Lavasoft, <http://www.lavasoft.com>, 2006-01-03.
- [26] E. Locard, “*L'enquête criminelle et les méthodes scientifiques*”, Flammarion, Paris France, 1920.
- [27] R. Martin, “Spy vs. spy”, in *Fortune Small Business*, Vol. 14, No. 4, 2004.
- [28] A. Oram, “*Peer-To-Peer: Harnessing the Benefits of a Disruptive Technology*”, United States of America: O'Reilly & Associates Inc., 2001.
- [29] S. Sariou, S.D. Gribble, and H.M. Levy, “Measurement and Analysis of Spyware in a University Environment”, in *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco CA, 2004.
- [30] S. Shukla and F. Fui-Hoon Nah, “Web Browsing and Spyware Intrusion”, in *Communications of the ACM*, Vol. 48, Issue 8, 2005.
- [31] J. Sipior, B.T. Ward, and G.R. Roselli, “A United States Perspective on the Ethical and Legal Issues of Spyware”, in *The Seventh International Conference on Electronic Commerce (ICEC2005)*, Xian China, 2005.
- [32] E. Skoudis, “*Malware - Fighting Malicious Code*”, Prentice Hall PTR, Upper Saddle River NJ, 2004.
- [33] Spyaudit, commissioned by Earthlink Inc., <http://www.earthlink.net/spyaudit/press/>, 2006-01-03.
- [34] Spyware is Windows-only, <http://www.securityfocus.com/news/9696/>, 2006-01-03.
- [35] SpywareGuide.com, <http://www.spywareguide.com>, 2006-01-03.
- [36] Stay Safe Online, “*AOL/NCSA Online Safety Study - December 2005*”, http://www.stay-safeonline.org/pdf/safety_study_2005.pdf, 2006-01-03.
- [37] J. Sterne and A. Priore, “*E-Mail Marketing - Using E-Mail to Reach Your Target Audience and Build Customer Relationships*”, John Wiley & Sons Inc., New York NY, 2000.
- [38] Threats Against Spyware Detectors, Removers, and Critics, <http://www.benedelman.org/spyware/threats/>, 2006-01-03.

- [39] K. Townsend, "Spyware, Adware, and Peer-to-Peer Networks: The Hidden Threat to Corporate Security" (technical white paper), PestPatrol, 2003., <http://www.moorecomputing.net/SpywareAdwareP2P.pdf>, 2006-01-03.
- [40] TRUSTe - Make Privacy Your Choice, <http://www.truste.com>, 2006-01-03.
- [41] S.D. Warren and L.D. Brandeis, "*The Right to Privacy*", in *Harvard Law Review*, Vol. 4, Issue 5, 1890.
- [42] Webroot Software, "*State of Spyware - Q3 2005*", <http://www.webroot.com/resources/>, 2006-01-03.
- [43] Webroot Software - Phileas, <http://www.webroot.com/resources/phileas/>, 2006-01-03.
- [44] A. Westin, "*Privacy and Freedom*", Atheneum, New York NY, 1968.
- [45] J. Wieslander, M. Boldt, and B. Carlsson, "Investigating Spyware on the Internet", in *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, Gjøvik Norway, 2003.
- [46] X. Zhang, "What Do Consumers Really Know About Spyware?", in *Communications of the ACM*, Vol. 48, Issue 8, 2005.
- [47] Zone Labs Sued Over Spyware Classification, <http://www.securityfocus.com/brief/68>, 2006-01-03.