Blekinge Institute of Technology
Licentiate Series No. 2007:01
ISBN 978-91-7295-100-6
ISSN 1650-2140

2007-01-02

Privacy-Invasive Software

Exploring Effects and Countermeasures

Martin Boldt

Department of Systems and Software Engineering
School of Engineering
Blekinge Institute of Technology
Sweden

Blekinge Institute of Technology
Licentiate Series No. 2007:01

ISBN 978-91-7295-100-6
ISSN 1650-2140

Printed in Sweden
Kaserntryckeriet AB, Karlskrona 2007

To Lena

This thesis is submitted to the Faculty of Technology at Blekinge Institute of Technology, in partial fulfillment of the requirements for the degree of Licentiate of Technology in Computer Science.

Contact Information

Martin Boldt
Department of Systems and Software Engineering
School of Engineering
Blekinge Institute of Technology
PO Box 520
SE-372 25 Ronneby
SWEDEN

E-mail: martin.boldt@bth.se
Web: http://www.bth.se/tek/aps/mbo.nsf

Abstract

As computers are increasingly more integrated into our daily lives, we need aiding mechanisms for separating legitimate software from their unwanted counterparts. We use the term *Privacy-Invasive Software* (PIS) to refer to such illegitimate software, sometimes loosely labelled as *spyware*. In this thesis, we include an introduction to PIS, and how it differs from both legitimate and traditionally malicious software. We also present empirical measurements indicating the effects that PIS have on infected computers and networks. An important contribution of this work is a classification of PIS in which we target both the level of user consent, as well as the degree of user consequences associated with PIS. These consequences, affecting both users and their computers, form a global problem that deteriorates a vast number of users' computer experiences today. As a way to hinder, or at least mitigate, this development we argue for more *user-oriented* countermeasures that focus on informing users about the behaviour and consequences associated with using a particular software. In addition to current reactive countermeasures, we also need preventive tools dealing with the threat of PIS *before* it enters users' computers.

Collaborative reputation systems present an interesting way forward towards such preventive and user-oriented countermeasures against PIS. Moving the software reputations from old channels (such as computer magazines or friends' recommendations) into an instantly fast reputation system would be beneficial for the users when distinguishing unwanted software from legitimate. It is important that such a reputation system is designed to address antagonistic intentions from both individual users and groups thereof, so that users could depend on the reputations. This would allow users to reach more informed decisions by taking the reported consequences into account when deciding whether they want a specific software to enter their computer or not.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor and collaborator, *Dr. Bengt Carlsson*, for both his creative support and guidance throughout this work and for always finding the time. I would also like to thank my examiner *Professor Paul Davidsson*, for the work he has put down in helping me form this thesis.

Colleagues at Blekinge Institute of Technology also deserve thanks. Not at least the members of the DISL research group for valuable discussions and paper reviews. In particular, I want to thank my friend and colleague *Andreas Jacobsson*, for his great humour, valuable feedback, and for many interesting discussions.

I also want to thank *Per Jönsson* for giving me access to his FrameMaker template and for helping me with various problems concerning FrameMaker, *Mikael Svahnberg* and *Patrik Berander* for valuable advice in writing a licentiate thesis. Further more, I want to thank *Tobias Larsson* and *Niklas Lindén* for their great work in implementing some of the ideas in this thesis into a proof-of-concept reputation system for software[1]. I also wish to thank all my old friends, whom I unfortunately see too little of these days. You're all great!

I am forever grateful to my parents *Ingegärd* and *Jerker* for their endless and unconditional support and love, and for forming such a wonderful family. Special thanks also to my brother *Christian* and my sister *Elisabeth* for many great memories, and for many still to come.

Most importantly, I want to thank my beloved *Lena* for keeping up with me during this work, including the sometimes odd working hours. Last (and I guess also least) I would like to thank our Bichon Frisé named *Tova*, for forcing me out on walks around the neighbourhood. During these nightly wanderings I often find the time to contemplate on various things, sometimes even about spyware and potential countermeasures.

1. More information is available at: http://www.softwareputation.com

Preface

Throughout this thesis I use "we" to clarify that several people in addition to the authors have made contributions to this work. All papers have been scrutinized by both colleagues and members of our research group, and they have also been peer-reviewed at the corresponding conferences. This thesis is based on the following four publications.

A. Jacobsson, M. Boldt and B. Carlsson, "Privacy-Invasive Software in File-Sharing Tools", in the *proceedings of the 18th IFIP World Computer Congress (WCC2004)*, 2004, Toulouse France.

M. Boldt, A. Jacobsson, and B. Carlsson, "Exploring Spyware Effects", in the *proceedings of the 9th Nordic Workshop on Secure IT Systems (NordSec04)*, Helsinki Finland, 2004.

M. Boldt and B. Carlsson, "Analysing Countermeasures Against Privacy-Invasive Software", in the *proceedings of the IEEE International Conference on Software Engineering Advances (ICSEA'06)*, Papeete French Polynesia 2006.

M. Boldt and B. Carlsson, "Privacy-Invasive Software and Preventive Mechanisms", in the *proceedings of the IEEE International Conference on Systems and Network Communications (ICSNC'06)*, Papeete French Polynesia, 2006.

The following papers are not included in this thesis:

T. Larsson, N. Lindén, M. Boldt and B. Carlsson, "Preventing Privacy-Invasive Software Using Online Reputations", to be submitted for publication at the *7th Workshop on Privacy Enhancing Technologies (PET2007)*.

J. Wieslander, M. Boldt and B. Carlsson, "Investigating Spyware on the Internet", in the *proceedings of the 7th Nordic Workshop on Secure IT Systems (NordSec03)*, Gjövik Norway, 2003.

Table of Contents

List of Figures xi

List of Tables xiii

Chapter 1 1
Introduction

1.1 Thesis Outline 2

Chapter 2 5
Spyware

2.1 Retrospective 5
2.2 Central Concepts 8
2.2.1 Privacy 8
2.2.2 Adware 9
2.2.3 Malware 10
2.2.4 Spyware 11
2.2.5 Informed Consent 13
2.3 Spyware and Informed Consent 14
2.4 Spyware Distribution 16
2.5 Spyware Implications 17
2.6 Spyware Countermeasures 19
2.7 Future Spyware Prediction 21

Chapter 3 23
Research Approach

3.1 Motivation and Research Questions 23
3.2 Research Methods 24
3.3 Thesis Contribution 25
3.3.1 Research Question 1 25
3.3.2 Research Question 2 26
3.3.3 Research Question 3 28
3.4 Discussion and Future Work 30
3.5 References 34

Chapter 4 39
Privacy-Invasive Software in File-Sharing Tools

4.1 Introduction 40

4.2 Privacy-Invasive Programs and their Implications 41
4.3 Experiment Design 45
4.3.1 Problem Domain. 45
4.3.2 Instrumentation and Execution 46
4.3.3 Data Analysis. 47
4.4 Experiment Results and Analysis 49
4.4.1 Ad-/Spyware Programs in File-Sharing Tools 49
4.4.2 The Extent of Network Traffic 50
4.4.3 The Contents of Network Traffic 52
4.5 Discussion 53
4.6 Conclusions 55
4.7 References 56

Chapter 5. 59
Exploring Spyware Effects

5.1 Introduction 60
5.2 On Spyware 62
5.2.1 The Background of Spyware 62
5.2.2 The Operations of Spyware 63
5.2.3 The Types of Spyware 64
5.2.4 On the Implications of Spyware. 66
5.3 Experiments 67
5.3.1 Method 67
5.3.2 Results and Analysis 69
5.4 Discussion 72
5.5 Conclusions 76
5.6 References 76

Chapter 6. 79
Analysing Countermeasures Against Privacy-Invasive Software

6.1 Introduction 79
6.2 Countermeasures 81
6.3 Computer Forensics. 82
6.4 Investigation 83
6.5 Results. 86
6.6 Discussion 90
6.7 Conclusions 92
6.8 References 93

Chapter 7. 97
Privacy-Invasive Software and Preventive Mechanisms

7.1 Introduction 98

7.2 Spyware and User Consent 99
7.3 Software Classifications 101
7.3.1 Spyware Classification 101
7.3.2 PIS Classification 102
7.4 PIS Countermeasures 106
7.4.1 Software Deeds 107
7.4.2 Software Preferences 107
7.4.3 Third Party Software Certification 108
7.4.4 Collaborative Reputation Systems 108
7.5 Discussion 109
7.6 Conclusions 112
7.7 References 112

List of Figures

1.1 Thesis outline. 2
4.1 Amount of programs in the experiment sample 48
4.2 Network data traffic. 51
6.1 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. 87

List of Tables

4.1 Identified ad-/spyware programs. 49
5.1 Identified Spyware Programs. 70
5.2 Resource Utilisation Measurements. 71
5.3 Spyware Effects. 73
6.1 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. 86
6.2 Number of PIS in three different P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions 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. 88
6.3 Total number of undiscovered PIS programs in three different P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions (3.5 to SE1.06) of Ad-Aware. 88
6.4 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. 89
7.1 Classification of spyware with respect to user awareness and permission (high or low) and user consequences (positive or negative). 102
7.2 Classification of privacy-invasive software with respect to user's informed consent (high, medium and low) and negative user consequences (negligible, moderate andsevere). 103
7.3 Difference between legitimate software and malware with respect to user's informed consent and negative user consequences. 111

CHAPTER

1

Introduction

As computers are being increasingly more integrated into our daily lives, we entrust them with sensitive information, such as online banking transactions. If this data was to escape our control, negative effects to both our *privacy* and our economic situation could be impaired. Privacy is a central concept in this work, and it could be described as the ability for individuals to control how personal data about themselves are stored and disseminated by other parties [61]. Another important aspect of privacy is the individuals' right to keep their lives and personal affairs out of the public space. The amount of personal data that affect our privacy will continue to grow as larger parts of our lives are represented in a digital setting, including for instance e-correspondence and e-commerce transactions.

In parallel with this development, a new type of software known as *spyware* has emerged. The existence of such software is based on the fact that information has value. Spyware benefit from the increasing personal use of computers by stealing privacy-sensitive information, which then is sold to third parties. Conceptually, these programs exist in-between legitimate software and malicious software (e.g. computer viruses). As an effect, there does not exist an agreed and precise definition for spyware since its exact borders have not yet been revealed. The lack of such a standard definition results in that spyware countermeasures do not offer users an accurate and efficient protection. Therefore, users' computers are infested with spyware that, among many things, deteriorates the performance and stability of their computers, and ultimately presents a threat to their privacy.

In this work, we contribute to the area of spyware by providing a classification of various types of *privacy-invasive software* (PIS). This classification does not only include spyware, but also both legitimate and malicious software. As there are no commonly agreed borders neither between legitimate software and spyware nor between spyware and malicious software, it is important to address both of these cases in the classification of PIS. After having classified PIS, we further explore how PIS programs affect users' computer systems and privacy. To help mitigate the effects from PIS we propose the use of *collaborative reputation systems* for preventing the infection and distribution of PIS. We have developed a proof-of-concept system for allowing users to share their opinions about software they commonly use. By using this system, users are asked to continuously grade software that they frequently use. In return, the user is presented with all previous users' opinions on software that is about to enter their own computer. Provided with this information the user can make a more informed decision on whether the software in question should be allowed to install on the computer or not.

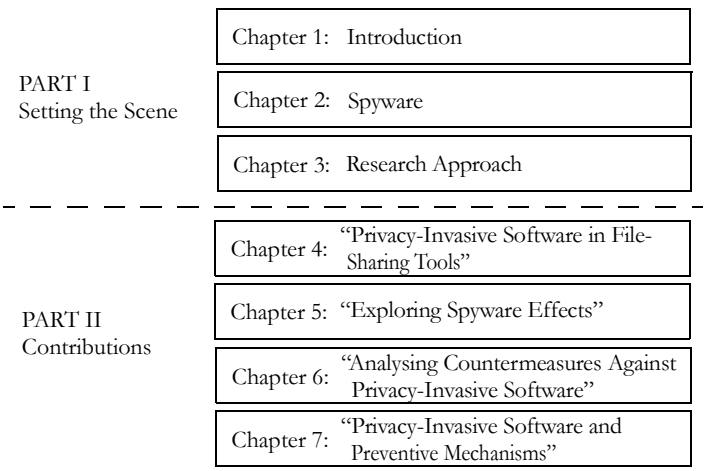

Figure 1.1 Thesis outline.

1.1 Thesis Outline

As presented in Figure 1.1, this thesis consists of two parts, where the purpose of part one is to set the scene for the thesis, using the next two chapters. In Chapter 2, we present related work and pro-

vide an extended introduction to spyware, and its central concepts. Chapter 3 describes the research approach, including the research motivation, research questions, and thesis contributions.

Four publications on spyware research in progress constitutes part two. The first two publications focus on spyware and its consequences to both the infested computer and the users' privacy. In the third publication we evaluate the accuracy of spyware countermeasures. The last included publication includes both a classification of PIS and an exploration of preventive countermeasures.

CHAPTER

2

Spyware

2.1 Retrospective

In the mid-1990s, the development of the Internet increased rapidly due to the interest from the general public. One important factor behind this accelerating increase was the 1993 release of the first browser, called Mosaic [1]. This marked the birth of the graphically visible part of the Internet known as the World Wide Web (WWW). Commercial interests became well aware of the potential offered by the WWW in terms of electronic commerce, and soon companies selling goods over the Internet emerged, i.e. pioneers such as book dealer Amazon.com and CD retailer CDNOW.com, which both were founded in 1994 [40].

During the following years, personal computers and broadband connections to the Internet became more commonplace. Also, the increased use of the Internet resulted in that e-commerce transactions involved considerable amounts of money [11]. As competition over customers intensified, some e-commerce companies turned to questionable methods in their battle to entice customers into completing transactions with them [10, 47]. This opened ways for illegitimate actors to gain revenues by stretching the limits used with methods for collecting personal information and for propagating commercial advertisements. Buying such services allowed for some e-commerce companies to get an advantage over their competitors, e.g. by using advertisements based on unsolicited commercial messages (also known as spam) [30].

Such questionable techniques were not as destructive as the more traditional malicious techniques, e.g. computer viruses or trojan horses. Compared to such malicious techniques the new ones differed in two fundamental ways. First, they were not necessarily illegal, and secondly, their main goal was gaining money instead of creating publicity for the creator by reaping digital havoc. Therefore, these techniques grouped as a "grey" area next to the already existing "dark" side of the Internet.

Behind this development stood advertisers that understood that Internet was a "merchant's utopia", offering huge potential in global advertising coverage at a relatively low cost. By using the Internet as a global notice board, e-commerce companies could market their products through advertising agencies which delivered online ads to the masses. In 2004, online advertisement yearly represented between $500 million and $2 billion markets, which in 2005 increased to well over $6 billion-a-year [34, 63]. The larger online advertising companies report annual revenues in excess of $50 million each [14]. In the beginning of this development such companies distributed their ads in a broadcast-like manner, i.e. they were not streamlined towards individual users' interests. Some of these ads were served directly on Web sites as banner ads, but dedicated programs, called *adware*, soon emerged. Adware used to display ads through pop-up windows without depending on any Internet access or Web pages.

In the search for more effective advertising strategies, these companies soon discovered the potential in ads that were *targeted* towards user interests. Once targeted online ads started to appear, the development took an unfortunate turn. Now, some advertisers developed software that became known as *spyware*, collecting users' personal interests, e.g. through their browsing habits. Over the coming years spyware would evolve into a significant new threat to Internet-connected computers, bringing along reduced system performance and security. The information gathered by spyware were used for constructing user profiles, including personal interests, detailing what users could be persuaded to buy.

The introduction of online advertisements also opened a new way to fund software development by having the software display advertisements to its users. By doing so the software developer could offer their software "free of charge", since they were paid by the advertising agency. Unfortunately, many users did not understand the difference between "free of charge" and a "free gift". The dif-

ference is that a free gift is given without any expectations of future compensation, but something provided free of charge expects something in return. A dental examination that is provided free of charge at a dentist school is not a free gift. The school expects gained training value and as a consequence the customer suffers increased risks. As adware were combined with spyware, this became a problem for computer users. When downloading software described as "free of charge" the users had no reason to suspect that it would report on for instance their Internet usage, so that presented advertisements could be targeted towards their interests.

Some users probably would have accepted to communicate their browsing habits because of the positive feedback, e.g. "offers" relevant to their interests. However, the fundamental problem was that users were not properly informed about neither the occurrence nor the extent of such monitoring, and hence were not given a chance to decide on whether to participate or not. As advertisements became targeted, the borders between adware and spyware started to dissolve, combining both these programs into a single one, that both monitored users and delivered targeted ads. The fierce competition soon drove advertisers to further "enhance" the ways used for serving their ads, e.g. replacing user-requested content with sponsored messages instead, before it were shown to the users.

As the chase for faster financial gains intensified, several competing advertisers turned to use even more illegitimate methods in an attempt to stay ahead of their competitors [9]. This accelerated the whole situation and pushed the "grey" area of the Internet closer and closer to the "dark" side [27]. During this development users experienced infections from unsolicited software that crashed their computers by accident, uninvitedly changed application settings, harvested personal information, and deteriorated their computer-experience through spam and pop-up ads [37]. Over time these problems lead to the introduction of countermeasures in the form of *anti-spyware* tools. These tools supported users in cleaning their computers from spyware, adware, and any other type of shady software located in that same "grey" area. As these tools were designed in the same way as anti-malware tools, such as anti-virus programs, they could only identify spyware that were already known, leaving previously unknown spyware undetected. To further aggravate the situation, a few especially illegitimate companies distributed *fake* anti-spyware tools in their search for a larger piece of the online advertising market. These fake tools claimed to remove spyware,

but instead installed their own share of adware and spyware on unwitting users' computers. Sometimes even accompanied by the functionality to remove adware and spyware from competing vendors.

As this thesis is being written the spyware situation is evolving in favour for the distributors of spyware. New spyware programs are being added to the setting in what seams to be a never-ending stream, although the increase has levelled out over the last years. However, there still does not exist any consensus on a common spyware definition or classification, which we believe negatively affect the accuracy of anti-spyware tools, further rendering in that spyware programs are being undetected on users' computers [26, 33]. Developers of anti-spyware programs officially state that the fight against spyware is more complicated than the fight against viruses, trojan horses, and worms [59]. We believe the first step for turning this development in favour for both users and anti-spyware vendors, is to create a standard classification of spyware. Once such a classification exists anti-spyware vendors can make a more clear separation between legitimate and illegitimate software, which result in more accurate countermeasures.

In the next section we discuss central concepts in this thesis, before moving to a further detailed description of spyware.

2.2 Central Concepts

The concepts that are covered in this section form a base, for the further work and discussions in this thesis. Since spyware is rather unexplored in the academic community, it should be pointed out that some of the concepts below unfortunately lack complete definitions. In the end, the purpose of this section is to declare our understanding and motivate the usage of the concepts in this thesis.

2.2.1 Privacy

The first definition of privacy was presented by Warren and Brandeis in their work "The Right to Privacy" in 1890 [57]. In their work, they define privacy as "the right to be let alone". Today, as we are being parts of complex societies, the privacy debate does not argue for the individual's right to physically isolate himself by living alone in the woods as a recluse, which could have been one main motivation a century ago. Instead the community presume that we

all must share some personal information so that our society to work properly, e.g. in terms of health care services and law enforcement. Discussions in the privacy community therefore focus on how, and to what extent users should share their personal information in a privacy respecting manner. Unfortunately, it is not possible to properly define privacy in a single sentence in this complex situation, or as Simson Garfinkel so concisely put it [23]:

> *"The problem with the word privacy is that it falls short of conveying the really big picture. Privacy isn't just about hiding things. It's about self-possession, autonomy, and integrity. As we move into the computerized world of the twenty-first century, privacy will be one of our most important civil rights."*

However, for the clarity of the remaining part of this work we make an approach to present our interpretation and usage of privacy in this thesis. In the end, we share the general understanding of privacy with the work presented by Simone Fischer-Hübner [28]. She divides the concept of privacy into the following three areas:

- *territorial privacy* focusing on the protection of the public area surrounding a person, such as the workplace or the public space
- *privacy of the person* which protect the individual from undue interference that constitute for instance physical searches and drug tests
- *informational privacy* protecting if and how personal information (information related to an identifiable person) is being gathered, stored, processed, and further disseminated.

Since this thesis has its origin in a computer setting we interpret the above areas into this setting. This is motivated since computers are being increasingly more weaved together with our daily lives which affect the individual's privacy. The problems analysed and discussed in this work are mostly related to the last two areas above, i.e. protecting the user from undue interference, and safeguarding users personal information, both while using computers. Our view of privacy does not only focus on the communication of personal information, but also include undue interference that affect the users' computer experience.

2.2.2 Adware

Adware is a concatenation of *advertising* and *software*, i.e. programs set to display ads delivered by advertising agencies, which are shown on

the computer users' screen. Throughout this thesis we use the following definition of adware [30]:

> *"Adware is a category of software that displays (commercial) advertisements, often tuned to the user's interests."*

2.2.3 Malware

Malware is a concatenation of *malicious* and *software*. Within the concept of malware lies any software that are designed or distributed with malicious intent towards users. The distribution of malware has intensified over the last decade as a result of the widespread use of the Internet. Another contributing factor is the mix between data and executable code in commonly used systems today. In these systems, executable code has found its way into otherwise traditionally pure data forms, e.g. Word documents, Web sites, and even music files and Jpeg images. The risk of malware infection follows in all these locations where executable code is being incorporated. Throughout this thesis we use the following definition of malware [50, 54]:

> *"Malware is a set of instructions that run on your computer and make your system do something that an attacker wants it to do."*

Spyware are often regarded as a type of malware, since they (in accordance with the malware definition) executes actions that are defined by the developer. However, there are differences between spyware and malware which we further explain when defining spyware below. To further enlight the reader, and as a way to exemplify, we include three definitions of malware types that often are being mixed-up in for instance media coverage. We start with the *computer virus* which probably is most publicly recognized malware type[50]:

> *"A virus is a self-replicating piece of code that attaches itself to other programs and usually requires human interaction to propagate."*

The second one is the *worm*, also publicly known through its global epidemics [54]. Although it is closely related to and often mixed-up with the computer virus, there exist some differences as shown in the definition [50]:

> *"A worm is a self-replicating piece of code that spreads via networks and usually doesn't require human interaction to propagate."*

The third malware type is the *Trojan horse,* which share some similarities with spyware as they deceive users by promising one thing but also delivers something different according to their operator's desires [50]:

> *"A trojan horse is a program that appears to have some useful or benign purpose, but really masks some hidden malicious functionality."*

One common misconception is that viruses or worms must include a payload that carry out some malicious behaviour. However, this is not the case since these threats are categorized by their distribution mechanisms, and not by their actions. An interesting example are the so called "white" or "ethical" worms that replicate instantly fast between computers, patch the hosts against security vulnerabilities, i.e. they are not set to spread destruction on the hosts they infect but instead help them protect against future threats. One could wonder if it is possible to "fight fire with fire without getting burned" [50]. Most security experts would agree in that these "white" worms are not ethical but instead illegal, as they affect computer systems without the owners consent. Such an ethical worm could harm a system if it were to include a programming bug that gave it another behaviour than intended, i.e. similar to what happened with the Morris worm [18]. Since various malware definitions does not say anything about the purpose of the attacker, they can not easily be related to spyware as these programs are classified according to their actions instead of their distribution mechanisms.

2.2.4 Spyware

In early 2000, Steve Gibson formulated the first description of spyware after realizing software, that stole his personal information, had been installed on his computer [24]. His definition reads as follows:

> *"Spyware is any software which employs a user's Internet connection in the background (the so-called 'backchannel') without their knowledge or explicit permission."*

This definition was valid in the beginning of the spyware evolution. However, as the spyware concept evolved over the years it attracted new kinds of behaviours. As these behaviours grew both in number and in diversity, the term spyware became hollowed out. This evolution resulted in that a great number of synonyms sprang up, e.g. thiefware, evilware, scumware, trackware, and badware. We believe

that the lack of a single standard definition of spyware depends on the diversity in all these different views on what really should be included, or as Aaron Weiss put it [60]:

> *"What the old-school intruders have going for them is that they are relatively straightforward to define. Spyware, in its broadest sense, is harder to pin down. Yet many feel, as the late Supreme Court Justice Potter Stewart once said, 'I know it when I see it.'."*

Despite this vague comprehension of the essence in spyware, all descriptions includes two central aspects. The degree of associated user consent, and the level of negative impact they impair on the user and their computer system. These are further discussed in Section 2.3 and Section 2.5 respectively. Because of the diffuse understanding in the spyware concept, recent attempts to define it has been forced into compromises. The Anti-Spyware Coalition (ASC) which is constituted by public interest groups, trade associations, and anti-spyware companies, have come to the conclusion that the term spyware should be used at two different abstraction levels [2]. At the low level they use the following, which is similar to Steve Gibson's original definition:

> *"In its narrow sense, Spyware is a term for tracking software deployed without adequate notice, consent, or control for the user."*

However, since this definition does not capture all the different types of spyware available they also provide a wider definition, which is more abstract in its appearance:

> *"In its broader sense, spyware is used as a synonym for what the ASC calls 'Spyware (and Other Potentially Unwanted Technologies)'. Technologies deployed without appropriate user consent and/or implemented in ways that impair user control over:*
> *1) Material changes that affect their user experience, privacy, or system security;*
> *2) Use of their system resources, including what programs are installed on their computers; and/or*
> *3) Collection, use, and distribution of their personal or other sensitive information."*

Difficulties in defining spyware, forced the ASC to define what they call *Spyware (and Other Potentially Unwanted Technologies)* instead. In this term they include any software that does not have the users' appropriate consent for running on their computers. Another group that

has tried to define spyware is a group called StopBadware.org, which consists of actors such as Harvard Law School, Oxford University, Google, Lenovo, and Sun Microsystems [51]. Their result is that they does not use the term spyware at all, but instead introduce the term *badware*. Their definition thereof span over seven pages, but the essence looks as follows [52]:

> *"An application is badware in one of two cases:*
> *1) If the application acts deceptively or irreversibly.*
> *2) If the application engages in potentially objectionable behaviour without: first, prominently disclosing to the user that it will engage in such behaviour, in clear and non-technical language, and then obtaining the user's affirmative consent to that aspect of the application."*

Both definitions from ASC and StopBadware.org show the difficulty with defining spyware. Throughout this thesis we regard the term spyware at two different abstraction levels. On the lower level it can be defined according to Steve Gibsons original definition. However, in its broader and in a more abstract sense the term spyware is hard to properly define, as concluded above. Throughout the rest of this chapter we presume this more abstract use of the term spyware, unless otherwise is stated. We also use the terms *illegitimate* and *questionable* software as synonyms to spyware.

One of the contributions of this thesis is our classification of various types of spyware under the term *privacy-invasive software* (PIS), which is introduced in Chapter 3. This classification was developed as a way to bring structure into the fuzzy spyware concept. However, as the PIS classification did not exist when we wrote the first two included publications we therefore use the term *ad-/spyware* in Chapter 4 and 5 instead of PIS.

2.2.5 Informed Consent

The degree of *informed consent* that is associated with software is an important and central part of spyware. Informed consent is a legal term which details that a person has understood and accepted both the facts and implications that is connected to an action. In this thesis we use the term when observing to what degree computer users comprehend that new software is installed and how it impact their computer-experience. We start by defining informed consent, before moving on to describe the relation between spyware and informed consent.

Throughout this thesis we use the same definition of *informed consent* as was originally defined by Friedman et al. [19]. This definition divide the term into the following two parts:

- *Informed*, i.e. that the user has been adequately briefed. The term informed is then further divided into *disclosure* and *comprehension*. Disclosure refers to that accurate information about both positive and negative feedback should be disclosed, without any unnecessary technical details. Comprehension targets that the disclosed information is accurately interpreted.
- *Consent*, i.e. that both positive and negative implications are transparent and approved by the user. The term consent is then broken down into *voluntariness*, *competence*, *agreement*, and *minimal distraction*. Voluntariness refers to that the individual has the possibility to decline an action if wanted, i.e. no coercion is allowed. The term competence concerns that the individual possess both the mental, emotional, and physical capabilities that are needed to give an informed consent. Agreement means that an individual should be given a clear and ongoing opportunity to accept or reject further participation. Finally, minimal distraction declare that individuals should not be diverted from their primary task through an overwhelming amount of interruptions that seek to "inform the user" or to "seek consent", i.e. to utilize user interaction sparsely [21].

For a user to be able to give an informed consent, e.g. with respect to allowing software to enter the system it is important that the implications of the software is fully transparent towards the user. Today, the main method used by software vendors to inform users of their software is not transparent as it were designed to primarily fulfil juridical purposes. End-User License Agreements (EULA) are widely used today and they form a contract between the producer and the user of a certain software. Most often users are forced to affirm that they have read, understood and accepted the EULA content before being able to install a specific software. Questionable software vendors use the EULA to escape liability from their software actions, by including juridical escape routes inside the EULA content [53].

2.3 Spyware and Informed Consent

As touched upon earlier, installing software that are funded by included spyware components allow for the vendor to distribute

their software "free of charge". However, the inclusion of such components may also result in a mismatch between the software behaviour that users assume, and the actual behaviour they realize. Such divergences have formed a sceptical user-base that disapprove of *any* software that e.g. monitor user behaviour. As a consequence, such users also label legitimate software as spyware, even if their behaviour is clearly stated in the corresponding EULA without the use of any deceptive techniques. Many computer users today are not capable of reading through EULAs, as they are written in a formal and lengthy manner [26, 53]. User license agreements that include well over 6000 words (compared to, e.g. the US Constitution that includes 4616 words) is not unusual [25]. Prior research shows that users need skills that correspond to a degree in contract law to understand the full EULA content [7]. This is used by questionable software vendors as a legal lifeline when they are challenged to explain their practices in court, using it as an escape route from liability.

Since the majority of users either do not have the prerequisite knowledge, or the time, to base an opinion on EULA content prior to installing software, they just accept it without reading it, i.e. the consent is not based on an *informed* decision. In the absence of user informed consent, software that does not comply with the user's security preferences (e.g. in terms of behaviour or stability) is allowed to enter their system. Since users lack the aiding mechanisms inside the operating system to distinguish illegitimate software from legitimate, they get their computers infested with spyware.

This lack of accurate aiding mechanisms that users could depend upon when evaluating software also result in scepticism against *all* software that for instance monitor user behaviour. Today, legitimate software vendors that, without any deceptive practices, state in the EULA that their software displays advertisement pop-ups, still run the risk of being labelled as spyware by the users, since they rarely read through the associated EULA [7]. Hence, the users can not deduce the pop-up ads on the computer screen with the approval of a software installation some time ago. So, once users think their computer-experience has been subverted by spyware, they become overly protective which further adds on this scepticism. We believe this to be very unfortunate since behavioural monitoring is both useful and an effective info-gathering measure to base tailored services towards users' individual needs [12, 41]. It is not the technology as such that is the main problem, but rather the uninformed man-

ner in which it is introduced toward the users. Legitimate software vendors need standardized mechanisms inside the operating system to inform potential users in how their software impacts the user's computer system.

If the technology was provided in a true overt manner towards the users it could equally well provide most beneficial services. Because of the personalization of these services they would also increase user benefits compared to non user-tailored services. Therefore, it is important for both software vendors and for users to safeguard users' right to make informed decisions on whether they want software to enter their system or not. In the end, we believe that an acceptable software behaviour is context-dependent, i.e. what one user regards as acceptable is regarded as unacceptable by others, and as a result only the user himself can reach such decisions [26]. This is further discussed in Section 3.3 as one of the contributions in this thesis. In the end we believe that user consent will become an increasingly more important aspect in computer security as computers are further introduced into people's daily lives, e.g. through mobile devices [43].

2.4 Spyware Distribution

Distribution of spyware differs vastly from the spreading of malware types such as viruses and worms. As by definition viruses and worms are distributed using self-propagation mechanisms, which spyware does not include.

Instead, most spyware distribution ironically is being carried out by the users themselves. Of course the users are not being aware that they install spyware because of a number of deceptive measures used by spyware vendors. One commonly used strategy is to *bundle* (piggyback) spyware with other software, which users are enticed to download and install. When users find useful software being provided free of charge they download them without questioning or being aware of the bundled components enclosed. Although the associated EULA often contain information about the bundled spyware and its implications, users do not read them because of their length and formal language. So, spyware vendors basically use software that attracts users as bait for distributing their own programs as bundles, e.g. together with file-sharing tools, games, or screen-saver programs.

Another spyware distribution mechanism relies on the exploitation of security vulnerabilities in the users' computer system. Microsoft's Web browser, Internet Explorer, has often been used for such purposes because of its unfortunate history of security flaws. By utilizing such vulnerabilities inside software on the user's computer allows attackers to run any programs of their choice on the user's system. Such attacks on Web browsers often start when the user visits, or is fooled to visit, a Web site controlled by the attacker. Next, the Web server sends a small program that exploits the security vulnerability in the user's Web browser. Once the attacker has gained this foothold, it is possible for him to deploy and start any software of his desire, for instance sponsored spyware programs. Because the users are kept totally out of this scenario without any choice for themselves, these installations go under the name drive-by downloads. For clarity, it should be added that spyware that rely on software vulnerabilities as a distribution mechanism are closely related to malware. It might even be the case that these programs should not be called spyware, but instead malware.

The third method used by spyware vendors is to distribute their software using tricks that deceive the user into manipulating security features that are designed to protect the user's computer from undesired installations. Modern Web browsers for example does not allow software to be directly installed from remote Web sites unless the user initiates the process by clicking on a link. With the use of deceptive tricks, spyware vendors manipulate users into unknowingly clicking on such links [35]. One example is that pop-up ads could mimic the appearance of a standard window dialog box which include some attractive message, i.e. "Do you want to remove a new spyware threat that has been detected on your computer?". This dialog box could also include two links that are disguised as buttons, reading "Yes" and "No", and despite which button the user press the drive-by download is started.

2.5 Spyware Implications

As we have seen, many spyware programs are distributed by being bundled together with attractive programs. When users install such programs the bundled spyware follows, and with it, system implications. As touched upon previously, these spyware exists in a grey area between legitimate software and traditional malware. One of the distinctions between the two software categories relate to their

implications on systems. Spyware does not result in the same direct destruction as with traditional forms of malware. Instead users experience a gradual performance, security, and usability degradation of their computer system. These system effects could be structured as follows [3, 47, 49]:

- *Security implications:* As with any software installation, spyware introduces system vulnerabilities when deployed on computer systems. However, the fundamental difference between general software installation and spyware, is the undisclosed fashion used by the latter. This covertness renders it virtually impossible for system owners to guarantee the software quality of their computer system. Poor software quality conveys an escalated risk of system vulnerabilities being exploited by remote malicious actors. If such a vulnerability was found and exploited inside one of the leading spyware programs, it could result in that millions of computers were controlled by attackers because of the widespreadness of these programs. In 2004, poorly written adware programs allowed remote actors to replace any files on users systems because of a deficiently designed update function [42]. Fortunately enough, this vulnerability was first identified by an honest individual that made sure that the adware developer corrected the problem before making a public announcement about the vulnerability.
- *Privacy implications:* Spyware covertly monitors, communicates, and refines personal information, which makes it privacy-invasive. In addition, such programs also displays ads and commercial offers in an aggressive, invasive, and many times undesirable manner. Such software behaviour negatively affects both the privacy and computer-experience of users [60, 63]. These privacy-invasions will probably render in greater implications for the users as computers are being increasingly more used in our daily lives, e.g. when shopping or carrying out online banking errands.
- *Computer capacity consumption*: As spyware is installed on users' computer systems in an uninformed way, the memory, storage, and CPU resources are being utilized without the users' permission. Combined with that users commonly have several instances of spyware on their systems makes the cumulative effect on computer capacity evident. Another threat to the local computation capacity comes from spyware that "borrow" the storage and computation resources from users' computers which it has infected. This combined storage and computational power were then combined into a distributed super computer,

which could be rented by the highest bidder. Again, unwitting users (after some time) found their computers being covertly used in projects that were not compatible with their opinions and ethics [15].

- *Bandwidth consumption*: In the same line of reasoning as above, the users network capacity is being negatively affected by the continuous transmission of ads and personal information. Some users might even be even more upset, if these highly irritating and undesired behaviours use resources that instead should be used for really important tasks. Bandwidth over consumption becomes even more significant when ads are being further enhanced using moving pictures and three-dimensional graphics.
- *System usability reduction*: The existence of spyware on computer systems negatively impact a user's computer-experience [26]. The covert manner in which spyware is installed render in that users do not know what is the cause of the strange system behaviour they are experiencing. This makes it hard to identify what is inducing for instance the flow of pop-up ads, irreversible changes in application settings, installation of unrequested and unremovable software, or degradation of system performance and stability. In addition to this, underaged users could be exposed to offending material such as ads promoting adult material. These implications further result in that users are interrupted in their daily work, negatively influencing their general computer-experience.

As the aggregated amount of these implications became too overwhelming for the users to bear, a new group of software labelled *spyware countermeasures* emerged. These tools helped users to remove spyware from their systems.

2.6 Spyware Countermeasures

Today, *spyware countermeasures* are being implemented using the same techniques as traditional anti-malware tools use, e.g. anti-virus programs. However, an important difference between malware and spyware is that the former is well defined, while there is a lack of both knowledge and definition of the latter. Without a clear understanding of what kinds of programs that should be removed, countermeasure vendors both miss some spyware and wrongly remove legitimate software. The key problem is that malware include pro-

hibited behaviour, such as virus and worm propagation mechanisms, while spyware does not. Anti-malware tools can therefore in an easier manner separate malware from legitimate software, by focusing on malware's illegal behaviours.

Spyware, on the other hand, often does not include prohibited behaviour, but instead compared with malware, rather innocent behaviours, e.g. displaying messages on the screen, monitoring of the Web address field in browsers, or making non-critical configuration changes to programs, such as altering the default Web page. Unfortunately enough for anti-spyware vendors, spyware share these behaviours with a vast number of legitimate software in general. Anti-spyware vendors therefore face a problem when trying to distinguish spyware from legitimate software based on the software behaviour [58]. The anti-spyware vendors' removal strategies therefore need to be placed on a sliding scale, between two extremes. Either they prioritize the safeguarding of legitimate software, or they focus on removing every single spyware out there. Unfortunately for the users, it is neither possible to remove every single spyware, because this would include many legitimate programs as well, nor to safeguard all legitimate software since this leaves most spyware untouched. Today, anti-spyware vendors have great difficulties in choosing where on this sliding scale they want to be, as none of these alternatives are very effective. Therefore the chosen strategy needs to be a compromise between these two extremes, rendering in both missed spyware programs and false labelling of legitimate software as spyware. In a prolongation, anti-spyware vendors need to chose to either miss spyware components, resulting in bad reputation, or to include legitimate software which lead to law suits.

This results in an arbitrariness for spyware vendors when deciding what software to label as spyware and what not. Further, leading to a divergence between what software different countermeasure vendors target, i.e. some countermeasures remove one program while others leave it untouched. These difficulties has further proved to result in legal disputes as software vendors feel unfairly treated by countermeasure vendors and therefore bring the case to court [26]. Such a situation is negative for both legitimate software vendors that find their products falsely labelled as spyware, anti-spyware vendors that are sued when trying to protect their users' interests. This further results in that users' success rate in countering spyware depends on the *combination* of different countermeasure tools being used, since no single one offers full protection.

Current spyware countermeasures depend on their own classifications of what software that should be regarded as spyware. We believe that this model provides a too coarse mechanism to accurately distinguish between the various types of spyware and legitimate software that exist, since this is based on the individual users' own opinion. Most of the current spyware countermeasures are reactive and computer-oriented in their design, i.e. they focus on system changes to identify known spyware once they *already* have infected systems[1]. Over the last years, some preventive countermeasures have also started to emerged which focus on hindering spyware *before* they have any chance to start executing on the computer. However, such countermeasures still suffer from the issues connected to the per vendor governed spyware classifications. Each vendor has its own list of what software that should be regarded as spyware and these lists do not correlate.

We argue that there is a need for more user-oriented countermeasures, which should complement the existing computer-oriented anti-malware tools. Such complementing countermeasures should focus on informing users when they are forced to reach difficult trust decisions, e.g. whether to install a certain software or not. However, the goal for such mechanisms should *not* be to make these trust decisions for users. In the end, it is up to the users themselves to consider advantages and disadvantages before reaching the decision.

2.7 Future Spyware Prediction

There are several trends integrating computers and software into people's daily lives. One example is traditional media-oriented products which are being integrated into a single device, called *media centres*. These media centres include the same functionality as conventional television, DVD-players, and stereo equipment, but combined with an Internet connected computer. In a foreseeable future these media centres are anticipated to reach vast consumer impact [29, 36]. In this setting, spyware could monitor and surveillance for instance what television channels are being watched, when/why users swap channel or what DVD movies users have purchased and watch. This is information that is highly attractive

1. Further information about spyware countermeasures is described in Chapter 6.

for any advertising or media-oriented corporation to obtain. This presents us with a probable scenario where spyware is tailored towards these new platforms; the technology needed is to a large extent the same as is used in spyware today.

Another interesting area for spyware vendors is the increasing amount of mobile devices being shipped. Distributors of advertisements have already turned their eyes to these devices. So far this development have not utilized the geographic position data stored in these devices. However, during the time this thesis is finalized companies are working on GPS-guided ads and coupons destined for mobile phones and hand-held devices [8]. In other words, development of location-based marketing that allow advertising companies to get access to personal geographical data so that they can serve geographically dependant ads and coupons to their customers. Once such geographic data is being harvested and correlated with already accumulated personal information, another privacy barrier has been crossed.

Finally, to counteract these new threats we predict the widespread use of more user-oriented countermeasures. These tools should focus on informing users as they are being confronted with difficult trust decisions. We further anticipate that such countermeasures will combine the experiences from individual users into a commonly shared knowledge-base, used in a collaborative manner. Allowing a user installing new software to be provided with the accumulated knowledge, or a selected subset thereof, from all other users that previously have experienced that specific software, i.e. aiding them when reaching the installation decision.

CHAPTER

3

Research Approach

3.1 Motivation and Research Questions

We believe that study of spyware and its associated countermeasures form an interesting research conjunction between technology, law, and human-computer interaction (HCI). Even though spyware is interesting to study from several angles, we will keep a technical focus in this thesis, but we will also occasionally touch upon the other areas as well. Academic research in spyware has been rather sparse, even parsimonious in relation to the degree of negative impact these programs currently have on users' computer experiences [37]. Today, the occurrence of illegitimate software has become a major security issue for both corporations and home users on the Internet, negatively affecting millions of users daily. As we migrate into an increasingly more computerized life, it will be of great importance to manage the problems associated with questionable software so that the integrity and control of users' computers can be protected. However, since no accurate definition or classification exists for such software, the reports and discussions of their effects are often vague and sometimes inconsistent. Although previous work shows that illegitimate software invades user privacy, disrupt the user's computer experience, and deteriorates system performance and security, one could wonder what actually is being measured. That such illegitimate software pose real-world problems have been known for some time, but their level of magnitude have not been thoroughly investigated.

Today, several countermeasures against questionable software exist, but most of them use a reactive rather than a preventive approach, i.e. removing software once it already has found its way into the system. Even though there exist some preventive tools that lock down a system so that no software can enter unless the user allows it to, these are often difficult for non-technical users to configure and operate. Such tools result in that users need to reach security related decisions based on the insufficient information presented to them through warning and notification messages. Messages that usually include a technical or juridical language which many users find hard to interpret and therefore benefit from. These problems have motivated us to put forward the following three research questions (all assuming the more abstract use of the term spyware described in Chapter 2):

RQ1 How could a classification of spyware be formulated with respect to privacy-invasions?

RQ2 How does the installation and execution of spyware impact performance and security on computer systems?

RQ3 How could a preventive system of mechanisms against spyware be designed?

3.2 Research Methods

Because of the rather sparse knowledge available about spyware, we used an exploratory research method throughout most of the work in this thesis [3]. This approach is often used when the objects or problems being studied has not been clearly defined, and where the researcher want to find out what is happening in little-understood situations, to seek new insights or to generate ideas for future research.

We used different research methods when approaching the three research questions. Both RQ1 and RQ3 were approached through a literature review, aiming to find and understand already existing classifications and countermeasures. The outcome from the literature review was then compiled and analysed in search of both strengths and weaknesses.

To approach research question RQ2 we used a method based on experiments to evaluate a set of software bundled with spyware and

their consequences on the host system. The empirical experiments were conducted in a systematic, replicable, and logical way, and was based on data collection, data analysis and data verification. Further information about the research methods used are presented in each of the four included papers.

3.3 Thesis Contribution

The main contributions of this thesis is associated with the three research questions presented, which further investigate the classification of spyware, what consequences such software impairs on the host system, and how preventive mechanisms against spyware could be designed. We also regard the extensive description of the spyware concept presented in Chapter 2 to be one of the contributions of this thesis. Another contribution is our conclusion that it is impossible to accurately define a global spyware categorization since many of the parts are subjective in respect to the users. This further leads to the introduction of user-oriented countermeasures where the user himself needs to define software as legitimate or not, based on new aiding mechanisms. In the next three sections we respectively address the research questions.

3.3.1 Research Question 1

Previous research has identified a problem with the lack of a standard spyware definition [25]. A joint conclusion is that it is important, for both software vendors and users, that a clear separation between acceptable and unacceptable software behaviour is established [7, 48]. As we conclude in Chapter 2 the concept of spyware is difficult to capture in a short, and yet commonly agreeable definition. The reason for this is the subjective nature of many spyware programs included, which result in inconsistencies between different users beliefs, i.e. what one user regards as legitimate software could be regarded as a spyware by others. As the spyware concept came to include increasingly more types of programs, the term got hollowed out, resulting in several synonyms, such as trackware, evilware and badware, all negatively emotive. We therefore choose to introduce the term *privacy-invasive software* (PIS) to encapsulate all such software. We believe this term to be more descriptive than other synonyms without having as negative connotation. Even if we use the word "invasive" to describe such software, we believe that an invasion of privacy can be both desired and beneficial for the

user as long as it is fully transparent, e.g. when implementing specially user-tailored services or when including personalization features in software.

We used the work by Warkentins et al. (presented in Section 7.3.1) as a starting point when developing a classification of PIS, where we classify PIS as a combination between *user consent* and *direct negative consequences*. User consent is specified as either *low*, *medium* or *high*, while the degree of direct negative consequences span between *negligible*, *moderate*, and *severe*. This classification allows us to first make a distinction between legitimate software and spyware, and secondly between spyware and malicious software. All software that has a low user consent, *or* which impairs severe direct negative consequences should be regarded as malware. While, on the other hand, any software that has high user consent, *and* which results in negligible direct negative consequences should be regarded as legitimate software. By this follows that spyware constitutes the remaining group of software, i.e. those that have medium user consent or which impair moderate direct negative consequences. This classification is described in further detail in Chapter 7.

In addition to the direct negative consequences, we also introduce *indirect negative consequences*. By doing so our classification distinguishes between any negative behaviour a program has been designed to carry out (direct negative consequences) and security threats introduced by just having that software executing on the system (indirect negative consequences). One example of an indirect negative consequence is the exploitation risk of software vulnerabilities in programs that execute on users' systems without their knowledge [42]. In the end, our intention with this classification is to exclude all spyware programs, which is further described as RQ3 is addressed and new countermeasures against PIS are discussed.

3.3.2 Research Question 2

To explore the effects that PIS bring about on computer systems we conducted a number of experiment that where set to investigate PIS bundled with five leading file-sharing tools. The results showed that all file-sharing tools included PIS classified as adware, spyware, and downloaders (programs that allow for new software and/or updates to be downloaded and installed without first asking the user). All file-sharing tools also included PIS that were involved in Internet communication. It was not practically possible to further

investigate exactly what information that was transmitted over the network, since the traffic was encrypted. However, in one case our empirical results confirmed that one of these tools transmitted privacy-invasive data such as visited Web sites, zip code, country, lists of other software installed on the computer, and the exact version of the operating system. Our results also confirm that many of the PIS components introduce new security risks since they allow for new software and/or updates to be automatically downloaded and installed.

When investigating the resource utilization of PIS on a local computer we used two different versions of the same file-sharing tool, in this case KaZaa and KaZaa Lite K++. By removing the resource utilization of KaZaa Lite K++, which had all PIS components removed (only leaving the file-sharing functionality) from the original KaZaa version (which included bundled with PIS), we were able to get a measurement of the amount of resources that was consumed by PIS. The results show that both the utilization of system resources, and network bandwidth were significantly higher for KaZaa compared to the cleaned version. The increased utilization of bandwidth and number of contacted servers were due to transmission of pop-up ads, banners, and new software updates for the PIS components themselves. Although the CPU utilization was rather low at 0.48%, it is interesting that PIS introduces a 32 times increase compared to the cleaned version[1]. Also, the usage of RAM was significantly higher with a 10 time increase, leaving the original version of KaZaa at a 65MB memory usage.

In contrast to PIS supported file-sharing tools, installing a cleaned software equivalence cause marginal impact to the system and network resources. However, due to the occurrence of PIS components in file-sharing tools, users with several such applications installed simultaneously will, as a result of the aggregated activity from PIS, suffer from a continuous system and network degradation. This includes increased security and stability risks.

More information about how these experiments were designed, executed, and their results are described in Chapter 4 and Chapter 5.

1. The experiments used identical computers which included a P4 2.8Ghz processor.

3.3.3 Research Question 3

So far, developers of countermeasure tools have used the same techniques as in malware countermeasures, e.g. anti-virus programs, when fighting spyware. Although there are several similarities between spyware and malware there also exist a few profound differences. Spyware, for instance, rather includes functions that show messages on the screen or monitors visited Web behaviour, instead of more malware like behaviour. When fighting malware it is therefore possible to define a boundary between those software that are considered malware, and those that are legitimate. Further more, this could be done without risking to include any legitimate software, since they are so different from malware. However, when instead targeting spyware located closer to legitimate software than malware, it is impossible not to (incorrectly) include innocent programs. This is a problem since vendors of anti-spyware tools rely on a central classification that does not respect users' personal opinion about software. Two users may disagree on whether a certain software should be classified as spyware or not, one might think it is a free useful tool that show valuable ads and offers, while the other finds it invasive and highly irritating. This results in that miss-classifications occur as a consequence of this static division, which may further render in law suits against the vendor. In other words, these techniques are not effective against PIS [53].

Instead of merely relying on the same techniques that anti-malware tools utilize, we believe that spyware countermeasures should focus on *user consent*, when distinguishing spyware from legitimate products that are beneficially tailored toward the users' needs. Any countermeasure not doing so are either forced to label legitimate software as spyware, or miss true spyware due to the user-centred opinion of spyware. We believe that this situation has originated from the lack of a proper understanding of the spyware concept which further has made spyware a buoyant and fuzzy concept. Rendering in that spyware absorbed new program behaviours over time, which further complicated the construction of a definition. User consent constitutes the essence of our definition of PIS, since we believe it must be up to the users themselves to distinguish legitimate software from illegitimate. This is impossible for any anti-spyware tool to do since they lack the personal and subjective preferences that each user has regarding software, i.e., some users accept targeted pop-up ads as something positive while others reject it with almost religious beliefs. Although this is impossible

today, we believe it could be possible in the future with the help from user-oriented countermeasures that aid users in this process.

Since spyware does not include as disastrous behaviour as malware, current anti-spyware tools face a more complicated task trying to pinpoint spyware. Therefore future countermeasures also need to focus on informing the users, so they can distinguish legitimate and illegitimate software based on their own individual preferences. Providing users that are about to install a certain software with the knowledge from previous users of that software, could help the users get a notion of either trust or mistrust towards it. By also providing the user with additional information, such as an overall rating of the software vendor, would allow interested users to further investigate the software in question. We therefore propose the use of *collaborative reputation systems* for providing users with these services [38, 62]. Such systems could handle individual users' knowledge, and refine it into a commonly shared knowledge-base. Similar reputation systems are currently used by, e.g. IMDb.com for rating movies, and by eBay.com where users rate the performance by other parties that they have produced transactions with. The overall intention with a reputation systems is to use user ratings as a trust enabler in the system. This will be further described in Section 3.4.

It should be noted that such a reputation system against PIS is tightly connected with the PIS classification. The introduction of this type of user-oriented countermeasures would transform the classification of PIS in an important way. As users are given a tool to make informed decisions regarding the behaviour and implications of software, it is possible to apply a sharp boundary based on user consent between all software in the PIS classification. Using the added knowledge provided by the reputation system would render in that all PIS that previously have suffered from a medium user consent level, now instead would be transformed into either a high consent level (i.e. legitimate software) or a low consent level (i.e. malware). In other words, all software with medium user consent, i.e. spyware, is transformed into either legitimate software or malware in the classification. Since anti-malware tools handle all malicious and deceitful software, the information about the rest of the software could be trusted to be correct, i.e., any software using deceitful methods is regarded as malware and are treated as such. This allow users to rely the information when reaching trust decisions regarding their computer system. Another aspect of this type of countermeasure is that no single organization, company or individual is responsible for the software ratings, since these are calcu-

lated based on all votes submitted by the users. This makes it hard for dissatisfied spyware vendors to sue the developer of the countermeasure for defamation.

In conclusion, as we continuously move into an increasingly more computerized society where software plays an important role, we need more accurate methods for distinguishing legitimate software from its illegitimate counterpart. Otherwise we will experience a gradual increase in the negative consequences resulted by PIS, affecting more and more of our daily lives, e.g. mobile devices or TV and media centres. We therefore introduce the use of reputation systems which utilize user ratings of software as a mechanism to mitigate distribution of deceptive software products.

3.4 Discussion and Future Work

The research presented in this thesis has resulted in an idea of a preventive mechanism that uses a collaborative reputation system to increase user awareness about software behaviour. To evaluate the impact of such a system we have built a proof-of-concept reputation system, that could be used as a test base for evaluating how to enable informed decision-making regarding software installation and execution[2].

In a way, the proposed reputation system would use the same software reputation that users today gain from, for instance, computer-magazines and Web sites. However, one important distinction is that these sources rely on the user him-/herself to find the information, i.e. the user needs to "pull" it, while our proposed countermeasure instead use a "push" approach.

We argue that the reputation system should constitute an active part in the installation process of the operating system, allowing it to notify the user each time a previously unknown software is about to execute or install on his/her system. When such an event occurs, the execution or installation process should pause until the associated information has been gathered from the knowledge-base, and has been presented to the user. This would allow the reputation system to provide users with important information when installing or executing new software. One example could be that the software

2. More information could be found at http://www.bth.se/tek/aps/mbo.nsf/

they are about to install is developed by a vendor that is known to rely on incorrect and deceiving information for sneaking their product into users' computers.

Although such a reputation system for PIS introduces many benefits, it also includes several security issues that need to be considered. Boosting the reputation of a specific software is definitely interesting from the perspective of a PIS vendor e.g. for increasing its distribution and popularity. Another problem would be companies or users that form alliances with the goal of smearing specific software. To address these threats the reputation system should use the five techniques presented below.

1. A single user should only be allowed to cast one single vote on each specific software.
2. Secondly, users should only be allowed to vote on software that has been started on the local computer more times than a certain threshold value, or which has a total execution time that exceeds a pre defined value. This would assure that the user has used the program for some time and therefore has gained at least some modest opinion before rating it.
3. In addition to the rating of software, the system should use meta-ratings that allow users to anonymously rate other users comments about a specific software. As a consequence any user that tries to boost the reputation of a specific software by inserting deceptive information would be down-rated by other users, which further affects his/her own reputation and influence in the reputation system negatively.
4. Even though all votes should be included there should be a distinction in the amount of influence they play. The exact factor should be calculated by the other users' ratings on the user's contributions, i.e. new users would have a low influence, but if they provide the system with useful information they will become more trustworthy and will thereby gain greater influence. This idea is similar to the PageRank technique that Google.com utilizes when ranking the importance of Web sites.
5. Signing up for using the system should include non automatable procedures[3] to prevent an attacker from automatically signing up a large amount of new users. In addition to this, there should also exist a restriction in the rate that the trustworthiness for a

3. For instance, techniques similar to the character recognition schemes used by for instance Hotmail.com.

> user is allowed to increase, i.e. it should be impossible to boost a user's influence to the highest level in a short amount of time. Therefore, a user must use his account on a frequent occasion over a relatively long period of time, e.g. 12 months, to be able to earn the highest vote impact. This measure forces any antagonistic actors to invest a considerable amount of time to increase the trustworthiness of their accounts, before being able to stage an effective attack. Do note that the user still needs to receive excellent ratings for his/her participation in the system by other users, to earn a higher influence.

We believe such a reputation system would mitigate PIS by refining the individual knowledge of all users in the system into software-based reputations that are shared collectively. Both users and legitimate software vendors would benefit from such a system. The legitimate software vendors could use the system to clarify and promote what their software have been designed to do, and how it would impact the user's computer system. Users, on the other hand, would automatically receive both recommendations of useful software that has been well received by previous users, and warnings against questionable software before allowing them to install. Hopefully, this combined benefit would make the users more willing to share information about their software installations with the reputation system. The proof-of-concept system relies on that users provide a valid e-mail address together with continuous information about their experiences concerning certain software. However, it should be noted that all such information should be stored in an unlinkable format which makes it impossible for the reputation system to consolidate, for instance, all software that a specific e-mail address has ranked. The privacy issues introduced with such a system needs to be properly addressed with regard to the users, so they are willing to trust it with their personal information. It is not only necessary to develop a well functioning system, but it is also of great importance that it is designed to handle the users' information in privacy respective way. If not, the very nature of such a system could be privacy-invasive towards its own users, e.g. with respect to information leakage.

Offering users mechanisms that enhance informed decisions regarding the software installation would also increase the liability of the user. In a way, these mechanisms would transfer some of the responsibility concerned with the protection against PIS to the users themselves. So, as users are being confronted with descriptions about behaviours and consequences for PIS, they are also

assumed to assimilate and use this information in a mature and reasonable way. Based on the reputation system, it would be up to the users themselves to decide on whether or not to allow certain software to enter their system.

Computer users today face similar difficulties when evaluating software as consumers did a hundred years ago when evaluating food products. In the nineteenth century food industry, distribution of snake-oil product flourished [55]. These products claimed to do one thing, for example to grow hair, while they instead made unwitting consumer addicted to habit-forming substances like cocaine and alcohol. In 1906 the Pure Food and Drug Act was passed by the United States Congress, allowing any manufacturer not complying to the rules to be punished according to the law [32]. As a consequence the manufacturers followed these rules, allowing consumers to trust the information on the food container to be correct. Further allowing them to make informed decisions on whether they should consume a product or not, based on individual preferences such as nutritiousness, degree of fat or sugar, price, or allergies. As long as the food does not include poisonous substances or use deceptive descriptions it is up to the consumer himself to make the final decision. Although the distribution of physical snake-oil products were mitigated in 1906, its digital counterpart continue to thrive under the name spyware. An important distinction between food products and software is that the former one relies on physical factories and companies with employed personnel, which software does not. It is possible for anyone with the programming skills to produce software which then is spread globally over the Internet. Since users do not always have the option to relate the software to a physical manufacturer we believe it is important for them to instead be able to use other users' previous knowledge about the product in question, offered to them by using a reputation system.

In addition to the reputation system, we are also currently looking into the various methods that could be used for informing users about a specific software behaviour, e.g. using pictograms or digital software descriptions at multiple abstraction levels. We also plan to carry out new experiments to investigate the different computer impact that various types of PIS cause.

3.5 References

[1] M. Andreessen, "NCSA Mosaic Technical Summary", National Center for Supercomputing Applications, 1993.

[2] Anti-Spyware Coalition, "Anti-Spyware Coalition", http://www.antispywarecoalition.org, Last checked: 2006-10-05.

[3] K.P. Arnett and M.B. Schmidt, "Busting the Ghost in the Machine", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[4] M. Boldt and B. Carlsson, "Analysing Countermeasures Against Privacy-Invasive Software", in *the proceedings of the IEEE International Conference on Software Engineering Advances (ICSEA'06)*, Papeete French Polynesia 2006.

[5] M. Boldt and B. Carlsson, "Privacy-Invasive Software and Preventive Mechanisms", in *the proceedings of the IEEE International Conference on Systems and Network Communications (ICSNC'06)*, Papeete French Polynesia, 2006.

[6] M. Boldt, A. Jacobsson, and B. Carlsson, "Exploring Spyware Effects", in *proceedings of the 8th Nordic Workshop on Secure IT Systems (NordSec04)*, Helsinki Finland, 2004.

[7] J. Bruce, "Defining Rules for Acceptable Adware", in the *Proceedings of the 15th Virus Bulletin Conference*, Dublin Ireland, 2005.

[8] Business 2.0 Magazine, "20 Smart Companies to Start Now", http://money.cnn.com/magazines/business2/business2_archive/2006/09/01/8384349/index.htm?source=yahoo_quote, Last checked: 2006-10-11.

[9] B. Carlsson, "*Conflicts in Information Ecosystems - Modelling Selfish Agents and Antagonistic Groups"*, Doctoral Dissertation Thesis Series No. 2001:03, School of Engineering, Blekinge Institute of Technology, Sweden, 2001.

[10] Center for Democracy & Technology, "Following the Money", http://www.cdt.org, Last checked: 2006-05-31.

[11] C. Abhijit, J.P. Kuilboer, *"E-Business & E-Commerce Infrastructure: Technologies Supporting the E-Business Initiative"*, McGraw Hill, 2002.

[12] R.K. Chellappa and R.G. Sin, "Personalization versus Privacy: An Empirical Examination of the Online Consumer's Dilemma", in the *ACM Information Technology and Management*, Volume 6, Issue 2, 2005.

[13] E. Chien, "Techniques of Adware and Spyware", in the *Proceedings of the 15th Virus Bulletin Conference*, Dublin Ireland, 2005.

[14] C|NET Anti Spyware Workshop, "The Money Game: How Adware Works and How it is Changing", San Francisco CA, 2005.

[15] C|NET News.com, "Stealth P2P Network Hides Inside KaZaa", http://news.com.com/2100-1023-873181.html, Last checked: 2006-10-16.

[16] L.F. Cranor, "Giving Notice: Why Privacy Policies and Security Breach Notifications aren't Enough", in *IEEE Communications Magazine*, Volume 43, Issue 8, 2005.

[17] P.M. Doney and J.P. Cannon, "An Examination of the Nature of Trust in Buyer-Seller Relationships", in the *Journal of Marketing*, Volume 61, 1997.

[18] M.W. Eichin and J. Rochlis, "With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988", in the *Proceedings of the 1989 IEEE Computer Society Symposium on Security and Privacy*, Oakland Ohio, 1989.

[19] B. Friedman, E. Felten, and L.I. Millett, "Informed Consent Online: A Conceptual Model and Design Principles", *CSE Technical Report*, University of Washington, 2000.

[20] B. Friedman, P.H. Kahn, and D.C. Howe, "Trust Online", in the *Communications of the ACM*, Volume 43, Issue 12, 2000.

[21] S. Furnell et al., "Considering the Usability of End-User Security Software", in the *Proceedings of the 21st International Information Security Conference (Sec2006)*, Karlstad Sweden, 2006.

[22] S. Ganesan, "Determinants of Long-Term Orientation in Buyer-Seller Relationships", in the *Journal of Marketing*, Volume 58, 1994.

[23] S. Garfinkel, *"Database Nation"*, O'Reilly & Associates, Sebastopol CA, 2001.

[24] Gibson Research Corporation, "OptOut - Internet Spyware Detection and Removal", http://www.grc.com/optout.htm, Last checked: 2006-10-05.

[25] N. Good et al., "Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware", in the *proceedings of the Symposium On Usable Privacy and Security (SOUPS 2005)*, Pittsburgh USA, 2005.

[26] N. Good et al., "User Choices and Regret: Understanding Users' Decision Process about Consentually Acquired Spyware", in *I/S: A*

Journal of Law and Policy for the Information Society, Volume 2, Issue 2, 2006.

[27] S. Görling, "An Introduction to the Parasite Economy", in the *proceedings of EICAR*, Luxembourg, 2004.

[28] S. Fischer-Hübner, *"IT-Security and Privacy: Design and Use of Privacy-Enhancing Security Mechanisms"*, Springer Verlag, Berlin Heidelberg, 2001.

[29] International Consumer Electronics Association, http://www.cesweb.org, Last checked: 2006-03-10.

[30] A. Jacobsson, "*Exploring Privacy Risks in Information Networks"*, Licentiate Thesis Series No. 2004:11, School of Engineering, Blekinge Institute of Technology, Sweden, 2004.

[31] A. Jacobsson, M. Boldt and B. Carlsson, "Privacy-Invasive Software in File-Sharing Tools", in *proceedings of the 18th IFIP World Computer Congress (WCC2004)*, Toulouse France, 2004.

[32] Landmark Document in American History, "Pure Food and Drug Act of 1906", http://coursesa.matrix.msu.edu/~hst203/documents/pure.html, Last checked: 2006-10-16.

[33] Malware-Test Lab, "AntiSpyware Comparison Reports", http://www.malware-test.com/antispyware.html, Last checked: 2006-12-06.

[34] P. McFedries, "The Spyware Nightmare", in *IEEE Spectrum*, Volume 42, Issue 8, 2005.

[35] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy, "A Crawler-based Study of Spyware on the Web", in the *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS 2006)*, San Diego CA, 2006.

[36] M.W. Newman et. al., "Recipes for Digital Living", in *IEEE Computer*, Vol. 39, Issue 2, 2006.

[37] Pew Internet & American Life Project, "The Threat of Unwanted Software Programs is Changing the Way People use the Internet", http://www.pewinternet.org, Last checked: 2006-10-16.

[38] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architechture for Collaborative Filtering of

Netnews", in *Proceedings of ACM Conference on Computer Supported Cooperative Work*, Chapel Hill NC, 1994.

[39] C. Robson, "*Real World Research*", 2nd edition, Blackwell Publishing, Ltd., Oxford UK, 2002.

[40] R.S. Rosenberg, "*The Social Impact of Computers*", 3rd edition, Elsevier Academic Press, San Diego CA, 2004.

[41] P.E. Sand, "The Privacy Value", in *I/S: A Journal of Law and Policy for the Information Society*, Volume 2, Issue 2, 2006.

[42] S. Saroiu, S.D. Gribble, and H.M. Levy, "Measurement and Analysis of Spyware in a University Environment", in *Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, 2004.

[43] F.B. Schneider, "The Next Digital Divide", in the *IEEE Security & Privacy*, Vol. 2, Issue 1, 2004.

[44] B. Schneiderman, "Designing Trust into Online Experiences", in the *Communications of the ACM*, Volume 43, Issue 12, 2000.

[45] B. Schneier, "Inside Risks: Semantic Network Attacks", in *Communications of the ACM*, Volume 43, Issue 12, 2000.

[46] K.B. Sheehan and M.G. Hoy, "Dimensions of Privacy Concern among Online Consumers", in the *Journal of Public Policy & Marketing*, Volume 19, Issue 1, 2000.

[47] S. Shukla and F. F. Nah, "Web Browsing and Spyware Intrusion", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[48] J.C. Sipior, "A United States Perspective on the Ethical and Legal Issues of Spyware", in *Proceedings of 7th International Conference on Electronic Commerce*, Xi'an China, 2005.

[49] J.C. Sipior, B.T. Ward, and G.R. Roselli, "A United States Perspective on the Ethical and Legal Issues of Spyware", in the *proceedings of the 7th International Conference on Electronic Commerce (ICEC 2005)*, Xi'an China, 2005.

[50] E. Skoudis, "*Malware - Fighting Malicious Code"*, Prentice Hall PTR, Upper Saddle River NJ, 2004.

[51] StopBadware.org, "StopBadware.org", http://www.stopbadware.org, Last checked: 2006-10-05.

[52] StopBadware.org, "Software Guidelines", http://www.stopbadware.org/home/guidelines, Last checked: 2006-10-05.

[53] Spyware: Research, Testing, Legislation, and Suits, http://www.benedelman.org/spyware/, Last checked: 2006-03-10.

[54] P. Szor, *"The Art of Computer Virus Research and Defence"*, Addison-Wesley, Upper Saddle River NJ, 2005.

[55] Technology Review, "The Pure Software Act of 2006", http://www.simson.net/clips/2004/2004.TR.04.PureSoftware.pdf, Last checked: 2006-10-16.

[56] M. Warkentin, X. Luo, and G. F. Templeton, "A Framework for Spyware Assessment", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[57] S.D. Warren, L.D. Brandeis, "The Right to Privacy", in *Harvard Law Review*, Volume 4, Issue 5, 1890.

[58] Webroot Software, "State of Spyware - Q1 2006", http://www.webroot.com/pdf/2005-q2-sos.pdf, Last checked: 2006-12-05.

[59] Webroot Software, "Differences between Spyware and Viruses", http://research.spysweeper.com/differences.html, Last checked: 2006-12-05.

[60] A. Weiss, "Spyware Be Gone", in the *ACM netWorker*, Volume 9, Issue 1, 2005.

[61] A. Westin, "*Privacy and Freedom"*, Atheneum, New York NY, 1968.

[62] G. Zacharia, A. Moukas, P. Maes, "Collaborative Reputation Mechanisms in Electronic Marketplaces", in *Proceedings fo the 32nd Hawai'i International Conference on System Sciencces*, Wailea Maui Hawaii, 1999.

[63] X. Zhang, "What Do Consumers Really Know About Spyware?", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

CHAPTER

4

Privacy-Invasive Software in File-Sharing Tools

18th IFIP World Computer Congress (WCC2004), 2004

Andreas Jacobsson, Martin Boldt and Bengt Carlsson

Personal privacy is affected by the occurrence of adware and spyware in peer-to-peer tools. In an experiment, we investigated five file-sharing tools and found that they all contained ad-/spyware programs, and, that these hidden components communicated with several servers on the Internet. Although there was no exchange of files by way of the file-sharing tools, they generated a significant amount of network traffic. Amongst the retrieved ad-/spyware programs that communicated with the Internet, we discovered that privacy-invasive information such as, e.g., user data and Internet browsing history was transmitted. In conclusion, ad-/spyware activity in file-sharing tools creates serious problems not only to user privacy and security, but also to network and system performance. The increasing presence of hidden and bundled ad-/spyware programs in combination with the absence of proper anti-ad/spyware tools are therefore not beneficial for the development of a secure and stable use of the Internet.

4.1 Introduction

As the Internet becomes more and more indispensable to our society, the issue of personal information is recognised as decisively important when building a secure and efficient social system on the Internet [3, 19]. Also, in an increasingly networked world, where new technologies and infrastructures, from pervasive computing to mobile Internet, are being rapidly introduced into the daily lives of ordinary users, complexity is rising [15]. As a consequence, vulnerabilities in systems are more eminent and greater in number than ever before. At the same time, the business climate on the Internet is tightening; e-commerce companies are struggling against business intelligence techniques, social engineering and frauds. A powerful component in any business strategy is user/customer information. In general, the company with the most information about its customers and potential customers is usually the most successful one [13, 19]. With respect to personal customer information, consumers generally want their privacy to be protected, but businesses, on the other hand, need reliable personal information in order to reach consumers with offers [13]. Undoubtedly, these demands must be satisfied to establish sound e-commerce, and a secure and well-functioning use of the Internet. However, these conflicting goals leave the control of user information at great risk, and a consequence may be that the users feel uneasy about sharing any personal information with commercial web sites. Human activity on the Internet will only thrive if the privacy rights of individuals are balanced with the benefits associated with the flow of personal information [13].

The problem of assuring user privacy and security in a computerised setting is not new, it has been a discussion for more than 30 years now [9]. However, there are some new aspects, that need to be highlighted. In this paper, we intend to explore privacy aspects concerning software components that are bundled and installed with file-sharing tools. Since file-sharing tools are used exclusively when connected to the Internet, users constitute a good foundation for online marketing companies to display customised ads and offers for users. The displayed contents of these offers are sometimes based on the retrieval of users' personal information. Usually, this kind of software operation is considered to be an invasion of personal privacy [8]. One of the most simple and clear definitions of privacy was first proposed in 1890 by Warren and Brandeis in their article "The Right to Privacy" [23], where privacy was defined as

"*the right to be let alone*". In general, privacy is the right of individuals to control the collection and use of information about themselves [3]. In an Internet setting, the extraction of the definition by Warren and Brandeis has come to mean that users should be able to decide for themselves, when, how, and to what extent information about them is communicated to others [7]. Previous work has suggested that malicious software, or malware, set to collect and transmit user information and/or to display ads and commercial offers without the consent of users have been found bundled with file-sharing tools [11, 22]. There are two kinds of software programs that perform such actions: adware displays advertisements, and spyware goes further and tracks and reports on users' web browsing, keystrokes or anything else that the author of the software has some interest in knowing. In reality, this means that software can be adware and spyware at the same time. However, not all adware is spyware and most spyware is not easily detected by displaying ads [11].

Ad-/spyware has gained a lot of space and attention lately. According to the Emerging Internet Threats Survey 2003 [6], one in three companies have already detected spyware on their systems, while 60% consider spyware to be a growing and future threat. Also, 70% of the companies say that peer-to-peer (P2P) file-sharing is creating an open door into their organisation. When it comes to adware, the Emerging Internet Threats Survey, states that adware and the use of file-sharing tools in office hours are devious and offensive threats that frequently evade both firewalls and anti-virus defences [6]. In effect, ad-/spyware creates problems, not only to user privacy, but also to corporate IT-systems and networks.

In this paper, we investigate what kind of privacy-invasive software that come bundled with five popular file-sharing tools. We also look into the Internet traffic that is being generated by these hidden programs. A discussion concerning the occurrence of ad-/spyware and its effects on privacy and security is undertaken. In the end, we present conclusions and findings.

4.2 Privacy-Invasive Programs and their Implications

One of the major carriers of ad-/spyware programs are P2P file-sharing tools [16, 22]. P2P refers to a technology which enables two

or more peers to collaborate in a network of equals [12, 18]. This may be done by using information and communication systems that are not depending on central coordination. Usually, P2P applications include file sharing, grid computing, web services, groupware, and instant messaging [12, 18]. In reality, there is little doubt that P2P networks furnish in spreading ad-/spyware [16]. Besides legal difficulties in controlling the content of P2P networks, another contributing factor is that the user is forced to accept a license agreement in order to use the software, but the contract terms are often formulated in such a way that they are hard for the user to interpret and understand. The effect is that most users do not really know what they have agreed to, and thus really cannot argue their right to privacy.

The occurrence of ad-/spyware programs in file-sharing tools pose a real and growing threat to Internet usage in many aspects, and to other interested parties than only to end users. Some examples argued on this topic are [6, 16, 22]:

- **Consumption of computing capacity:** Ad-/spyware is often designed to be secretly loaded at system start-up, and to run partly hidden in the background. Due to that it is not unusual for users to have many different instances of ad-/spyware running covertly simultaneously, the cumulative effect on the system's processing capacity can be dramatic. Another threat is the occurrence of distributed computing clients, bundled with file-sharing tools, that can sell the users' hard drive space, CPU cycles, and bandwidth to third parties.
- **Consumption of bandwidth:** Just as the cumulative effect of ad-/spyware running in the background can have serious consequences on system performance, the continual data traffic with gathering of new pop-ups and banner ads, and delivery of user information can have an imperative and costly effect on corporate bandwidth.
- **Legal liabilities:** With the new directives[1] concerning the use of file-sharing tools in companies, it is the company rather than a single user who is legally liable for, for instance, the breach of copyright (e.g., if employees share music files with other peers)

1. Examples on legal directives are the "Directive on Privacy and Electronic Communications" [5] of the European Union, and the "Spyware Control and Privacy Protection Act" [2] of the Senate of California, U.S.

and the spreading of sensitive information (e.g., if spyware programs transmit corporate intelligence).

- **Security issues:** Ad-/spyware covertly transmits user information back to the advertisement server, implying that since this is done in a covert manner, there is no way to be certain of exactly what information is being transmitted. Even though adware, in its purest form, is a threat to privacy rather than security, some adware applications have begun to act like Trojan horses allowing installation of further software, which may include malware. Security experts use the term Trojan horse for software that carries programs, which mask some hidden malicious functionality, but many web users and privacy experts use it to describe any program that piggybacks another. It is claimed that most of the latter are P2P file-sharing software that emerged as ad-supported alternatives in the wake of Napster's decline. In effect, if a computer has been breached by a Trojan horse, it typically cannot be trusted. Also, there is a type of spyware that has nothing to do with adware, the purpose here is to spy on the user and transmit keystrokes, passwords, card numbers, e-mail addresses or anything else of value to the software owner/author. In reflect, most security experts would agree that the existence of ad-/spyware is incompatible with the concept of a secure system.
- **Privacy issues:** The fact that ad-/spyware operates with gathering and transmitting user information secretly in the background, and/or displays ads and commercial offers that the user did not by him-/herself chose to view, makes it highly privacy-invasive.

Most ad-/spyware applications are typically bundled as hidden components of freeware or shareware programs that can be downloaded from the Internet [22]. Usually, ad-/spyware programs run secretly in the background of the users' computers. The reason for this concealing of processes is commonly argued as that it would hardly be acceptable if, e.g., free file-sharing software kept stopping to ask the user if he or she was ready to fetch a new banner or a pop-up window. Therefore, the client/server routine of ad-/spyware is executed in the background. In practice, there would be nothing wrong with ad-/spyware running in the background provided that the users know that it is happening, what data is being transmitted, and that they have agreed to the process as part of the conditions for obtaining the freeware. However, most users are unaware of that they have software on their computers that tracks

and reports on their Internet usage. Even though this may be included in license agreements, users generally have difficulties to understand them [22].

Adware is a category of software that displays commercial messages supported by advertising revenues [20]. The idea is that if a software developer can get revenue from advertisers, the owner can afford to make the software available for free. The developer is paid, and the user gets free, quality software. Usually, the developer provides two versions of the software, one for which the user has to pay a fee in order to receive, and one version that is freeware supported by advertising. In effect, the user can choose between the free software with the slight inconvenience of either pop-up ads or banners, or to pay for software free of advertising. So, users pay to use the software either with their money or with their time. This was the case until marketers noted three separate trends that pushed the development of adware into a different direction. Standard banner ads on the Internet were not delivering as well as expected (1% click-through was considered good) [22]. Targeted Internet advertising performed much better [21]. While office hours were dead-time for traditional advertising (radio, TV, etc.), many analyses showed a surprisingly high degree of personal Internet usage during office hours [21].

The conclusion was that targeted Internet advertising was a whole new opportunity for the marketing of products and services. All that was required was a method for monitoring users' behaviour. Once the adware was monitoring users' Internet usage and sending user details back to the advertiser, banners more suited to the users' preferences and personality were sent to the users in return. The addition of monitoring functionality turned adware into ad-/spyware, and the means to target advertising to interested parties accelerated. In reality, the data collected by ad-/spyware is often sent back to the marketing company, resulting in display of specific advertisements, pop-up ads, and installing toolbars showed when users visit specific web sites.

Spyware is usually designed with the same commercial intent as adware [20]. However, while most adware displays advertisements and commercial offers, spyware is designed with the intent to collect and transmit information about users. The general method is to distribute the users' Internet browsing history [22]. The idea behind this is that if you know what sites someone visits, you begin to get an idea of what that person wants, and may be persuaded to buy

[21]. Given the fact that more than 350 million users have downloaded KaZaa and supposedly also installed it on their computers [4], this enables for customised and personalised marketing campaigns to millions and millions of end users. Moreover, information-gathering processes have been implicated in the rising occurrence of unsolicited commercial e-mail messages (so called spam) on the Internet [6].

Besides the monitoring of Internet usage, there is an even greater danger, namely when spyware is set to collect additional and more sensitive personal information such as passwords, account details, private documents, e-mail addresses, credit card numbers, etc.

4.3 Experiment Design

4.3.1 Problem Domain

Programs designed with the purpose of locating and defeating ad-/spyware components are available throughout the Internet. Even so, these programs are not very refined. For instance, there is usually no linking between the identified ad-/spyware processes inside the computers and the corresponding servers outside, on the Internet. Also, there is no anti-ad-/spyware program that analyses what data content is being transmitted to other third parties on the Internet. So, even when using existing software, it is difficult do keep track of what is going on inside the computer, and what nodes outside it that obtain user-oriented information. As a consequence, Internet browsing records and/or credit card numbers could easily be distributed without the user's consent or knowledge.

In this light, the overall research problem for this paper was to explore the nature and occurrence of privacy-invasive software included in file-sharing tools used over P2P networks. On an experiment level, the research problem was divided into the following subquestions:

- What ad-/spyware programs can be found in file-sharing tools?
- What is the content and format of network data generated as a result of ad-/spyware programs involved in Internet communication?
- What is the extent of network traffic generated by such programs?

Even though there may be numerous components bundled with the installation of file-sharing tools, it is primarily the programs engaged in Internet communication that are of interest to us. There are two reasons for this. First, without this delimitation, the experiment data would be too comprehensive to grasp. Second, for ad-/spyware programs to leak personal information, they must be involved in communication over the Internet. This is of course particularly interesting from a privacy perspective.

Throughout this paper, we use the word ad-/spyware as a synonym for both adware and spyware. In general, both adware and spyware are namely considered to be privacy-invasive software. Also, since they typically are closely intervened with each other, and more or less perform similar actions it is problematic to separate adware from spyware [22].

4.3.2 Instrumentation and Execution

The experiment sample consists of the five most downloaded file-sharing tools [4]. The tools are, in order, the standard, freeware versions of KaZaa, iMesh, Morpheus, LimeWire and BearShare. Also, to be sure that the experiment results were derived from the installed file-sharing tools, we set up a reference computer, which was identical to the other work stations, i.e., the same configuration, but with no file-sharing tool installed. The experiment was executed in January 2004 as one consecutive session that lasted three days. This time range was chosen, because we wanted to avoid getting excessive data quantities, but at the same time be able to capture reliable results.

The experiment was carried out in a lab environment on PC work stations equally connected to the Internet through a NAT gateway. We used OpenBSD's packet filter to deny any inbound network requests, which allowed us to protect the work stations from external threats. The packet filter also helped in reducing the network traffic and in doing so, resulting in less data to analyse. By not downloading or sharing any content in the file-sharing tools we further reduced the amount of network data generated. All incoming and outgoing network traffic of the local computer's network interface were dumped into a file using Winpcap.

Hardware were equivalent for all work stations, which also contained byte identical installations of both the operating system

Microsoft Windows 2000 and program applications[2]. In order to reflect work stations in use, they were all set to browse the Internet according to a predefined schedule containing the 100 most visited web sites in the world [1]. This was done through an automatic surf program. Also, ten identical searches (e.g., "lord of the ring", "star wars", and "britney") were carried out in each of the file-sharing tools, but no files were downloaded. In the end of the experiment, several anti-ad-/spyware programs[3] were used to locate any known ad-/spyware programs previously installed.

Binding network communication to programs is a key feature in the experiment. For allowing continuous monitoring and logging of processes and their use of sockets, we developed a program in C++, which was based on Openport. We chose not to use any Win32 firewalls claiming to support outbound filtering on application level for two reasons. First, they fail in allowing real outbound filtering per application, and there are a number of programs capable of penetrating these fake protections [14, 17]. Second, we have no detailed knowledge in the internal workings of such firewalls and therefore cannot foresee what to expect from them. Finally, it should be emphasised that there exist ways for a malicious program to send network data undetected by the monitoring application, due to the architecture of Windows.

4.3.3 Data Analysis

After having performed the experiment, we compiled the data results and set to identify all programs that were bundled with each file-sharing tool. This data was provided by our own process-to-network mapping program in cooperation with the selected anti-ad-/spyware programs. We then isolated the operating system related programs found on the reference work station, since they were established as harmless. Next, we reduced all benign programs handling file-exchange tasks. Remaining were a set of programs that were not related to either the operating system or file-exchange tasks. Further, by using the results from the anti-ad-/spyware tools, we divided the set of programs into two subsets, namely known ad-/spyware programs and unknown programs. The nature of these unknown programs was analysed based on their corresponding net-

2. These configuration properties were enabled through a self-developed disc cloning system based on standard FreeBSD components.
3. For a detailed list of the programs used, see Appendix of this thesis.

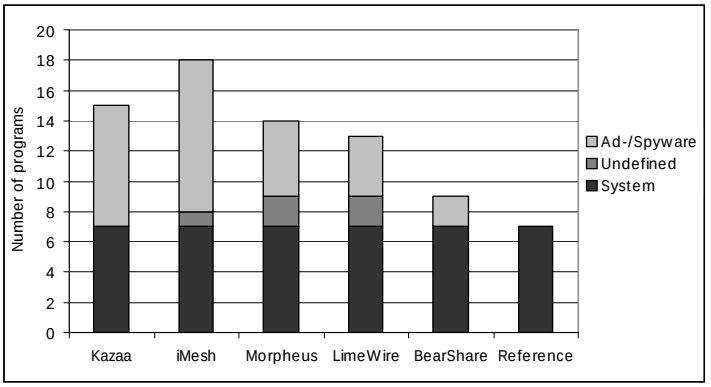

Figure 4.1 Amount of programs in the experiment sample

work traffic. Also, in some cases we needed additional information and thus turned to Internet resources. Based on this analysis, the remaining ad-/spyware programs were located. In the final step, we divided the retrieved set of ad-/spyware programs into two subsets, namely those involved in Internet communication and those that were not. This analysis was founded on the data from our process-to-network mapping program. In effect, the results from the program analysis lead to a classification of programs as either ad-/spyware programs, system programs or unknown programs.

All data analysis was done in a Unix environment. The data was analysed and filtered using standard Unix programs such as sed, awk, sort, uniq and grep. Much of the analysis was automated using shell scripts and where this could not be done small programs in C were created. To analyse and filter network data, the program Ethereal was used.

In addition, we wanted to see if the corresponding servers were known ad-/spyware servers. Therefore, an effort to map the server names that were involved in Internet communication with a blacklist specifying known ad-/spyware servers [10] was also undertaken.

4.4 Experiment Results and Analysis

4.4.1 Ad-/Spyware Programs in File-Sharing Tools

According to the results, several programs were located for each file-sharing tool (see Figure 4.1). Of these programs, we identified 10 ad-/spyware programs for iMesh, and eight for KaZaa. Interestingly, these two file-sharing tools were among the two most popular ones [4]. The rates for the other file-sharing tools were five for Morpheus, four for LimeWire and two for BearShare. Also, iMesh, Morpheus and LimeWire contained programs that we were unable to define. However, these programs were all involved in Internet communication.

Name	Host	Adware	Spyware	Download	Internet
BroadcastPC	M	x	x	x	**X**
KeenValue	K	x	x	**X**	**X**
Morpehus	M	**X**	x	**X**	**X**
BargainBuddy	I, K	x	x	x	
TopMoxie	L, M	x	x	x	
Cydoor	I, K	x	x		**X**
Gator	I, K	**X**	x		**X**
SaveNow	B	**X**	**X**		**X**
BonziBuddy	L	x	x		
Web3000	I	x	x		
ShopAtHomeSelect	I		**X**	**X**	**X**
WebHancer	K		x	x	
BrilliantDigital	K	x		**X**	**X**
MoneyMaker	L, M	**X**		**X**	**X**
Claria	I, K	x			**X**
iMesh	I	x			**X**
WeatherCast	B	x			**X**
CasinoOnNet	L	x			
MyBar	I, K, M	x			
New.Net	I			**X**	**X**
FavoriteMan	I			x	

Table 4.1 Identified ad-/spyware programs.

We discovered that all of the file-sharing tools contained ad-/spyware programs that communicated with the Internet. KaZaa and

iMesh included a relatively high amount of such programs. Even so, the anti-ad-/spyware tools defined several other ad-/spyware programs also installed on the computers. Although this was the case, these programs did not communicate with servers on the Internet during the experiment session.

In Table 4.1, a detailed list of the retrieved ad-/spyware components can be found. As can be seen, the ad-/spyware components were divided into "Adware" respectively "Spyware" based on their actions. Also, we included a category entitled "Download" because some of the ad-/spyware programs included functionality that allowed further software and/or updates to be downloaded and installed on the computers. In addition, programs involved in Internet communication are specified in the category called "Internet". In the column entitled "Host", the five file-sharing tools utilised as carriers of ad-/spyware are listed[4]. In the cases where the empirical results could confirm the recognised view shared by anti-ad-/spyware tools and Internet resources, the x-markers in the table are declared with bolded capital letters.

One reason to why we could not confirm that every ad-/spyware program was involved in Internet communication was that so called Browser Helper Objects (BHO) were installed in Internet Explorer. Malicious BHOs infiltrate the web browser with the intent to access all data generated by Internet Explorer in order to spy on the user and transmit user behaviour to third parties [20]. Such BHOs typically gain the same privileges as its host (i.e., Internet Explorer), which endorse them to penetrate personal firewalls. This means that any possible ad-/spyware traffic distributed via BHOs is highly problematic to detect since it may very well be ordinary browser traffic. In Table 4.1, we also included two programs, New.Net and FavoriteMan, even though they were not classified as neither adware nor spyware. However, they allowed for installation of further software, which may be malicious.

4.4.2 The Extent of Network Traffic

The results showed that a significant amount of network traffic was generated, although there was no exchange of files between the file-sharing tools and other peers on the Internet (see Figure 4.2). In

4. In the category entitled "Host", K is for KaZaa, I for iMesh, M for Morpheus, L for LimeWire and B is for BearShare.

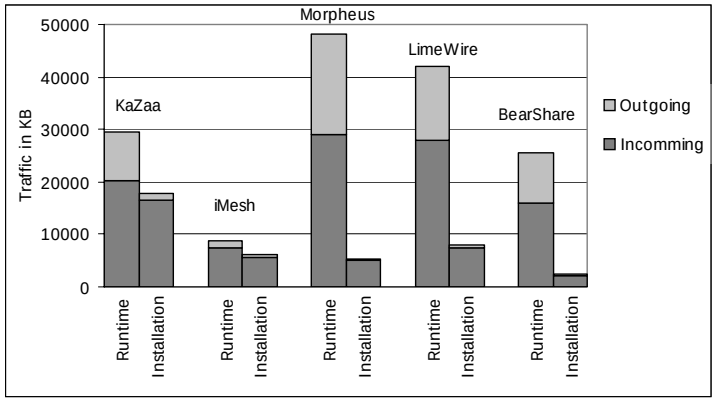

Figure 4.2 Network data traffic.

that light, the amount of network traffic generated in this experiment can be seen as a minimum rate to be expected when running file-sharing tools. Notably, installing Morpheus and LimeWire resulted in a relatively high traffic quote, both when it came to incoming as well as outgoing traffic. On the contrary, iMesh, who also had the largest quantity of bundled programs, represented the least amount of network traffic.

In Figure 4.2, we included compilations of network traffic for both the installation process and the runtime part per file-sharing tool. In the cases of Morpheus, LimeWire and BearShare, a considerable amount of network activity was generated after the installation. For KaZaa, a significant quantity of network traffic was caused during the installation. In comparison, iMesh produced a notably limited size of network traffic, both during and after installation.

Furthermore, the results suggested a diversity in Internet communication. This is shown in that programs in the file-sharing tools communicated with several different servers on the Internet. Although Morpheus did not contain a particularly great number of bundled programs, it generated notably much network traffic. In reflection, Morpheus communicated with the largest amount of Internet servers, whereas the rates for the other file-sharing tools were in a relatively low accordance with each other. In addition, the results substantiated that most of the invoked servers had domain names. Overall, each of the file-sharing tools contained programs that communicated with known ad-/spyware servers from the specified blacklist [10].

4.4.3 The Contents of Network Traffic

The outgoing network data was overall problematic to analyse and understand. In most cases the data was not readable, meaning that it was either encrypted or in a format not graspable. This is also an explanation to why we could confirm only two spyware programs (see Table 4.1). Although most traffic data was not in clear text, we were able to extract and interpret some of the contents. We discovered that sensitive data such as information about the user (e.g., user name), geographical details (e.g., zip code, region and country) and Internet browsing history records were sent from identified ad-/spyware components to several servers on the Internet. Also, there were other types of information that were transmitted, for example, machine ID, details about program versions, operating system, etc.

According to the results, one spyware program (ShopAtHomeSelect) was found in the iMesh file-sharing tool. In the experiment, that program transmitted traffic measurement reports and Internet browsing history records to invoked servers on the Internet. Also, in BearShare, one spyware program (SaveNow) transmitted data such as Internet history scores and user-specific information.

The experiment results also reveal one of the methods for ad-/spyware programs to transmit user and/or work station data. In the BearShare tool, the information that was fed into the file-sharing software by the user was re-distributed within the tool to one or numerous ad-/spyware programs (SaveNow and WeatherCast) that transmitted the information to servers called upon. This method makes it difficult to map various program components to the actual file-sharing activity. Also, it undermines the ability to control what software objects are useful and legitimate in relation to the redundant or privacy-invasive programs that clog down the computers, systems and networks.

The analysis of the contents of the incoming network traffic was more problematic to conduct than in the case of outgoing traffic. Foremost, because the data quantity was both comprehensive and widespread. Since our focus was on privacy-invasive software, the outgoing traffic content was the most interesting so the efforts were mainly put into that. This, in combination, with vast quantities of incoming network data made it difficult to confirm adware recognised by the anti-ad-/spyware tools and Internet resources. Also,

the same discussion concerning the occurrence of BHOs would apply for the unconfirmed adware. However, in the retrieved incoming data, a few interesting results were found.

The retrieved adware programs performed activities such as displaying commercial ads, causing browser banners and pop-ups. In particular, Morpheus and LimeWire proved to contain adware programs that generated much incoming data traffic. In LimeWire, results showed that lists of Internet sites and new programs were retrieved from the Internet by the adware MoneyMaker. In Morpheus, the P2P program itself downloaded and displayed ads and banners.

4.5 Discussion

With the occurrence of ad-/spyware technology in file-sharing tools, the monitoring of Internet usage has become a common feature. Today, most ad-/spyware programs gather and transmit data such as Internet browsing history records to third parties. That type of information can be correlated to a user and thus employed for marketing purposes.

The experiment has shown that all of the investigated file-sharing tools contained ad-/spyware programs. The ad-/spyware programs that operated inside the computers had an open connection to several Internet servers during the entire experimental session. We know that content-sensitive information was sent, but we may only guess the full extent of information harvesting, because most packets were not sent in clear text. Even though we saw no example of highly sensitive personal information, such as passwords and keystrokes, were transmitted by the ad/spyware programs in the experiment, we cannot be sure that these activities were not happening. Spyware may collect and transmit genuinely sensitive information about users such as, e.g., account details, private documents, e-mail addresses, and credit card numbers. The information is secretly sent back to numerous servers owned by companies that make a profit on these activities. Although it is problematic to elaborate on the business ethics of these companies, the occurrence of ad-/spyware programs are reasons enough to question this behaviour. In addition, ad-/spyware programs are responsible for all kinds of unwanted actions. Besides invasion of privacy, they can make the system unstable, degrade system performance, create scores of cop-

ies of itself to make removal difficult, and act as security holes in the system.

The actions performed by ad-/spyware programs are approaching the operations of a virus. Since users install them on voluntary basis, the distribution part is taken care of by the file-sharing tools. This makes ad-/spyware programs function like a slowly moving virus without the distribution mechanisms usually otherwise included. The general method for a virus is to infect as many nodes as possible on the network in the shortest amount of time, so it can cause as much damage as conceivable before it gets caught by the anti-virus companies. Ad-/spyware, on the other hand, may operate in the background in such a relatively low speed that it is difficult to detect. Therefore, the consequences may be just as dire as with a regular virus. In addition, the purpose of ad-/spyware may not be to destroy or delete data on the work stations, but to gather and transmit veritably sensitive user information. An additional complicating factor is that anti-virus software companies do not usually define ad-/spyware as virus, since it is not designed to cause destruction. Overall, the nature of ad-/spyware substantiates the notion that malicious actions launched on computers and networks get more and more available, diversified and intelligent, rendering in that security is extensively problematic to uphold.

Ad-/spyware enables for the spreading of e-mail addresses that may result in the receiving of spam. Due to the construction of ad-/spyware, it may collect information that concerns other parties than only the work station user. For example, information such as telephone numbers and e-mail addresses to business contacts and friends stored on the desktop can be gathered and distributed by ad-/spyware. In the context that ad-/spyware usually is designed with the purpose of conveying commercial information to as many users as possible, not only the local user may be exposed to negative consequences of ad-/spyware. In other words, the business contacts and friends may be the subjects of ad-/spyware effects such as, e.g., receiving unsolicited commercial e-mail messages. This means that even though my computer may be secure, a breached computer owned by a network neighbour can cause me harm. So, the security of a neighbour very much becomes my own concern.

Besides security issues, ad-/spyware creates intrusion to privacy. An inconvenience commonly argued is that ad-/spyware programs display commercial messages based on the retrieval of personal information fetched without the explicit consent of the users. Even

though the offers of these advertising campaigns may be in the interest of some users, there is a fine line between what users in general regard as useful information and what is an intrusion to personal privacy. One thought is that the more personalised the offers get, the more likely users are to regard them as privacy invaders. If so, what happens when users are presented with advertisements in such an extent that they hardly are able to distinguish the possibly serious offers from all the offers. If users ignore marketing messages, there is evidently a great risk for the success of consumer-based e-commerce.

A second privacy concern is the spreading of content that the ad-/spyware distributor did not intend for. One example of this would be a malicious actor that gained control of ad-/spyware servers, and broadcasted offensive unsolicited messages (e.g., adult material, political messages and/or smearing campaigns, etc.) to a great number of users. Although users may consider regular commercial ads to be harmless, most people react negatively upon frequently receiving repulsive pictures and texts. This suffices for that the ad-/spyware providers need to take their own security with great seriousness. If they lose control of their servers, the damage may be devastating. This could be even more devastating if the ad-/spyware program updates on the company servers were replaced with malicious software. In effect, real and destructive malware (e.g., viruses, Trojans, and worms) could be spread to vast groups of ad-/spyware hosts.

4.6 Conclusions

The experiment has shown that all of the investigated file-sharing tools contained ad-/spyware programs. The ad-/spyware programs operating inside the computers had an open connection where the information was secretly sent back to numerous servers owned by companies that make a profit on these activities. Measurements suggested that the carriers of ad-/spyware, file-sharing tools, generated a significant amount of network traffic, even when not exchanging files. The presence of ad-/spyware programs and the network traffic that they generate contribute in over-consumption of system and network capacity.

Ad-/spyware is acting like a slowly moving virus, installed on a voluntary basis, with hidden properties problematic to detect and

remove. The payload of ad-/spyware may not be to destroy or delete data on the work stations, but to gather and transmit veritably sensitive user information. The distribution part is taken care of by the file-sharing tools with an additional complicating factor; antivirus software companies do not usually define ad-/spyware as virus, since it is not designed to cause destruction.

The nature of ad-/spyware may lead to that not only host users are affected. Ad-/spyware may gather and distribute the details of business contacts and friends resulting in negative consequences to other parties than the infected desktop owner. This means that even though my computer may be secure, a breached computer owned by a network neighbour can cause me harm. So, the security of a neighbour very much becomes my own concern.

Furthermore, the occurrence of ad-/spyware can render in that privacy-invasive messages may be distributed and displayed to large amounts of users. Exposure to messages not chosen by the user, or collection and transmission of user information are two key privacy concerns. In this way, users' right to control what, how and when information about themselves is communicated to other parties is almost non-existing. In conclusion, the nature of ad-/spyware programs ignore users' right to be let alone. The increasing presence of hidden and bundled ad-/spyware programs in combination with the absence of proper anti-ad/spyware tools are therefore not beneficial for the development of a secure and stable use of the Internet.

4.7 References

[1] Alexa Web Search., http://www.alexa.com/site/ds/topsites?ts mode=global&lang=none, 2004-04-27.

[2] California Senate Assembly Bill 1386, United States of America, 2003., http://info.sen.ca.gov/pub/bill/asm/ab1351-1400/ab1386bill20030904 chaptered.html, 2004-04-27.

[3] M. Caloyannides, "Privacy vs. Information Technology", in *IEEE Security & Privacy*, Vol. 1, No. 1, pp. 100-103, 2003.

[4] C|Net Download.com., http://www.download.com/, 2004-04-27.

[5] "Directive on Privacy and Electronic Communications", Directive 2002/58/EC of the European Parliament and of the Council of 12

July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector, 2002.

[6] "Emerging Internet Threats Survey 2003", commissioned by Websense International, Ltd., February, 2003., http://www.websense.com/company/news/research/EmergingThreats2003EMEA-de.pdf, 2004-04-27.

[7] S. Fischer-Hübner, "Privacy in the Global Information Society", in *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, Lecture Notes in Computer Science LNCS 1958, Springer-Verlag, Berlin Germany, 2000.

[8] S. Garfinkel, "*Database Nation: The Death of Privacy in the 21st Century*", O'Reilly & Associates, Inc., Sebastopol CA, 2001.

[9] E. Grenier, "Computers and Privacy: A Proposal for Self-Regulation", in *Proceedings of the First ACM Symposium on Problems in the Optimization of Data Communications Systems*, ACM Press, New York NY, 1969.

[10] Gorilla Design Studio: The Hosts Files., http://www.accs-net.com/hosts/, 2004-04-27.

[11] M. McCardle, "How Spyware Fits into Defence in Depth", SANS Reading Room, SANS Institute, 2003., http://www.sans.org/rr/papers/index.php?id=905, 2004-04-27.

[12] A. Oram, "*Peer-To-Peer: Harnessing the benefits of a Disruptive Technology*", O'Reilly & Associates, Inc., Sebastopol CA, 2001.

[13] T. Otsuka, and A. Onozawa, "Personal Information Market: Toward a Secure and Efficient Trade of Privacy", in *Proceedings of the First International Conference on Human Society and the Internet*, Lecture Notes in Computer Science LNCS 2105, Springer-Verlag, Berlin Germany, 2001.

[14] Outbound., http://www.hackbusters.net/ob.html, 2004-04-27.

[15] L. Palen, and P. Dourish, "Unpacking Privacy for a Networked World", in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ACM Press, New York NY, 2003.

[16] B. Robertsson, "Five Major Categories of Spyware", in Consumer WebWatch, October 21, USA, 2002., http://www.consumerwebwatch.org/news/articles/spyware categories.htm, 2004-04-27.

[17] Robin Keir's FireHole., http://keir.net/rehole.html, 2004-04-27.

[18] D. Schoder, and K. Fischbach, "Peer-to-Peer (P2P) Computing", in *Proceedings of the 36th IEEE Hawaii International Conference on System Sciences*, IEEE Computer Society Press, Los Alamitos CA, 2003.

[19] C. Shapiro, and H. Varian, "*Information Rules: A Strategic Guide to the Networked Economy*", Harvard Business School Press, Boston MA, 1999.

[20] E. Skoudis, "*Malware - Fighting Malicious Code*", Prentice Hall PTR, Upper Saddle River NJ, 2004.

[21] 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.

[22] K. Townsend, "Spyware, Adware, and Peer-to-Peer Networks: The Hidden Threat to Corporate Security" (technical white paper), Pest-Patrol, 2003., http://www.pestpatrol.com/Whitepapers/CorporateSecurity 0403.asp, 2004-04-27.

[23] S.D. Warren, and L.D. Brandeis, "The Right to Privacy", in *Harvard Law Review*, No. 5, pp. 193-220, 1890-91.

CHAPTER

5

Exploring Spyware Effects

9th Nordic Workshop on Secure IT Systems (NordSec04), 2004

Martin Boldt, Andreas Jacobsson and Bengt Carlsson

In this paper, we discuss various types of spyware programs, their behaviour, how they typically infect computers, and the propagation of new varieties of spyware programs. In two experiments, we investigate the occurrence and impact of spyware programs found in popular P2P applications. Based on the findings from the empirical investigations, we try to lift the perspective to a more general view on spyware deriving from the theory of (virtual) network effects. In a model, we categorize in what ways spyware might decrease the utility of belonging to a large virtual network. Here, the baseline is that spyware programs intrude systems and networks, but since they profit from user data they also intrude user privacy. In the model, the intrusions are classified as moderate, severe or disastrous. We found that spyware has the potential to overthrow the positive aspects of belonging to a large network, and network owners should therefore be very careful about permitting such programs in applications and on networks.

5.1 Introduction

During recent years, the world has seen the introduction of peer-to-peer (P2P) systems. P2P technology provides several beneficial solutions like, e.g., file-sharing, grid computing, web services, groupware and instant messaging (IM) [7]. P2P refers to a technology which enables two peers or more to collaborate in a network of equals [7 , 10]. This may be done by using information and communication systems that are not depending on central coordination. P2P technology was first widely deployed and popularized by file-sharing applications such as KaZaa and IM tools like ICQ.

Even though there are several benefits with belonging to a large virtual network such as a P2P file-sharing network, the rising occurrence of malicious software (malware) may seriously impact the positive utility of using P2P applications. Usually, only the positive effects that increase utility are emphasized when discussing participation in large networks [5]. One example is the theory of virtual network[1] effects. Network effects are usually described as when the value of a product to one user depends on how many other users there are [11]. Often, utility of the system is proportional to the aggregate amount of resources that the participants are willing to put together. On information technologies, users generally benefit from utilising a popular format, system or application [11]. Typically, technologies subject to strong network effects tend to exhibit long lead times until a critical mass of users is obtained [5]. Then, explosive growth is followed. From the perspective of a network owner, a large network may help to create a strategic advantage useful for competition and growth purposes [1]. From the perspective of a network user, the larger the network is, the more valuable it will be to participants and users [1].

There are two kinds of feedback from network effects: positive and negative [11]. Positive feedback can be explained in that when a person joins a network, the network gets bigger and better, to everyone's benefit. However, large networks may also be exposed to negative feedback, which bring about significant risks and severe consequences for all of the network nodes. Therefore, negative feedback may decrease the utility of belonging to that network. To

1. A virtual network describes a network of users bound together by a certain standard or technology, and where the exchange of information is the foundation for any information transaction. One example is the Internet.

large networks, such as P2P file-sharing networks, there could be numerous examples of applications (e.g., malware), which contribute in creating negative effects that impact network utility. However, in this paper, we focus on one of these applications, namely spyware.

There are many different kinds of spyware, and hundreds of such programs exist throughout the Internet today [9]. Spyware programming is a relatively new computing phenomenon. Although there is no precise definition, the term "spyware" is typically used to refer to a category of software that, from a user's perspective, covertly gathers information about a computer's use and relays that information back to a third party. In this paper, we use the term spyware in conformity with this common usage. However, in 5.2, we look into and discuss some of the current views on the concept of spyware.

Even though most people are aware of spyware, it seems that the research community has spent limited effort on understanding the nature and extent of the spyware problem. However, so far there have been some initial research attempts (see for example [4 , 9 , 17]) of which this paper is an additional effort. On the other hand, most network practitioners and experts agree that spyware is a real problem with increasingly negative effects. One example of this view is derived from the Emerging Internet Threats Survey 2003 [3], which states that one in three companies have detected spyware on their systems, while 60% consider spyware to be a growing and future threat. Also, 70% of the companies consider that file-sharing over P2P networks is creating an open door into their organisation. Another example is an investigation made by Earthlink (one of the major American ISPs) [13]. Earthlink set to measure the occurrence of spyware on more than 2 million computers connected to their network. A total number of 12.1 million different spyware types were detected. Out of these, Trojan horses and system monitors approached 700 000 instances, and the remaining 11.4 million instances were classified as adware. Also, experts suggest that spyware infect up to 90% of all Internet-connected computers [13].

In summary, spyware is a problem that should be taken seriously, because it may have the potential to threaten the utility of belonging to a large virtual network. In this paper, we focus on exploring the effects of spyware programs that are bundled with several P2P applications. The aim is to investigate the implications on system capacity, network bandwidth, security and privacy. Besides introduc-

ing results from empirical investigations, we also discuss the network effects of spyware.

The paper is organised as follows. First, we give an introduction to spyware, in which we discuss the various kinds of spyware programs, their behaviour, how they typically infect computers, and the proliferation of new varieties of spyware. Next, we investigate the occurrence and impact of spyware programs found in popular P2P applications. In 5.4, we discuss the findings from the experiments and also try to lift the perspective to a more general view on spyware deriving from the theory of virtual network effects. In the end, conclusions are presented.

5.2 On Spyware

5.2.1 The Background of Spyware

As stated by [9], spyware exists because information has value. The idea with spyware is simply to fetch information. If a software developer can get revenue from advertisers, the owner can afford to make the software available for free. The developer is paid, and the user gets free, quality software. Usually, the developer provides two versions of the software, one for which the user has to pay a fee in order to receive, and one version that is freeware supported by advertising. In these cases, free software typically includes programs set to display advertisements and offers to the users (that is; adware). Therefore, the user can choose between the free software with the slight inconvenience of either pop-up ads or banners, or to pay for software free of advertising. So, users pay to use the software either with their money or with their time.

This method of including rather benign adware when developing and distributing free software was common until marketers noted three separate trends that pushed the development of adware into a different direction. The background was that:

- standard banner ads on the Internet were not delivering as well as expected (1% click-trough was considered good) [15],
- targeted Internet advertising typically performed much better [14], and

- while office hours were dead-time for traditional advertising (radio, TV, etc.), many analyses showed a surprisingly high degree of personal Internet usage during office hours [14].

The conclusion was that targeted Internet advertising was a whole new opportunity for the marketing of products and services. All that was required was a method for monitoring users' behaviour. So, once the adware was monitoring users' Internet usage and sending user details back to the advertiser, banners more suited to the users' preferences and personality was sent to the users in return. The addition of monitoring functionality turned adware into spyware, and the means to target advertising to interested parties accelerated [15]. In reality, the data collected by spyware is often sent back to the marketing company, resulting in display of specific advertisements, pop-up ads, and installing toolbars showed when users visit specific web sites. In this sense, spyware programs became technologies used to fetch valuable customer information.

5.2.2 The Operations of Spyware

The usual method for a spyware is to run secretly in the background of the users' computers [6]. The reason for this concealing of processes is commonly argued as that it would hardly be acceptable if, e.g., free file-sharing software kept stopping to ask the user if he or she was ready to fetch a new banner or a pop-up window [15]. Therefore, the client/server routine of spyware is normally executed in the background. In practice, there would be nothing wrong with spyware running in the background provided that the users know that it is happening, what data is being transmitted, and that they have agreed to the process as part of the conditions for obtaining the freeware. However, most users are unaware of that they have software on their computers that tracks and reports on their Internet usage. Typically, a spyware program covertly gathers user information and spreads it without the user's knowledge of it. Once installed, the spyware monitors, e.g., user activity on the Internet and transmits that information in the background to third parties, such as advertising companies. In reality, spyware run constantly, even when their carrier program, e.g., a file-sharing tool, has been terminated.

A more or less legal grey area is exploited by the spyware actors, since they in most program licenses specify that information may be gathered for corporate purposes. However, the usual model is to

collect more information than have been asked for [15]. Besides this, most license agreements are formulated in such a way that they are extensively hard for users to understand.

5.2.3 The Types of Spyware

There are many different kinds of spyware. For instance, one of the leading anti-spyware tools, PestPatrol, has a record of over 1400 instances of spyware published on their web site [8]. In order to make the spyware domain more graspable, we present the following classes of spyware. This classification is in conformity with a recently published study on measurement and analysis of spyware [9], although when presented here, the order of spyware types ranges from minimum to maximum user impact:

- Cookies and web bugs: Cookies are small pieces of state stored on individual clients' 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 behaviour of users across many Internet sites. Web bugs are usually described as invisible images embedded on Internet pages used for locating a connection between an end user and a specific web site. They are related to cookies in that advertisement networks often make contracts 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. Instead they rely on existing web browser functions.
- Adware: Adware is a more benign form of spybot (see below). Adware is a category of software that displays advertisements tuned to the user's current activity. Although most "genuine" adware programs only display commercial content, some hybrids are involved in reporting the aggregate or anonymised user behaviour to a third party, as described in 5.2.1.
- Tracks: A "track" is a generic name for information recorded by an operating system or application about actions that the user has performed. Examples of tracks include lists of recently visited web sites, web searches, web form input, lists of recently opened files, and programs maintained by operating systems. Although a track is typically not harmful on its own, tracks can be mined by malicious programs, and in the wrong context it can tell a great deal about a user.

- Browser hijackers: Hijackers attempt to change a user's Internet 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: install a browser extension (called a "browser helper object"), modify Windows registry entries, or directly manipulate and/or replace browser preference files. Browser hijackers are also known to replace content on web sites with such promoted by the spyware authors [12].
- Spybots: Spybots are the prototypes of spyware. A spybot monitors a user's behaviour, collects logs of activity and transmits them to third parties. Examples of collected information include fields typed in web forms, lists of e-mail 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 program launched whenever the host operating system boots.
- System monitors: System monitors record various actions on computer systems. This ability makes them powerful administration tools for compiling system diagnostics. However, if misused system monitors become serious threats to user privacy. Keyloggers are a group of system monitors commonly involved in spyware activities. Keyloggers were originally designed to record all keystrokes of users in order to find passwords, credit card numbers, and other sensitive information.
- Malware: Malware is a set of instructions that run on a computer and make the system do something that an attacker wants it to do [12]. Malware refers to a variety of malicious software that includes viruses, worms, and Trojan horses. Spyware is one form of malware, but as will be discussed later on, spyware may also include instructions for downloading and installing, e.g., a virus.

Spyware succeeds because some of today's desktop operating systems make spyware simple to build and install [9]. Many instances of spyware have the ability to self-update, or automatically download new versions of themselves to the local host. Self-updating allows spyware authors to introduce new functions over time, but it may also be used to evade anti-spyware tools by avoiding specific signatures contained within the tools' signature databases using polymorphic techniques.

5.2.4 On the Implications of Spyware

Spyware may occupy resources of the computer that it infects or alter the functions of existing applications on the affected computer to the benefit of a third party. In that sense, spyware poses several risks. One commonly argued is that spyware compromises a user's privacy by transmitting information about that user's behaviour [4]. Even so, a spyware can also detract from the usability and stability of the computing environment of the user [9]. In addition, a spyware has the ability to introduce new security vulnerabilities to the infected host by downloading software updates [6]. Due to that spyware is widespread, such vulnerabilities put numerous amounts of computers at risk.

To summarize, the occurrence of spyware programs raise a real and growing threat to Internet usage in many aspects, and to other interested parties than only to end users. Four categories frequently argued on this topic are [3 , 6 , 15]:

- Consumption of system capacity: Spyware is often designed to be secretly loaded at system startup, and to partly run hidden in the background. Due to that it is not unusual for users to have many different instances of spyware running covertly simultaneously, the cumulative effect on the system's processing capacity can be dramatic.
- Consumption of bandwidth: The continual data traffic with gathering of new pop-ups and banner ads, and delivery of user data can have an imperative and costly effect on both private and corporate bandwidth.
- Security issues: Spyware covertly transmits user information back to the advertisement server, implying that since this is done in a covert manner, there is no way to be certain of exactly what data is being transmitted. Even though spyware, in its purest form, is a threat to privacy rather than security, some spyware programs have begun to act like Trojan horses. Most security experts would agree that the existence of spyware is incompatible with the concept of a secure system.
- Privacy issues: The fact that spyware operates with gathering and transmitting user information secretly in the background, and/or displays ads and commercial offers that the user did not by him-/herself chose to view, makes it highly privacy-invasive. Also, spyware enables for the spreading of e-mail addresses that

may result in the receiving of unsolicited commercial e-mail (so called spam).

5.3 Experiments

We have developed a method for identifying and analysing spyware components and their behaviour on their host systems. This method has been used in several experiments (see, e.g., [4 , 17]). In this section, we present the method applied in two experiments. Thereafter, a compilation of the experiment results is given.

5.3.1 Method

The method is tightly coupled with our security laboratory. Mainly because our experiment method is based on state preservation of computer systems, which can be provided due to the computer architecture of the security laboratory[2]. By storing the initial baseline state of a system it is later possible to conclude what changes occurred with regards to this baseline. In practice, this means that we store the state of a base system before installing any application carrying spyware components. Afterwards, it is possible to conclude any changes between the two. By also capturing all network data sent and binding that traffic to the corresponding program, we can correlate network data to specific programs. It is also possible to include measurements of, e.g., CPU and network utilization during the experiments.

By using this method, all systems that are measured consist of identical hardware and network setups. Therefore, operating systems and their applications are bitwise identical for all subjects in the experiment sample. This suffices for the generation of reliable results. In order to be sure that the results are derived from a certain spyware, we included a "clean" reference computer in the experiment.

Since file-sharing tools are notoriously known for bundling spyware, we used such applications in both of the experiments. In this context, it should be pointed out that no file-sharing activity took

2. Throughout the experiments, we used 2.8Ghz Pentium 4 computers with 512MB primary memory.

place in terms of sharing or downloading any content on the P2P networks. Our examination was limited to software versions released between January and May 2004, and as such, our observations and results might not hold for other versions. Also, we used an Internet surfing program that automatically simulated a user visiting 100 preconfigured Internet sites. This was an attempt to trigger any spyware to either leak this information to third parties or to hijack the web sessions. In order to identify and locate the spyware programs, several anti-spyware tools were used[3].

5.3.1.1 Experiment 1

In the first experiment, we investigated the occurrence and operations of five popular file-sharing tools[4]. More specifically, we examined spyware programs that were bundled with the file-sharing tools, the content and format of network data caused by spyware involved in Internet communication, and the extent of network traffic generated by such programs. Even though there may be numerous components bundled with the installation of file-sharing tools, it was primarily the programs engaged in Internet communication that were of interest to us. There are two reasons for this. First, without this delimitation, the experiment data would be too comprehensive to grasp. Second, for spyware programs to leak user data, they must be involved in communication over the Internet.

5.3.1.2 Experiment 2

In the second experiment, we set to explore the effects in terms of resource usage that spyware bring about on a local system. A major problem introduced when setting up such an investigation involve how to choose the experiment sample. What we wanted was a program instance that was free of spyware and another instance (of the same program) that included spyware. Unfortunately it is almost impossible to remove only the spyware components and still have a working version of the original program since such components are very tightly coupled with the original program. We came to an acceptable solution by selecting KaZaa and KaZaa Lite K++ as the two subjects in the experiment sample. KaZaa Lite K++ is an instance of KaZaa where all spyware components have been

3. For a detailed list of the programs used, see http://www.ipd.bth.se/aja/SpywEffects_Ref.pdf
4. The file-sharing tools were the standard (free) versions of BearShare, iMesh, KaZaa, LimeWire, and Morpheus.

removed by an independent group that reverse-engineered the original KaZaa program, carefully excluding or disabling all bundled components not solely used for file-sharing purposes. By using these two KaZaa versions, it was possible to subtract the resource utilization of KaZaa Lite K++ from the utilization of the original KaZaa and thereby receive a measurement of resources used by the spyware programs.

5.3.2 Results and Analysis

5.3.2.3 Experiment 1

A detailed list of the identified spyware programs is presented in Table 5.1 After having analysed the captured data, we concluded that all file-sharing tools contained spyware.

The two main carriers of spyware were iMesh and KaZaa (they included ten respectively eight programs each). The rates for the remaining file-sharing tools were five for Morpheus, four for LimeWire, and two for BearShare. In addition to these findings, we also discovered that all file-sharing tools contained spyware that were involved in Internet communication.

As can be seen in Table 5.1, the retrieved spyware components were divided into "Adware" and "Spybot" based on their operations. We also included a category called "Download" because some of the components allowed for further software and/or updates to be downloaded and installed. In this category, examples such as hijackers and malware potentially could be included by the spyware distributors. In addition, all programs involved in any form of Internet communication were specified in a category called "Internet". Finally, the category entitled "Host" specifies which file-sharing tool that carried what spyware[5]. In the cases where our empirical results could confirm the view shared by anti-spyware tools, the markers in the table are declared with bolded capital letters.

When analysing the outgoing network communication from the spyware components, we discovered that most of this traffic was not sent in clear text. This means that the transactions between the spyware components and their corresponding servers were either

5. B is for BearShare, I for iMesh, K is for KaZaa, L for LimeWire, and M for Morpheus.

Name	Host	Adware	Spybot	Download	Internet
BroadcastPC	M	x	x	x	X
KeenValue	K	x	x	X	X
Morpehus	M	X	x	X	X
BargainBuddy	I, K	x	x	x	
TopMoxie	L, M	x	x	x	
Cydoor	I, K	x	x		X
Gator	I, K	X	x		X
SaveNow	B	X	X		X
BonziBuddy	L	x	x		
Web3000	I	x	x		
ShopAtHomeSelect	I		X	X	X
WebHancer	K		x	x	
BrilliantDigital	K	x		X	X
MoneyMaker	L, M	X		X	X
Claria	I, K	x			X
iMesh	I	x			X
WeatherCast	B	x			X
CasinoOnNet	L	x			
MyBar	I, K, M	x			
New.Net	I			X	X
FavoriteMan	I			x	

Table 5.1 Identified Spyware Programs.

obfuscated or encrypted. This is also an explanation to why we were able to only identify two genuine spybot components. Since most traffic was sent in non-clear text, we could not really measure the extent to which such traffic was broadcasted. However, we did manage to identify some network traffic sent to spyware servers on the Internet that included, e.g., web sites visited, zip codes, country, and information about programs and operating system versions on the local host. In example, one of the spybot programs (ShopAtHomeSelect) that was found bundled with the iMesh file-sharing tool transmitted Internet browsing history records to several invoked servers on the Internet. The Internet records that were transmitted could be correlated to the web sites included in our preconfigured web surfing program.

	KaZaa Lite K++	KaZaa	Alteration
1. CPU usage (in%)	0.015	0.48	0.47
2. RAM usage (in%)	1.4	14	12.6
3. Addition of new files	50	780	730
4. Change in hard disk size (in MB)	8.6	46	37.4
5. Amount of network traffic (in MB)	0.6	29	28.4
6. No. of programs involved in Internet communication	1	11	10
7. No. of corresponding servers	60	349	289
8. No. of spyware programs installed	0	8	8

Table 5.2 Resource Utilisation Measurements.

5.3.2.4 Experiment 2

A compilation of the results from the resource utilization measurement can be seen in Table 5.2. The measurements indicate that if KaZaa was installed, the rates for consumption of both system capacity (categories 1-4) and network bandwidth (categories 5-7) were significantly higher. This can be explained in that the spyware programs included in KaZaa affected both consumption of system capacity and network bandwidth. The high amount of network traffic was due to that the spyware components invoked numerous spyware servers on the Internet for the gathering of ads, pop-ups and banners. The accumulated local storage of collected commercial messages can have noticeable consequences on hard drive size, which also was the case for KaZaa.

In Table 5.2, the measurements for the reference subject is subtracted from the file-sharing tools. The column entitled "Alteration" is represented by the difference between KaZaa and KaZaa Lite K++, that is; the spyware resource usage. Interestingly, three computer resources were significantly affected by the installation of spyware. In the first category of Table 5.2, the occurrence of spyware had a measurable effect on CPU usage, KaZaa used 32 times more CPU capacity than KaZaa Lite K++. In category two, a significant difference was measured where the installation of KaZaa resulted in a ten times, or 65MB, increase of RAM usage. Finally, spyware pro-

grams had an imperative effect on the amount of network traffic generated by the file-sharing tools. More specifically, there was a 48 times augmentation of network traffic due to the spyware programs bundled with KaZaa. So, in contrast to KaZaa, installing a clean file-sharing tool (i.e., KaZaa Lite K++) caused marginal impact to system consumption and network bandwidth. However, due to the occurrence of spyware in file-sharing tools (see Table 5.1), users with several such applications installed will, as a result of aggregate spyware activity, suffer from a continuous system and network degrading.

5.4 Discussion

Based on the findings in 5.3, we can conclude that spyware programs exist, that they engage themselves in Internet communication, that they transmit user data, and that their existence have a negative impact on system and network capacity. Since we also can conclude that spyware programs are bundled with highly popular file-sharing tools[6], we can make out that spyware in accumulation may have a negative impact on networks and systems. In fact, the occurrence of spyware might decrease the overall utility of belonging to a large network such as a P2P file-sharing network. Thus, it might be relevant to elaborate on the theory of negative network effects to see whether spyware programs can threaten a large network.

In a model (Table 5.3), we specify in what ways spyware might decrease the utility of belonging to a large virtual network. The baseline is that spyware programs intrude systems and networks, but since they profit from user data they also intrude user privacy. In the model, the intrusions are classified as moderate, severe and disastrous.

On user effects, some P2P providers include spyware in order to maximise profitability. Spyware may collect user data (such as e-mail addresses for spam distribution, surf records for personalised advertisement exposure, etc.) for commercial purposes. At present, spyware programs as such are rather benign, but cause problems to user privacy. In general, privacy is the right of individuals to control

6. As an example, there are more than 350 million downloaded instances of KaZaa [2].

	User	Computer	Network
Moderate	Commercially salable data	Consumption of capacity	Consumption of bandwidth
Severe	Personal data	Inferior code dissemination	Malware distribution
Disastrous	Critical data	Takeover	Breakdown

Table 5.3 Spyware Effects.

the collection and use of information about themselves [16]. This means that users should be able to decide for themselves, when, how, and to what extent information about them is communicated to others. Even though the user data exemplified in this category may not be that sensitive, spyware programs ignore user rights, and must therefore be considered privacy-invasive.

A more troublesome concern is the distribution of personal data, such as personal details (name, gender, hobby, etc.), e-mail conversation, and chat records. This may be the result of spyware techniques intended not only for commercial purposes, but also motivated by malicious intentions. Although, such spyware programs may not be that wide-spread today, a technological platform for these kinds of operations is available. This mean that although the probability of being infected by such a spyware is very low, the consequences may be devastating.

A third view would be if the spyware program updates on the servers were replaced with, e.g., keyloggers. In effect, harmful software could be distributed to vast groups of P2P tool users with the purpose of transmitting personally critical information such as financial data, private encryption keys, digital certificates or passwords. In reflection, financial threats from spyware programs may signify disastrous outcomes to vast groups of users.

In the experiments, we established a correlation between the presence of spyware programs and the consumption of computer capacity. Typically, spyware components utilised significant amounts of system resources, rendering in that computer resources were exploited in a larger extent than would otherwise be necessary. In accumulation, spyware operations degrade system capacity.

Also, it is problematic to comment on the quality of the code in the spyware programs, since the software requirements that have been used during the development process are left out in obscurity. The

result can be that possibly inferior code is executed locally, which may have a negative influence on the entire system (i.e., not only to security). For example, as an effect of executing insufficient code, a system may lack performance or crash with, e.g., loss of important data as a result. In addition to this, software vulnerabilities may be exploited by malicious persons when breaking into a system, or when infecting it with destructive software (e.g., viruses).

As an utmost consequence, spyware programs deprive control over the system from the system owner. In effect, the installation of spyware programs may render in further installations of malware such as viruses and/or Trojans. Local services that are based on defect code and executed without the knowledge of the system owner are vulnerable to exploits, which may allow malicious actors to gain access over the computer. This is a disastrous situation because a takeover of system control affects both the local system and the surrounding network. A conquered system can be used as a platform for further distribution of malware.

At the network level, spyware operations in accumulation may contribute in network congestion. On one hand, the effects are unnecessary costs for network maintenance and expansion. On the other hand, network performance may be degraded. In either case, it is the network users that in the long run bear the costs.

The operations performed by spyware programs are approaching the operations of a virus with both a distribution and a payload part. Since users install, e.g., file-sharing tools that contain spyware programs on a voluntary basis, the distribution part is taken care of by the users themselves. This makes spyware programs function like a slowly moving virus without the typical distribution mechanisms usually otherwise included. The general method for a virus is to infect as many nodes as possible on the network in the shortest amount of time, so it can cause as much damage as conceivable before it gets caught by the anti-virus companies. Spyware, on the other hand, may operate in such a relatively low speed that it is difficult to detect. Therefore, the consequences may be just as dire as with a regular virus. The payload of a spyware is usually not to destroy or delete data, but to gather and transmit user information, which could be veritably sensitive. An additional complicating factor is that anti-virus companies do not generally define spyware as virus, since it does not typically include the ability to autonomously replicate itself. Overall, the nature of spyware substantiates the notion that malicious actions launched on computers and networks

get more and more available, diversified and "intelligent", rendering in that security is extensively problematic to uphold.

In theory, even a large network such as a P2P network may suffer an ultimate breakdown if it is continuously flooded with data. Should spyware programs continue to increase in number and to be more and more technologically refined, a network breakdown might be a final step. Although, in reality, this is not a plausible outcome. Nonetheless, if security and privacy risks are increasing as a result of being part of a P2P network, the positive value of using an application and thus belonging to that network will likely decrease. If users should experience that a threshold value (where the negative effects overthrow the positive aspects of using the application) is overstepped, then they will restrain from utilising that network. However, the experiment results indicate that even though spyware programs operate over P2P file-sharing networks, their effects are thus far rather modest. At least when it comes to system and network consumption. On the other hand, spyware programs that invade user privacy must be looked upon seriously. Spyware technologies mainly involved in gathering user data have a true value potential for marketers and advertisers. If these privacy-invasive activities should continue to evolve, there might be a great risk that spyware will be engaged in more malicious activities than simply fetching anonymised user/work station data. If so, that can lead to negative network effects and thereby cause a network to become less useful.

Hidden spyware components permit distribution of privacy-invasive information and security breaches within the network. Due to the construction of spyware, it may collect information that concerns other parties than only the work station user, e.g., telephone numbers and e-mail addresses to business contacts and friends stored on the desktop. In the context that spyware usually is designed with the purpose of conveying commercial information to as many users as possible, not only the local user may be exposed to negative feedback of spyware. As well, the business contacts and friends may be the subjects of network contamination, e.g., receiving vast amounts of spam or other unsolicited content.

With the continuous escalation of spyware programs and the refinement of spyware technologies, network availability may be degraded to such an extent that ordinary transactions are overthrown by obscure malware traffic. A disastrous situation may occur where a network is seriously overloaded by malware distributed by compu-

terised systems that are controlled by malicious actors. In conclusion, spyware activity may persuade users to abandon networks.

5.5 Conclusions

Based on the discussions of spyware and on the findings from the two experiments, we can conclude that spyware have a negative effect on computer security and user privacy. We have also found that a subsequent development of spyware technologies in combination with a continuous increase in spyware distribution will affect system and network capacity. A disastrous situation may occur if a network is seriously overloaded by different types of spyware distributed by computerised systems that are controlled by malicious actors. Then, the risk is a network breakdown. However, a more plausible outcome may be that users will abandon the network before that happens. In effect, spyware has the potential to overthrow the positive aspects of belonging to a large network, and network owners should therefore be very careful about permitting such programs in applications and on networks.

5.6 References

[1] S.-Y. Choi, D.O. Stahl, and A.B. Winston, "*The Economics of Electronic Commerce",* Macmillan Technical Publishing, Indianapolis IN, 1997.

[2] C|Net Download.com., http://www.download.com/, 2004-06-29.

[3] "Emerging Internet Threats Survey 2003", commissioned by Websense International Ltd., February, 2003. http://www.websense.com/company/news/research/Emerging_Threats_2003_EMEA-de.pdf, 2004-06-29.

[4] 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.

[5] M.L. Katz, and C. Shapiro, "Systems Competition and Network Effects", in *Journal of Economic Perspectives 8:93-115*, 1994.

[6] M. McCardle, "How Spyware Fits into Defence in Depth", SANS Reading Room, SANS Institute, 2003. http://www.sans.org/rr/papers/index.php?id=905, 2004-06-29.

[7] A. Oram, "*Peer-To-Peer: Harnessing the Benefits of a Disruptive Technology"*, United States of America: O'Reilly & Associates Inc., 2001.

[8] PestPatrol, http://research.pestpatrol.com/Lists/NewPests(Pest Counts).asp, 2004-06-29.

[9] 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.

[10] D. Schoder, and K. Fischbach, "Peer-to-Peer (P2P) Computing", in *Proceedings of the 36th IEEE Hawaii International Conference on System Sciences (HICSS'03)*, IEEE Computer Society Press, Los Alamitos CA, 2003.

[11] C. Shapiro, and H. Varian, *"Information Rules"*, HBS Press, Boston MA, 1999.

[12] E. Skoudis,*"Malware - Fighting Malicious Code"*, Prentice Hall PTR, Upper Saddle River NJ, 2004.

[13] "Spyaudit", commissioned by Earthlink Inc., http://www. earthlink.net/spyaudit/press/, 2004-06-29.

[14] 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.

[15] K. Townsend, "Spyware, Adware, and Peer-to-Peer Networks: The Hidden Threat to Corporate Security" (technical white paper), PestPatrol, 2003., http://www.pestpatrol.com/Whitepapers/PDFs/ SpywareAdware P2P.pdf, 2004-06-29.

[16] A. Westin, "*Privacy and Freedom"*, Atheneum, New York NY, 1968.

[17] J. Wieslander, M. Boldt, and B. Carlsson, "Investigating Spyware on the Internet", in *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, Gjövik Norway, 2003.

CHAPTER

6

Analysing Countermeasures Against Privacy-Invasive Software

International Conference on Software Engineering Advances, 2006

Martin Boldt and Bengt Carlsson

User privacy is widely affected by the occurrence of privacy-invasive software (PIS) on the Internet. Various forms of countermeasures try to mitigate the negative effects caused by PIS. We use a computer forensic tool to evaluate an anti-spyware tool, with respect to found PIS over a four years period. Within the anti-spyware tool PIS was slowly identified, caused classification problems, and formerly classified PIS were sometimes excluded. 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.

6.1 Introduction

Technology has revolutionized the way we collect, store and process information. With the help of information technology it is possible to accumulate huge data quantities for instant or 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 [21]. 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). *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]. However, the term spyware is also used at a higher abstraction level to include any software that users dislike [3, 13]. Unfortunately there does not exist any proper definition of this notion of the term. All this software are to various degree encapsulated inside the term PIS [6]. Our use of the concept of *privacy* lies within Warren and Brandies original definition, "the right to be let alone." [42]. 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 sneaks 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, 7, 25].

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 [22, 33]. 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 [41]. 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, 31]. 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 [25]. Which leaves the user with trespassed systems containing unsolicited harmful software, that result in a reduction of performance, stability, privacy and ultimately the security [7].

In this paper however, we use *computer forensic* tools and methods to evaluate the accuracy of PIS countermeasures. This paper also touch upon the evolution of PIS countermeasures and the legal disputes between developers of PIS and related countermeasures.

6.2 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 [23]. In an ongoing struggle between anti-virus companies and virus distributors, refined detection mechanisms have to fight more and more sophisticated viruses searching for competitive advantages over each other. Anti-spyware vendors face three major problems to solve.

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.

The most common technique used when countering PIS is the *signature based* identification which relies on a database holding signatures of known PIS. A signature captures unique properties of PIS, and could be thought of as the associated 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.

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 [39]. The most recent case involves the online marketing company "180Solutions" that sued firewall company "Zone Labs" for classifying their advertising client as spyware [48]. 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 [8, 32]. 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.

6.3 Computer Forensics

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* [11]. 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 [39]. 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* [27] 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.

6.4 Investigation

In previous investigations of PIS we used a manual investigation method that is based on system state preservation [7, 25, 46]. 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 the standard BSD Unix component *dd*. 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 snapshots allow 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.

In an experiment we used this method 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 [20, 29].

In earlier experiments we have investigated 10 different anti-spyware tools but in this work we choose to instead evaluate a single tool over four years development instead. We choose to investigate an anti-spyware tool that is developed by Lavasoft which is called Ad-Aware. This specific tool was selected since it is the most downloaded anti-spyware tool from Download.com (October 2005) and

since we could locate versions of this tool for the years 2002-2005. The experiment used 13 identical physical computers holding four versions of the three most downloaded P2P file-sharing tools; iMesh, LimeWire, and Kazaa, together with one reference machine without any file-sharing tool installed. 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 [35].

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 as 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 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 efficient techniques that are highly useful for an investigator. Such techniques are for instance pre-indexation of data, and a known-file-filter. Pre-indexation means 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.

Any 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 [19]. This information was then checked against two resources for classification [16, 36].

6.5 Results

In Table 6.1 the total number of PIS, cookies, registry 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[1]. The shadowed part shows the added components found by a present or future version of Ad-Aware, i.e. the actual protection against certain version of the P2P programs.

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

Table 6.1 **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.**

In general, present versions of Ad-Aware find more components than older. 2002, 2003 and 2004 versions of the P2P-programs show a many times increase of added components. Ad-Aware 2005 reported fewer components on average than the 2004 version of the program.

1. We used one instance of each version of Ad-Aware. In Table 6.1 the release month of this instance is given.

Figure 6.1 presents the amount of PIS programs, registry keys, and other traces that are being injected into a system when a P2P-program is installed and been running for 72 hours. Registry keys are not complete programs but are used by PIS during execution. The most dominating group of traces consist of registry keys followed by not specified traces, e.g. suspicious files and folders. The actual number of executable PIS is much lower 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.

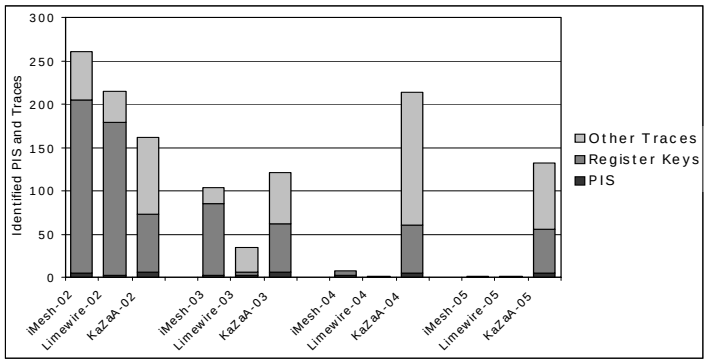

Figure 6.1 Number of bundled PIS programs, registry keys, and suspicious files/ folders for iMesh, LimeWire and Kazaa reported by Ad-Aware over a four year

In Table 6.2 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 6.2. 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.

Ad-Aware	aug-00	jun-01	mar-02	mar-03	sep-04	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

Table 6.2 **Number of PIS in three different P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions 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.**

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 6.2 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.

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

Table 6.3 **Total number of undiscovered PIS programs in three different P2P-programs (iMesh, LimeWire and Kazaa) measured by six different versions (3.5 to SE1.06) of Ad-Aware.**

In Table 6.3 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 6.3) 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..

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

Table 6.4 **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.**

Table 6.4 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.

6.6 Discussion

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 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 6.3 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, suffers 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 certain behaviour. Also, a lack of a deepened context analysis may influence the amount of false positives, i.e. warnings, generated by the anti-spy-

ware tool that does 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 was 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[2]. Without such an agreement it is a more arbitrary task to distinguish PIS from legitimate software than separating malicious software, such as virus and worms, from legitimate software. Since the inner workings of PIS does not necessarily include any malicious behaviours, they rather include normal behaviour such as showing content on the screen (advertisements), it is not possible for countermeasures to only target PIS behaviour when distinguishing PIS from legitimate software. Instead they need to incorporate user consent when distinguishing between legitimate and illegitimate software. Without proper techniques that safeguard true informed user consent during installation, it is extremely hard (if not impossible) for any spyware countermeasure to accurately decide on what software to target since there exists no common guidelines to follow.

2. Despite we only used one anti spyware-tool, a lot of different versions during several years were investigated.

This fact combined with that the software that anti-spyware vendors classify as PIS are developed by companies that are ready to take legal actions if needed, pose a great risk for these vendors. 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 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 security and privacy, on both home computers and mobile devices. TRUSTe has started one such promising approach called "Trusted Download Program" in which they will provide a guideline on how to distinguish legitimate software [41]. Based on this guideline they will publish a white-list of approved applications. Any software vendor submit their software for certification must also enter into a contract with TRUSTe in which their software functionality is specified. Using these guidelines TRUSTe continuously evaluate the correctness and ongoing compliance of all certified software.

6.7 Conclusions

Identifying and removing PIS and keeping/protecting legitimate files are major problems to solve for anti-spyware vendors. 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. This paper evaluated the accuracy of a

leading anti-spyware tool called Ad-Aware which uses signatures to counteract PIS. The effectiveness of comparable versions of Ad-Aware 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. This imply that the signature based approaches, which is de facto standard in anti-virus tools, is not ideal when targeting PIS. We believe the failure from anti-spyware tools to deal with the three qualities above not only rely on obsolete identification techniques, but also on the lack of a general agreement on what should be considered as privacy-invasive behaviour of software.

It is of most importance to develop routines that allow users to make informed decisions during the software installation process, on whether to install a certain software or not. Until such routines and mechanisms are being widely deployed, computer users risk being victims of systematic privacy invasions in their digital environment from questionable actors.

6.8 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, and B. Carlsson, "Privacy-Invasive Software and Preventive Mechanisms", in *Proceedings of the International Conference on Systems and Network Communications*, Papeete Tahiti, 2006.

[7] M. Boldt, B. Carlsson, and A. Jacobsson, "Exploring Spyware Effects", in *Proceedings of the Eighth Nordic Workshop on Secure IT Systems*, Helsinki Finland, 2004.

[8] J. Bruce, *"Defining Rules for Acceptable Adware", in the Fifteenth Virus Bulletin International Conference (VB2005)*, Dublin Ireland, 2005.

[9] B. Carrier, "*File System Forensic Analysis"*, Addison-Wesley Professional, Upper Saddle River, NJ, 2005.

[10] H. Carvey, "*Windows Forensics and Incident Recovery"*, Addison-Wesley, Upper Saddle River, NJ, 2005.

[11] E. Casey, "*Digital Evidence and Computer Crime: Forensic Science and the Internet"*, Academic Press, London UK, 2004.

[12] D. M. Chess and S. R. White, "An Undetectable Computer Virus", in *Virus Bulletin Conference*, Orlando FL, 2000.

[13] E. Chien, "Techniques of Adware and Spyware", in *Fifteenth Virus Bulletin International Conference (VB2005)*, Ireland, 2005.

[14] F. Cohen, "Computational Aspects of Computer Viruses", in *Computers & Security*, Vol. 8, Issue 4, San-Fransisco CA,1989.

[15] F. Cohen, "Computer Viruses - Theory and Experiments", in *IFIP-Sec 84*, Toronto Canada, 1984.

[16] Computer Associates Spyware Information Center, http://www3.ca.com/securityadvisor/pest/, 2006-01-03.

[17] Download.com, http://www.download.com, 2006-01-03.

[18] "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.

[19] D. Farmer, and W. Venema, *"Forensic Discovery"*, Addison-Wesley, Upper Saddle River NJ, 2004.

[20] 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.

[21] S. Görling, "An Introduction to the Parasite Economy", in EICAR 2004, Luxemburg, 2004.

[22] G. Hoglund, and G. McGraw, *"Exploiting Software - How To Break Code"*, Addison-Wesley, Boston MA, 2004.

[23] L. Hunter, "*Stopping Spyware"*, Addison-Wesley, Upper Saddle River, NJ, 2005.

[24] A. Jacobsson, "Exploring Privacy Risks in Information Networks", in *Blekinge Institute of Technology Licentiate Thesis Series No. 2004:11*, Sweden, 2004.

[25] 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.

[26] Lavasoft, http://www.lavasoft.com, 2006-01-03.

[27] E. Locard, *"L'enquête criminelle et les méthodes scientifiques"*, Flammarion, Paris France, 1920.

[28] R. Martin, "Spy vs. spy*"*, in *Fortune Small Business*, Vol. 14, No. 4, 2004.

[29] A. Oram, "*Peer-To-Peer: Harnessing the Benefits of a Disruptive Technology"*, United States of America: O'Reilly & Associates Inc., 2001.

[30] 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.

[31] S. Shukla and F. Fui-Hoon Nah, "Web Browsing and Spyware Intrusion", in *Communications of the ACM*, Vol. 48, Issue 8, 2005.

[32] 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*.

[33] E. Skoudis,*"Malware - Fighting Malicious Code"*, Prentice Hall PTR, Upper Saddle River NJ, 2004.

[34] Spyaudit, commissioned by Earthlink Inc., http://www.earthlink.net/spyaudit/press/, 2006-01-03.

[35] Spyware is Windows-only, http://www.securityfocus.com/news/9696/, 2006-01-03.

[36] SpywareGuide.com, http://www.spywareguide.com, 2006-01-03.

[37] Stay Safe Online, *"AOL/NCSA Online Safety Study - December 2005"*, http://www.staysafeonline.org/pdf/safety_study_2005.pdf, 2006-01-03.

[38] 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.

[39] Threats Against Spyware Detectors, Removers, and Critics, http://www.benedelman.org/spyware/threats/, 2006-01-03.

[40] K. Townsend, "Spyware, Adware, and Peer-to-Peer Networks: The Hidden Threat to Corporate Security" (technical white paper), Pest-Patrol, 2003., http://www.moorecomputing.net/SpywareAdwareP2P.pdf, 2006-01-03.

[41] TRUSTe - Make Privacy Your Choice, http://www.truste.com, 2006-01-03.

[42] S.D. Warren and L.D. Drandeis, *"The Right to Privacy*", in *Harward Law Review*, Vol. 4, Issue 5, 1890.

[43] Webroot Software, *"State of Spyware - Q3 2005*", http://www.webroot.com/resources/, 2006-01-03.

[44] Webroot Software - Phileas, http://www.webroot.com/resources/phileas/, 2006-01-03.

[45] A. Westin, *"Privacy and Freedom"*, Atheneum, New York NY, 1968.

[46] J. Wieslander, M. Boldt, and B. Carlsson, "Investigating Spyware on the Internet*"*, in *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, Gjövik Norway, 2003.

[47] X. Zhang, *"What Do Consumers Really Know About Spyware?"*, in *Communications of the ACM*, Vol. 48, Issue 8, 2005.

[48] Zone Labs Sued Over Spyware Classification, http://www.securityfocus.com/brief/68, 2006-01-03.

CHAPTER

7

Privacy-Invasive Software and Preventive Mechanisms

International Conference on Systems and Network Communications, 2006

Martin Boldt and Bengt Carlsson

Computers are increasingly more integrated into peoples' daily lives. In this development, user privacy is affected by the occurrence of privacy-invasive software (PIS), sometimes loosely labelled as spyware. The border between legitimate software and PIS is vague and context dependent, at best specified through End-User License Agreements (EULA). This lack of spyware definition result in that current countermeasures are bound to noticeable misclassification rates. In this work we present a classification of PIS from which we come to the conclusion that additional mechanisms that safeguard users' consent during software installation is needed, to effectively counteract PIS. We further present techniques that counteract PIS by increasing user awareness about software behaviour, which allow users to base their software installation consent on more informed decisions.

7.1 Introduction

A powerful component in any business strategy is user/customer information gained either with or without violating the privacy and security of users. In general, the company with the most information about its customers and potential customers is usually the most successful one [25]. This situation has given raise to a parasitic market, where questionable actors focus on short time benefits when stealing personal information for faster financial gain [6, 9, 18]. In addition, this situation is further fueled by money from advertisers who want their online ads distributed; often ending up in advertising software (*adware*) on users' trespassed computers [8, 29]. Throughout this paper we describe such software as *privacy-invasive software* (PIS); ranging from legitimate software and spyware to truly malicious software (*malware*). Some of the most common technical problems associated with PIS include [1, 9, 17, 21, 32]:

- Unauthorised resource utilisation causing deteriorate system stability
- Third party software installation without user consent
- Displaying of unsolicited advertising content at varying frequency and substance
- Settings and properties of other software are being changed
- Personally identifiable information is covertly transmitted to third parties
- Poor or non existing removal mechanisms
- Time invested in recovering systems from such unsolictited programs

Various sources claim that up to 80% of all Internet connected computers have one or more spyware infections on their computer systems [3, 14, 35]. One general problem concerning these investigations is the lack of a proper definition of what is being measured and investigated, i.e. what spyware actually is [17]. There exist numerous terms that are used as synonyms to spyware, e.g. evilware, scumware, snoopware, thiefware, or trackware. These terms are all used to group software together that is somehow being disliked by users, regardless of being illegal or not [6, 27]. The border between legitimate and illegitimate software is non existing, or at least very vague and context dependent. Wrongly classified software render in that legitimate software vendors get their products labelled as spyware, or some of its synonyms. To protect their busi-

ness, these vendors take legal actions against the responsible developers of such inaccurate spyware countermeasures [30]. This means that the absence of a proper spyware definition result in legal disputes between software vendors and developers of spyware countermeasures. In the end, users have no alternative but to put their confidence in inaccurate countermeasure tools that leave them with trespassed systems [19].

Today, much software that incorporates advertisements is regarded as spyware by the users, which in turn render in distrust of advertised financed software in general. This is very unfortunate since advertise financed software development is a powerful tool, that allow vendors to provide a product "free of charge". This way users do not have to pay since the software developer get revenues from the advertising agency for delivering ads to the users [8]. But as have been shown over recent years, there must exist regulations and rules of conduct that control how these techniques interface with the users [6, 11].

7.2 Spyware and User Consent

In major operating systems today, e.g. Microsoft Windows, software installations are carried out in a rather ad-hoc manner. Basically the user retrieves a software package from a source, such as a non trusted Web site, and then executes it on the system. Even though some operating systems offer more standardized installation methods, e.g. FreeBSD's ports system, this method is still available in parallel [15]. The disadvantage with this ad-hoc software installation method is the lack of knowledge about the software that enters the system [4, 26]. Also, the instrumentation that allow users to evaluate the software prior to the actual installation is inaccurate, or non-existing. Without such instrumentation it is troublesome for the users to give an *informed consent* for the software to enter their system. Our use of the term informed consent is based on the work by Friedman, Felten, and Millett, in which they divide the word *informed* into: *disclosure* and *comprehension* [16]. The word *consent* is divided into: *voluntariness, competence,* and *agreement*.

Today, users give their consent to software installations by accepting the terms stated in End-User License Agreements (EULA). Unfortunately many computer users today are not capable of comprehending these EULAs, since they are disclosed in a very legal,

formal, and lengthy manner [30]. A license agreement that includes well over 6000 words (compared to US constitution which includes 4616 words) is not unusual, which users need a degree in contract law to understand [6]. Even users that have the prerequisite knowledge do not have the time to read through lengthy EULAs each time they install new software, resulting in that most users simply accept the license agreements without reading through them first [27]. Due to this it is next to impossible for users to reach sustainable trust-related decisions during software installation since today's computing systems doesn't provide sufficient support for making such decisions. By not being able to evaluate software entering their system, users unknowingly allow illegitimate software to enter.

In addition to the threat from infection of inferior software, the lack of informed consent during software installations also impose very vague user awareness on what they have agreed upon; e.g. users can't deduce the pop-up ads on the computer screen with an earlier approval of the corresponding EULA [6]. The underlying problem is how software vendors should disclose information to their customers about their product's implications on the user and the user's system. More importantly, how to present it in a clear and comprehensive manner towards the users. Vendors of PIS target this problem when using deceptive methods for deploying their software on users' machines. Since these attacks target the human/computer interface it should be understood as a semantic problem, not a syntactic one, or as Bruce Schneier put it "any solutions will have to target the people problem, not the math problem" [24].

In the end, it is of paramount importance, for both users and legitimate software vendors, that a clear separation between acceptable and unacceptable software behaviour is established [6, 27]. However, we believe that an acceptable behaviour is context dependent, i.e. what one party regard as acceptable software behaviour is regarded as unacceptable by others. Therefore, users need to know what they install and, with the help of aiding mechanisms, learn to distinguish between what they believe is acceptable and unacceptable software, prior to any actual software installation on their system.

7.3 Software Classifications

In the introduction section we defined PIS as any software violating user's privacy, ranging from legitimate software to malware, with spyware in-between. As a matter of fact we intend to exclude spyware from the list of PIS by either classifying them as legitimate software or malware, from a preventive point of view. Before taking this initiative we first look at current spyware classifications followed by the classification of PIS before, in the discussion area, coming back to the prevention view.

7.3.1 Spyware Classification

The term spyware is used at two different abstraction levels [1, 17]. In the more precise version, the term is defined as "any software that monitors user behaviour, or gathers information about the user without adequate notice, consent, or control from the user" [1]. On the other hand, the more abstract use of the term has showed itself hard to define [1, 6]. This notion of the term is often used to describe any software that is disliked by users, even though properly introduced toward the users and with their "uninformed" consent. As a group of software, spyware is located in between legitimate and malicious software, but unfortunately the exact border has not been unveiled [6]. If we could identify these borders and thereby both differentiate between spyware and malware, and secondly between spyware and legitimate software we would have come close to encapsulating spyware.

One difference between spyware and malware is that spyware, to a large extent, target sensitive but not critical information, while malware do. However, the main difference between spyware and malware is that spyware present users with some kind of choice during the installation or entrance into their system [17]. This means that any software that installs itself without asking for the user's permission should no longer be treated as spyware, but instead as malware. Do note that there exists malware which extends the behaviour of spyware, but these are no longer situated in between legitimate software and malware [28]. The difference between legitimate software and spyware is based on the degree of user consent associated with it, i.e. informed consent means legitimate software. Hence, the problem really boils down to inaccurate mechanisms for users to evaluate the software's degree of appropriateness to enter their sys-

tem (see Section 7.2). Even though there exist no accurate definition of the wider use of the term spyware, legislation against these threats is being implemented in a number of nations. Specifying legislative actions against something that is not properly defined and constrained impose risks on items located close to the target [34].

	Positive Consequences	**Negative Consequences**
High Consent	Overt providers	Double agents
Low Consent	Covert supporters	Parasites

Table 7.1 Classification of spyware with respect to user awareness and permission (high or low) and user consequences (positive or negative).

Warkentin et al. present a classification of spyware as the combination of *user consequences* and the *level of consent* from the user; see Table 7.1 [34]. User consequences are divided into either positive or negative, and the consent level is represented as a continua spanning between high and low. They define spyware in a two by two matrix, with the following four different types: *overt provider*, *covert supporter*, *double agent*, and *parasite*. The overt provider is synonymous to any legitimate software, while a covert supporter have less, or none, user consent. Both types provide the user with some useful service, i.e. the existence of the software is in some sense beneficial for the user. Double agents act as trojan horses, which obtain user consent for providing one task, but really execute another task causing unexpected negative consequences. Finally, the parasite has no user consent what so ever and it impairs negative consequences on the user, and his system.

7.3.2 PIS Classification

In this paper we further improve the classification provided by Warkentin et al. in three ways. First, we introduce an intermediate value on both the consent and consequence axis, so that the matrix size is increased to three by three. This results in that the consent level is classified as *high*, *medium*, or *low.* By including this middle value we prepare a base for further discussions on an important group of PIS that get user consent, but not as a result of an informed decision.

Secondly, we remove all positive consequences from the grading on the consequence axis. Since we focus on spyware we exclusively focus on the negative consequences towards users. The grading consists of *negligible*, *moderate*, and *severe* negative user consequence. Software with negligible negative consequences include legitimate software which always have some degree of negative impact, e.g. with regard to resource utilisation.

Thirdly, we split user consequence property into both *direct negative consequences* and *indirect negative consequences*. This means that user consequences consists of both the direct consequences of the software, i.e. what it is designed to do, combined with the indirect consequences, i.e. how the mere existence of the software affect the whole computer system. Note that the indirect consequences are not visible in our matrix, but is used when describing the various entries inside it.

	Negligible Negative Consequences	**Moderate Negative Concequences**	**Severe Negative Conceuences**
High Consent	1) Legitimate software	2) Adverse software	3) Double agents
Medium Consent	4) Semi-transpar-ent software	5) Unsolicited software	6) Semi-para-sites
Low Consent	7) Covert software	8) Trojans	9) Parasites

Table 7.2 Classification of privacy-invasive software with respect to user's informed consent (high, medium and low) and negative user consequences (negligible, moderate andsevere).

Since the wider use of the term spyware includes so much more than only information gathering software, any classification trying to capture all these various types of behaviour must basically contain any software that includes questionable user consent and negative user consequences [17]. Therefore we enumerate every software permutation that exist as a combination of user consent and negative software consequences. By doing so we capture every form of "active" legitimate software, spyware and malware. Our

three by three matrix then consists of nine different types of PIS, ranging from legitimate software to full fetched malware, such as worms or viruses. Anyone of these nine types of PIS collect information about the user and his/her system, or negatively affects the user's computer experience.

Our classification of PIS in Table 7.2 presents three different groups of software, the first has high user consent to provide its service (top row), the second group include software that has some kind of user consent but it does not correspond to the software's full behaviour (middle row), and the last group include software that does not have any consent from the user at all (bottom row). By inspecting these three groups as software with PIS behaviour, we can narrow down what types of software that should be regarded as spyware, and which ones that should not. The top row includes software that has received full permission from the user, and where the behaviour is fully transparent towards the user, i.e. they should not be regarded as spyware.

1. *Legitimate software* has the user's full consent and provides some beneficial service to the user. Information collection and other potential spyware behaviour should be treated as a legitimate functionality, as long these are fully transparent to the user.
2. *Adverse software* has the same properties as any legitimate software, but with the exception that it renders in increased negative consequences. However, this software does not include any covert behaviour from the users and should therefore not be labelled as spyware.
3. *Double agent* has been designed to cause a number of negative effects on the user's system, but all these consequences are fully transparent to the user, i.e. user has given his/her consent based on an informed decision. One example includes installation of file-sharing tools that are bundled with questionable software. A user might chose to install these tools despite being aware of the consequences, since the gained benefit of the tool motivates it [17].

The middle row in Table 7.2 includes software that has gained some sort of consent from the user, but it was not based on an informed decision and should therefore be regarded as spyware.

4. *Semi-transparent software* includes any software that provides a requested and beneficial service toward the user, but where some essential functionality is not communicated to the user.

Even if the covert functionality does not impair any negative consequences, the software could introduce vulnerabilities to the user's system. This is especially alarming since the user is unaware of it and therefore unable to address the threat. Based on this reasoning this type of software should be regarded as spyware.

5. *Unsolicited software* is installed on users systems without their explicit consent, i.e. requested software that covertly installs further software. Commonly, users give their permission to the first software, but are usually, to various degrees, unaware of the existence and behaviour of any third party bundled software. This group of software should be labelled as spyware.
6. *Semi-parasite* is a type of spyware that is pushed on users when, for instance visiting Web pages. These software often deceive users into thinking they are needed to access for example a Web page. Since no information about the consequences are presented to the user they are left unaware of the covert functionality that cause major negative consequences.

Software that installs and executes without any user consent at all is represented in the bottom row of Table 7.2. By covertly sneaking inside users' systems such software has clearly crossed the line of what should be regarded as acceptable behaviour, and should therefore be labelled as malware. This group of malware is further divided into three types depending on the degree of negative user consequences they are causing.

7. *Covert software* is software that secretly installs themselves on systems without causing any direct negative consequences. However, threats from indirect negative consequences mark these software as malware, see reasoning for semi-transparent software.
8. *Trojan* is software that deceives users into installation in the belief that they provide some beneficial service, but which include covert functionality that impose negative consequences on the user.
9. *Parasite* includes software with pure negative consequences that gains entrance to the user's system without his/her awareness or consent, e.g. through vulnerabilities. Once inside, these software does whatever the attacker has designed them to do.

This PIS classification emphasis on user's informed consent, i.e. the user competence and comprehension of software behaviour is

essential for the classification of PIS. High comprehension means legitimate software, medium means spyware, and no comprehension means malware, if all users really are able to make such a decision. If not, which is the case today, most of the software on the first row will belong to the spyware section. This classification is user dependent, i.e. more knowledgeable users may change the awareness in a dynamic way, which is further discussed in the next section. The PIS classification also emphasis on negligible, moderate or severe negative consequences when the system is misused. This is essential when more accurate mechanisms to inform users about software behaviours are deployed, allowing users to individually decide what consequences are acceptable (compared to the software's positive effects), and which is not.

7.4 PIS Countermeasures

The inner workings of PIS does not necessarily include any malicious behaviours, but rather benign behaviours such as showing content on the screen (advertisements) or sending non-critical information (visited Web sites) through the network. Vendors of countermeasures against PIS do not target "dangerous" behaviours, as is the case with for example anti-virus tools; making it harder to separate PIS from legitimate software. In addition, the only property that distinguishes PIS from legitimate software is the lack of user consent. Without proper techniques that safeguard user consent during the installation process, it is impossible for any PIS countermeasure to accurately decide on what software to target.

Current PIS countermeasures are based on centrally governed classifications of what software that should be regarded as spyware and which should not. This model provides a too coarse mechanism to accurately distinguish between the various types of PIS that exists since this relation is based on individual users' perspectives. That is, the degree of user consent needs to be regarded when distinguishing spyware from products that are beneficially tailored toward the users' needs. In the following subsections we present mechanisms that support users when making difficult trust decisions about whether to allow certain software to enter their system, or not. The goal is *not* to make these trust decisions for the users, but instead to develop mechanisms that support them. In the end it's up to the users to make the decision [13].

7.4.1 Software Deeds

In Section 7.2 we describe the problems that users face when trying to give an informed consent based on the corresponding EULA content. The purpose of using EULAs is to establish a juridical agreement between the software user and vendor, not to enlighten users about potential implications. However, since no additional standardized methods exists that aid users in comprehending software behaviour, users need to put their faith in third party reviews (e.g. from friends or magazines) [13].

Clearware.org try to increase software transparency towards users based on *software deeds*, which refine the EULA content and present it in a more human readable format [10]. These deeds exist at different comprehension levels and include pictograms (small pictures) that denote various behaviours of software, e.g. if the software display advertisements or if it includes fully functional removal routines. At the easiest level users are able to get a basic understanding about the behaviour of software they are about to install, by simply skimming through a set of pictograms.

In addition to these human readable deeds there also exists a corresponding machine readable deed in XML-format, making it possible for the operating system to automatically filter software in its perimeter based on local user preferences.

7.4.2 Software Preferences

By individually configuring software preferences, e.g. with respect to security, privacy, or performance, it is possible for users to define their own level of acceptance for new software. Exporting these preferences into a machine readable format, e.g. XML, allows the operating system to access and enforce them. This would result in that only specific software that matches the user's preferences is allowed to enter their system, e.g. with respect to pop-up advertisements. As soon as a user starts installing new software, including an XML-deed described in previous section, the values extracted from this deed are automatically matched against the user's preference list. If they match, the software is allowed to enter. Otherwise, the operating system would take some secondary action, e.g. notify the user about the mismatch or ask for permission to proceed anyway.

This technique is used by the *Platform for Privacy Preferences* (P3P) to allow Web sites to specify what information they gather from users when visiting their site [7, 12]. Users define their own local P3P preferences in their Web browser, which then automatically compares these against the remote preferences distributed by the requested Web server. As a result users' browsers only accept interaction with Web servers whose privacy-preferences correspond to their own local preferences [23].

7.4.3 Third Party Software Certification

Software *white-listing* is a technique that uses trusted third parties to certify acceptable software. One example is TRUSTe's *Trusted Download Program* which certifies "privacy-friendly" software [33]. A widespread service using such techniques provides market incentives for vendors of PIS to clearly and unavoidably communicate key functionalities of their software.

Problems concerned with third party certification include what software the third party regard as acceptable and which is not. A single organization that reach wide spread use with their certification will gain powerful influence in deciding what software to certify. Some sort of verification must also be carried out to check so that the software behaviour really corresponds to the certification requirements. Because of the vast amount of software that needs to be certified, some sort of automatic verification of software behaviour is probably needed which harden this approach.

The opposite of white-listing is *black-listing* where a trusted third party specifies software that is unacceptable. To some extent anti-malware tools, such as anti-virus software, function as a black-listing mechanism that identify and remove unacceptable software.

7.4.4 Collaborative Reputation Systems

Reputation systems include an algorithm which allows members of a community, such as eBay.com, to estimate other members' behaviour before an interaction. A collaborative reputation system presents an interesting method to address PIS, by collecting the experience from previous users' trust decisions regarding installation of software. While techniques such as third party certification or software deeds aim at increasing user awareness, reputation sys-

tems instead collect and refine user experiences. This experience is then used in a collaborate manner to inform (novel) users about the general opinion that exist for a specific software. The fundamental idea is that users make more accurate trust decision when incorporating such information, i.e. accompanying rumours from friends.

A first, modest, version of such a reputation system could include simple information about whether users decide to install certain software or not, i.e. if they choose "install" or "cancel". Based on this information, subsequent users are presented with statistics about previous users' installation decisions for a specific software. A more useful system needs to include evaluations of software that previously has been installed, e.g. when users decide to uninstall software they are asked to evaluate it by specifying a grade and a comment. This information is then processed by the reputation system, together with for how long the software were installed on the user's system. When subsequent users wish to install the same software, they are presented with the collective view on this software, i.e. what previous users' has thought about the software and for how long they decided to use it. Since such a system utilize sensitive information (e.g. what software users install) it is crucial that privacy and anonymity concerns are properly addressed.

Sandra Steinbrecher presents a design for a "privacy-respecting reputation system", which is based on continual changes between several user pseudonyms inside the reputation system [31]. This approach allows the system to protect user privacy, anonymity, and unlinkability between former actions in the system.

7.5 Discussion

Since user's informed consent distinguish legitimate software from spyware, it is important to safeguard and support users' authority to make informed decisions about software, a priori to the installation. Without such measures it is insuperable to correctly define and mitigate spyware, based on the software functionality, since this is depending on every single user's relation to specific software, e.g. what one user regard as a spyware another user see as a beneficial tool. This relation is impossible to capture at a later stage by any countermeasure tool, i.e. such tools are condemned to vast classification failures of spyware. From this follows that removal of malware "only" is a technical problem, while removal of spyware is

both a technical and juridical problem. In previous investigations we have observed that leading anti-spyware tools are quite inaccurate in their classification of spyware, rendering in notable false alarms and false accusations rates [5].

Current spyware countermeasures are reactive, i.e. are designed to remove known spyware. Protecting users' systems with such techniques often target the threats once already inside systems. In an attempt to improve this situation countermeasure vendors try to broadening their defences, e.g. by relocating into network routers and servers [23]. However, these countermeasures could not properly capture all software labelled as spyware by the users, and at the same time protect legitimate software, due to the individual nature of these classifications. In this paper, we present a number of mechanisms that aid users in their evaluation of software before actually installing them. Mechanisms that improve user awareness fundamentally change the classification of spyware, since fully transparent software never can be labelled as spyware, by definition. Introducing such mechanisms result in a transformation of Table 7.2 into Table 7.3, i.e. the middle row is removed. Any software that presents complete and correct information to the user during the installation is represented as one of the three legitimate software types on the top row. Exactly which one depends on their behaviour and consequences on the system, and the user. Users can with the help of local preferences differentiate between what software, on the top row of Table 7.3, they regard as acceptable and which they don't, i.e. there exist a possibility of individual adjustment for the user. On the other hand, software that does not play by the rules, by not presenting complete and correct information, are defined as one of the three malware types on the bottom row.

Any software not playing by the rules, in terms of properly announcing their intent prior to the installation, should rightfully be targeted and handled by anti-malware tools. This imply that anti-malware tools should not only target software that use exploitable system vulnerabilities to gain entrance to systems, but also software that deceive users about their business by using inferior user disclosures. From this follows that anti-malware mechanisms handles any software not subordinate to the rules of complete prior disclosure and consent from the users. Once this group of software has been excluded, users can depend on the information found in deeds, EULAs and other documentation to be correct. Any "dishonest" software slipping through the anti-malware tools' detection would impact a few initial systems. However, the affected users will surely

downgrade the responsible software via the reputation system, so that subsequent users are more restrictive. Over time the reputation will catch up on these questionable software vendors, forming a future deterrent effect.

	Negligible Negative Consequences	**Moderate Negative Consequences**	**Severe Negative Consequences**
High Consent	Legitimate software	Adverse software	Double agent
Low Consent	Covert software	Semi-parasites	Parasites

Table 7.3 Difference between legitimate software and malware with respect to user's informed consent and negative user consequences.

Software certification and deeds described in Section 7.4, further imply that users' uncertainty about what software that is installed on their system decrease significantly. This render in that any indirect negative consequences associated with unsolicited software are removed, i.e. software with exploitable vulnerabilities that execute on users systems without their awareness. Introducing preventive mechanisms against PIS offer the following three benefits towards the users:

1. A lowest level with regard to software behaviour, deed, and EULA correctness that is accepted for software in general.
2. Basis on which software behaviour and consequence can be evaluated prior to any installation, blocking unacceptable software before entering the system.
3. Possibility for users to define individual software preferences, which allow only a subset of all legitimate software to enter their system.

It is notable that all types of software that currently is targeted by traditional anti-spyware mechanisms are either removed by the introduction of the preventive mechanisms, or are fully covered by anti-malware tools. We therefore believe that, after an initial transitional period, all anti-spyware tools will be outmanoeuvred by, or integrated in, already existing anti-malware tools. These anti-malware tools will act as regulators that safeguard both users' systems from illegal infections, and indirectly protect the correctness of information about software, e.g. in EULAs.

7.6 Conclusions

Users need to know what they install, and learn how to distinguish between acceptable and intolerable software, a priori to any software installation. Everyone needs to be presented with complete, accurate and condensed information about the software's functionality during the installation process. We argue that additional mechanisms that safeguard user's informed consent are required to fight PIS effectively.

Our classification of PIS put emphasis on user consent, where high consent means legitimate software, medium consent means spyware and low consent means malware. To exclude spyware from legitimate software and malware, the classification emphasis on negligible, moderate or severe user consequences in an environment of either high, medium or low user consent.

As future work we will develop and evaluate a proof-of-concept PIS preventing reputation system, including a client for the Microsoft Windows environment.

7.7 References

[1] W. Ames, "Understanding Spyware: Risk and Response", in the *IEEE Computer Society*, Volume 6, Issue 5, 2004.

[2] Anti-Spyware Coalition, http://www.antispywarecoalition.org, 2006-07-18.

[3] AOL/NCSA Online Safety Study, http://www.staysafeonline.org, 2006-07-18.

[4] K. P. Arnett and M. B. Schmidt, "Busting the Ghost in the Machine", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[5] M. Boldt and B. Carlsson, "Analysing Countermeasures against Privacy-Invasive Software", in the *Proceedings of ICSEA'06*, Papeete Tahiti, 2006.

[6] J. Bruce, "Defining Rules for Acceptable Adware", in the *Proceedings of the Fifteenth Virus Bulletin Conference*, Dublin Ireland, 2005.

[7] S. Byers, L.F. Cranor, and D. Kormann, "Automated Analysis of P3P-Enabled Web Sites", in the *Proceedings of the Fifth International Conference on Electronic Commerce (ICEC2003)*, Pittsburgh USA, 2003.

[8] Center for Democracy &Technology, "Following the Money", http://www.cdt.org, 2006-07-18.

[9] E. Chien, "Techniques of Adware and Spyware", in the *Proceedings of the Fifteenth Virus Bulletin Conference*, Dublin Ireland, 2005.

[10] Clearware.org, http://www.clearware.org, 2006-07-18.

[11] L. F. Cranor, "Giving notice: why privacy policies and security breach notifications aren't enough", in *IEEE Communications Magazine*, Vol. 43, Issue 8, 2005.

[12] L. F. Cranor, "P3P: Making Privacy Policies More Useful", in the *IEEE Security & Privacy*, Volume 1, Issue 6, 2003.

[13] L. F. Cranor, *"Security and Usability"*, O'Reilly, Sebastopol, 2005.

[14] Earthlink Spy Audit, http://www.earthlink.net/spyaudit/press/, 2006-07-18.

[15] FreeBSD Ports, http://www.freebsd.org/ports/, 2006-07-18.

[16] B. Friedman, E. Felten, and L. I. Millett, "Informed Consent Online: A Conceptual Model and Design Principles", *CSE Technical Report*, University of Washington, 2000.

[17] N. Good, et al., "Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware", in the *Proceedings of the Symposium on Usable Privacy and Security,* Pittsburgh USA, 2005.

[18] S. Görling, "An Introduction to the Parasite Economy", in EICAR 2004, Luxemburg, 2004.

[19] 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.

[20] Lavasoft, http://www.lavasoftusa.com, 2006-07-18.

[21] D. M. Martin Jr, R. M. Smith, M. Brittain, I. F. Fetch, and H. Wu, "The privacy practices of Web browser extensions", in *Communications of the ACM*, Volume 44, Issue 2, 2001.

[22] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy, "A Crawler-based Study of Spyware on the Web", in the *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS 2006)*, San Diego CA, 2006.

[23] Proxy Appliances Blue Coat Systems, Inc., http://www.bluecoat.com, 2006-07-18.

[24] B. Schneier, "Inside risks: semantic network attacks", in *Communications of the ACM*, Volume 43, Issue 12, 2000.

[25] C. Shapiro and H. R. Varian, *"Information Rules - A Strategic Guide to the Network Economy"*, Harvard Business School Press, Boston Massachusetts, 1999.

[26] S. Shukla and F. F. Nah, "Web Browsing and Spyware Intrusion", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[27] J. C. Sipior, "A United States Perspective on the Ethical and Legal Issues of Spyware", in *Proceedings of Seventh International Conference on Electronic Commerce*, Xi'an China, 2005.

[28] E. Skoudis,*"Malware - Fighting Malicious Code"*, Prentice Hall PTR, Upper Saddle River NJ, 2004.

[29] Spyware Developers Net Huge Profits, Outrage - With Annual Revenues of $2 Billion, Pop-up Ads are a High-Stakes Game, http://www.msnbc.msn.com/id/13757388/, 2006-07-18.

[30] "Spyware": Research, Testing, Legislation, and Suits, http://www.benedelman.org/spyware/, 2006-07-18.

[31] S. Steinbrecher, "Design Options for Privacy-Respecting Reputation Systems within Centralised Internet Communities", in *Proceedings of the 21st IFIP SEC 2006*, Karlstad Sweden, 2006.

[32] StopBadware.org, http://www.stopbadware.org, 2006-07-18.

[33] TRUSTe - The Trusted Download Program (Beta), http://www.truste.org/trusteddownload.php, 2006-07-18.

[34] M. Warkentin, et. al, "A Framework For Spyware Assessment", in *Communications of the ACM*, Volume 48, Issue 8, 2005.

[35] Webroot Software, *"State of Spyware - Q3 2005*", http://www.webroot.com/resources/, 2006-07-18.