

Master Thesis
Computer Science
Thesis no: MCS-2007:20
September, 2007



Countering Privacy-Invasive Software (PIS) by End User License Agreement Analysis

Arvind Dathathri

Jules Lazare Atangana

Department of
Interaction and System Design
School of Engineering
Blekinge Institute of Technology
Box 520
SE - 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science.
The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Authors:

Arvind Dathathri

E-mail: arvind.rocky@gmail.com

Jules Lazare Atangana

E-mail: obalaz@gmail.com

University advisors:

Bengt Carlsson

Department of Interaction and System Design

Martin Boldt

Department of Interaction and System Design

Department of
Interaction and System Design
Blekinge Institute of Technology
Box 520
SE - 372 25 Ronneby
Sweden

Internet: www.bth.se/tek

Phone: +46 457 38 50 00

Fax: +46 457 102 45

Abstract

In our thesis we use a preventive approach to stop spyware from entering the system. We aim at increasing the user awareness about the background activities of the software. These activities are implicitly written in End User License Agreement (EULA). We are using a multi-layer user notification approach to increase the user awareness and help him make a good decision, which is in accordance with the European legal framework [3]. A proof of concept tool is developed that will use the user preferences to present the EULA in a compact and understandable form thereby helping the user in deciding with the installation of a software.

Keywords: Privacy, PIS, EULA.

Table of Contents

Chapter 1 Introduction	3
1.1 Background	4
1.2 Definitions	5
Chapter 2 Research	6
2.1 Research Motivation	6
2.2 Research Questions	6
2.3 Research Methodology	7
Chapter 3 Experiment	8
3.1 Experiment Design.....	8
3.2 Results	9
3.2.1 Kazaa	10
3.2.2 LimeWire	12
3.2.3 ABC Scrabble	13
3.2.4 Wengo	13
3.2.5 WinAntiVirus 2006	14
3.3 Analysis of the results	14
Chapter 4 Development of the Proof of Concept Tool	17
4.1 State-of-the-Art Regarding EULA Analyser Tools.....	17
4.2 Design of the Tool	18
4.2.1 EULA Analyser	18
4.2.2 User Software Preferences Editor	19
4.2.3 Software History	21
4.3 Design Choices.....	21
4.3.1 Black List of Words	21
4.3.2 Overcoming False Positive Warnings	22
Chapter 5 Assessment of the Tool	23

5.1 Results	23
5.2 Analysis.....	26
Chapter 6 Survey	28
6.1 Methodology	28
6.2 Percentages Collected From Individual Questions	29
Chapter 7 Discussion.....	32
7.1 Improvements to the Usage of EULA	34
7.1.1 EULA Framework	34
7.1.2 Using Multimedia in EULA	34
7.1.3 Machine Readable Deeds	35
7.1.4 Reputation System	36
7.1.5 Deeds at Several Abstraction Layers	36
7.2 Research Questions	36
Chapter 8 Conclusion.....	38
Chapter 9 Future Work.....	40
Bibliography	41
Appendix A Software GUI.....	43

Chapter 1

Introduction

A recent study has concluded that, Privacy-Invasive Software (PIS) has the potential to overthrow the positive aspects of belonging to a large community network [1]. In fact, the components that bundle software applications are involved in some suspicious background activities that reduce the computing power and increase network load. Worst, they are also responsible for the misuse of network users' personal information. Therefore, the issue of countering PIS should be addressed more seriously [1]. Today, many software applications come as freeware or shareware. In shareware software applications, users are allowed to use the software for a certain period of time, after which they have to purchase the software. In many freeware applications, users can use the software for free of charge but they will be viewing sponsored advertisements while running them. These sponsored ads will be delivered either in a small section of the software interface or in a separate pop-up window. Adware is the term used to describe this group of software applications¹. A specific category of PIS called spyware, tracks the user activities like his browsing habits, and deliver targeted ads. These spyware programs also transfer users' personal information to third-party servers.

Many users are not aware of the existence of spyware programs in their system, since these programs run in the background [2]. There are a number of anti-spyware tools that try to detect and remove these spyware programs. However, anti-spyware tools work much like anti-virus programs using a signature-based method. This requires anti-spyware vendors to explicitly classify a program as a spyware. The classification of software applications as adware or spyware is still ambiguous and

¹ Wikipedia – The Free Encyclopedia, “*Adware Article*”, <http://en.wikipedia.org/wiki/Adware/>

strongly influenced by legal issues between anti-spyware vendors and PIS developers. Software applications can be considered as legitimate as long as all their activities are explicitly notified to the user and require his consent [1]. These activities are often mentioned in the End User License Agreement (EULA) and require user approval for installing and using the software applications. Unfortunately, many users do not read the EULA before installation. They complain about the bad presentation format. In fact the EULA text is often very long, and contains a lot of terms that are difficult to understand [2].

Some EULA analysers are available that scan the EULA, capture and present parts of EULA text containing important and suspicious terms. The output of this operation reduces the number of lines to read, but remains in an unintelligible format. Users still have to read and understand legal terms and sentences pointed by these analysers. This does not help users to take a correct decision regarding the installation. Many users regret installing the software when they are informed later about all its malicious activities [2].

1.1 Background

Privacy is one of the major concerns raised by spyware, but the larger issues are transparency and control. Users are typically unaware that spyware programs are being installed on their computers and often hard to uninstall [4]. Tools that assist users in monitoring their computers are available. Anti-spyware and reputation system are two categories of tools that jointly address this issue very well. The problem of transparency is an important concern. Although all the software activities are notified in EULA, users still have limited understanding of EULA content. They show little desire in reading lengthy notices, and prefer short, concise notices. However, legal requirements require companies to provide complete notices that do not fit this standard. There is a serious concern over user notice and consent required during an installation process. The complexity of notices hampers users' ability to understand such agreements. One attempt to improve software vendors' notification ability about the behaviours of their product is the Platform for Privacy Preferences Project (P3P) [5]. Under this standard, websites' policies are expressed in a predefined grammar and vocabulary expression, a sort of standardisation of the EULA. Researchers from the Centre for Information Policy Leadership (CIPL), however, suggest a global solution to user

notification based on multi layer notice [18]. This thesis is concerned about increasing user awareness during the software installation by applying a multi layer notification.

1.2 Definitions

The following terms are used throughout our thesis and definitions will help in understanding them clearly.

Spyware: Software programs that collect and transmit users' information without the consent from the user.

Adware: Software programs, which display advertisements when using the application.

Malware: Any binary executable or malicious code that damages the users' systems [10].

PIS: Any software program that invades users' privacy is called privacy-invasive software and this includes spyware and adware programs [1].

EULA: End User License Agreement, which is a legal contract between the software vendor and the user. It grants a particular license to user and specifies the perimeters of the permission granted.

ToS: Terms of Service are the rules to which a user has to agree to use the service provided by the software vendor. It contains many legal terms and informs the user how he can use their service.

Privacy Policy: A declaration made by a software vendor regarding the privacy of users. It specifies what will be done with the information provided by users and how they are going to use it. For example, when a user signs up by giving his name, address and other details to use a particular service from a software vendor, the privacy policy will state if any third party is going to use this information.

Chapter 2

Research

2.1 Research Motivation

PIS are prevalent problems in our computer society. Users involuntarily contribute to the proliferation of PIS. Unlike worms and viruses that spread without the user consent, PIS require user approval. Important software behaviours are described in the EULA. Unfortunately, users have limited understanding about the content and terms specified in EULA, and little desire to read lengthy notice. Users often regret their installation decision, when they are informed of the actual contents of the EULA they agreed on. There is a great need in helping users to take a good decision concerning the installation of a software application.

2.2 Research Questions

The overall thesis focuses on helping users to stop PIS from entering their system, by aiding them in making good decisions regarding software installation. To achieve this objective, we are addressing the following research questions:

- How are the behaviours of existing software programs notified to users through EULA?
- How can the usage of EULA be improved by new mechanisms?

- How well could the activities of the software, contained in EULA, be provided to users in an effective way?

2.3 Research Methodology

We are using a mixed research methodology in our thesis [7]. A literature study is made to know the existing problems and causes for the proliferation of PIS. We will carry out an experiment with five popular software applications to investigate their activities and to know how these activities are notified in their respective EULA. From this experiment, we will draw conclusions that allow us to develop a proof-of-concept tool based on user preferences to stop PIS from entering the system. The tool integrates several modules: the analyser, the software preferences editor, and the software preferences history. The analyser scans the EULA text supplied by users, searching for important and suspicious terms. We present the result of the analysis in an easy to read window format known as software deeds. Software preference editor is a window interface where users select their security preferences about software to be installed. Selected options are mapped to produce corresponding security levels. In addition, every software application installed is stored along with its security options for history purpose. Finally, we will carry a survey to test the impact our tool might receive from users, and get new directions for further improvements.

Chapter 3

Experiment

The present experiment investigates how the behaviours of software applications are described to users through EULA. Prior to installation of a software application, the user is required to give his consent regarding the conditions and functionalities related to the software. This experiment focuses on bad behaviours that is, any behaviour that intrudes users privacy. It aims at emphasizing the existence of additive components that bundle software applications. It also questions the notification of background activities of these additives components to users through the EULA.

3.1 Experiment Design

The experiment was carried out in the security laboratory at Blekinge Institute of Technology, Sweden. Blekinge security laboratory was suitable for us since, it provided a cloning system and thereby making it easy to install the operating system. All the computers used were of the same configuration and had Windows XP Professional Edition as their operating system. We selected the five most popular software applications, available at download.com on October 5th 2006², namely: Kazaa, Limewire, WinAntiVirus, Wengo and ABC Scrabble.

² Download.com - *Free Software Downloads and Software Reviews* <http://www.download.com/>

These applications were scanned using Ad-aware SE Personal 1.06, a recent and effective anti-spyware tool³. Each software application was installed and analysed on a separate system. Default options recommended by the software applications were accepted and there was no anti-spyware tool present during the installation. Finally, we analysed the EULA of every software application previously installed and scanned. We used a free version of EULalyzer 2.1, a commercial EULA analyser tool. The main goal of the experiment was to identify some bundled components, and to check if and how the suspicious activities they generate are reported in their EULA. However, the limitation of this experiment was the absence of Internet connection, which means, we could have missed some components that are triggered by Internet connection.

3.2 Results

We detail the experiment results in the present section. We individually describe additive components detected by the anti spyware tool. Table 3.1 summarizes our finding about bad behaviours notification. We present the results found from the experiment goals perspective in Fig. 3.1.

Table 3.1 Bad behaviours notification status

Bad behaviour overview	Kazaa	LimeWire	ABCscrabble	Wengo	Win Antivirus
Display advertisement	R	NR	R		NR
Connect to a remote server in Internet	R	NR	NR		R
Download third party software without notice	R	NR	R		
Unable to remove all the files installed	R	NR	NR		
Change the system configuration	NR	NR	NR		NR
Redirect to third party websites	R	NR	R		
Collect user information	R	R	R		
Tracking or monitoring user activity	NR	NR	R		
Display unethical contents	NR	NR	NR		NR

³ Ad-Aware @ Lavasoft, <http://www.lavasoftusa.com/>

Table 3.1 presents an overview of bad behaviours found during the scanning phase. Some of these bad behaviours are explicitly notified to users in EULA. We use characters R and NR to denote the notification status: reported (R), non-reported (NR) in the EULA. The blank cells in the table refer to behaviours that are not present; the application concerned is clean from that particular behaviour. Fig 3.1 summarizes and clarifies the results.

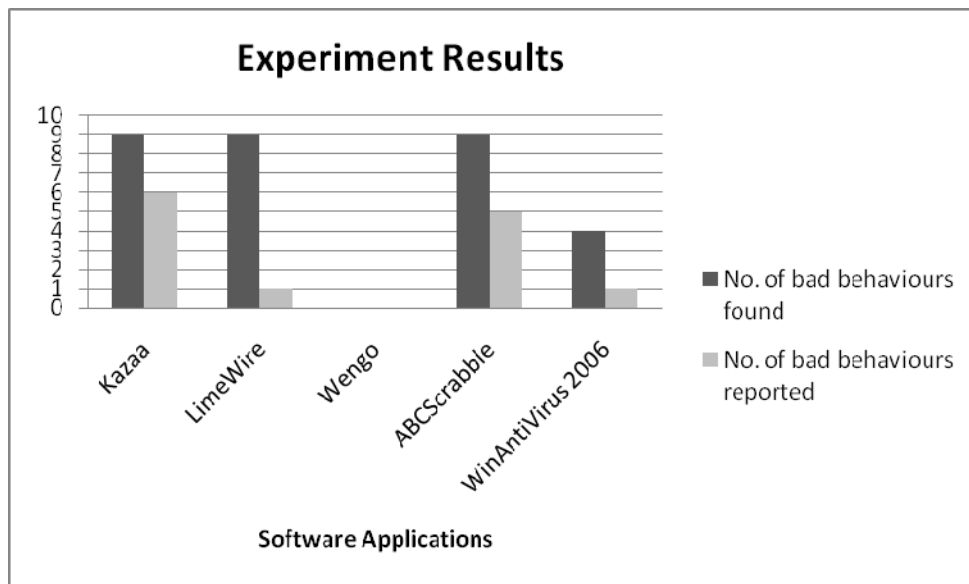


Fig 3.1 Experiment results

Additive components that are responsible for bad behaviours found are described below.

3.2.1 Kazaa

Kazaa is a popular P2P file sharing application, which lets users search for, download and share files using FastTrack Protocol [9]. Most of the files that are exchanged with this application are mp3 and video files even though any kind of files can be shared. The official Kazaa client is available as freeware, but it is ad-supported. A list of Adware/Spyware programs detected after installing the Kazaa P2P client is described below.

- 1) Adware.P2PNetworking is a content-distribution system based on P2P principles. It uses system resources and bandwidth for distribution of contents as ads, music, commercials and so on. Files associated with this suspicious activity are P2P Networkin.exe and Marshal.dll. It also changes the registry values in order to run at the system start-up⁴.
- 2) Adware.Topsearch acts as a search engine. It will supply advertising content to Kazaa users. This will also be installed along with other software programs like Grokster. Files associated with this activity are Topsearch.dll, asm.exe, asmeps.dll, and Manager.exe. More over this adware runs at system start-up and even opens unsolicited websites and pop up windows⁵.
- 3) AltnetBDE is another spyware component installed along with the Kazaa client and is published by Brilliant Digital Corporation. It displays animated advertisements to users. This can be remotely turned on and when done, it becomes a part of a network that Brilliant Digital controls. All these actions will take place without user's awareness and will transmit usage statistics and personal identifiable information⁶. Although the component can be installed separately, Kazaa will install it along with its P2P client.
- 4) Adware.TopSearch.B is also an adware that acts as a search engine. It can also be installed via an ActiveX downloader. A known file, Topsearch.dll is associated with this adware component.
- 5) RXToolbar represents another spyware that tracks the web sites visited by users and sends the keywords that are searched on common search engines like Google and Yahoo to a remote server⁷. Two files: RXToolbar.exe and RXToolbar.dll are associated with this adware and is published by www.searchenginebar.com
- 6) Adware.InstaFinder is a Browser Helper Object (BHO), which is a plug-in written for web browsers like Internet Explorer (IE) [10]. It will display advertisements and will download and execute other adware

⁴ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2004-100815-2121-99.

⁵ Lavasoft.com, "Spyware Education Centre",
http://www.lavasoftnews.com/ms/display_main.php?tac=TopSearch.

⁶ Lavasoft.com, "Spyware Education Centre",
http://www.lavasoftnews.com/ms/display_main.php?tac=AltnetBDE.

⁷ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2005-040716-3445-99.

programs⁸. It will also redirect the searches to a pre-determined web site. Files associated with this adware are instafink.dll, instafin.dll, instafinderk_inst.exe and instafinder_inst.exe. Adware.InstaFinder is published by www.instafinder.com.

7) Adware.MWSearch is a search toolbar for Internet Explorer and will display advertisements. It modifies the registry keys in such a way that the toolbar gets loaded as an Internet Explorer search toolbar. Also if this search toolbar is used, the adware will send the searched string to morwillsearch.com domain. Such activity is also considered as a spyware⁹. A single file iacad.dll is associated with this adware.

After uninstalling Kazaa, the components P2PNetworking and AltnetBDE were not removed from the system.

3.2.2 LimeWire 4.12

This is another P2P client like Kazaa used for sharing files over the Internet. A list of malware programs identified is detailed below.

1) Adware.ISTbar is a toolbar from Integrated Search Technologies, which will add a toolbar to the Internet Explorer. This will also hijack the searches done by users and will change the home page in IE. Many pop-ups will also be displayed, and the registry is modified so that this toolbar will be a default search bar for IE. This suspicious behaviour is associated with the file ISTbar.exe and will be present even after uninstalling the game. So there is a potential threat that may install this toolbar again¹⁰.

2) Webhancer is associated with the files whAgent.exe, whiqlpr.dll, whieshm.dll and whSurvey.exe. These files are added to system start-up folder, hence causing the web browser instability and auto-updates without the permission of users. It tracks both the system performance and users' online habits and reports this information to a remote server. Webhancer's publisher is Webhancer corp.¹¹.

⁸ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2005-040516-0442-99.

⁹ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2005-022414-4111-99.

¹⁰ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2003-091913-2632-99.

¹¹ Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2003-080814-0724-99.

3) Adware.MWSearch is another toolbar for IE that will be installed without the notice of users. We can notice that the same toolbar was also installed along with Kazaa client⁶.

4) Adware.DollarRevenue is also a component installed along with LimeWire. But, it is not removed from the system even after the main program is uninstalled. It displays ads and downloads other adware programs that can be remotely turned on and controlled¹².

The Webhancer components were added to system start-up so that they start automatically once the system boots. The file webh.dll present in the C:\Windows directory was not removed after uninstallation.

3.2.3 ABC Scrabble

This is a game of scrabble from 2MGames.com. It installs additional components that act as spyware programs without any information given to the user. Even after completing the installation of this game, many additional components are continuously installed and automatically added to the user's start-up folder so that they run when system starts. It also fails to uninstall the bundled components when the main game is uninstalled. Here is a list of components that were identified as malware.

1) Webhancer: This trackware was explained in the previous section 3.2.2.2.

2) Adware.ISTbar is a toolbar from Integrated Search Technologies, which was explained in the previous section 3.2.2.1.

After uninstalling the original software, the bundled webhancer and ISTbar will not be removed from the system. So users should manually uninstall these added components from Add/Remove Programs in the control panel.

3.2.4 Wengo 0.99

Wengo is a popular Voice over IP (VoIP) provider that can be used to do voice chat and call landlines and mobile phones all over the world at cheap rates when compared to other popular VoIP providers like Skype

¹² Symantec.com, "Security Responses",
http://www.symantec.com/security_response/writeup.jsp?docid=2005-101413-2726-99.

and Yahoo¹³. Since this software is more popular, we were curious to know if it contained any spyware and adware. After scanning, we did not find any malware programs associated with this VoIP tool. Also this will not install any other component and all the files will be completely removed upon uninstallation.

3.2.5 WinAntiVirus 2006

As the name suggests, this is an anti-virus tool for Windows machines. But it is considered as a badware since it will encourage the user to purchase the full version in an unethical manner. This is done by, displaying to the user that, his system has many threats like spyware and adware, even when the system will be clean [11]. It also runs in the background and will install some updates without the permission of the user. Since it will be running in the background, pop-up ads will be displayed asking the user to purchase the full version in order to clean his system and protect his privacy. We can say that this program will give a false threat to the user and will persuade him to buy their product.

When the windows process table is checked, the process FWSvc.exe and WinAV will be continuously running in the background. Also this process is added to the system start-up registry so that it starts automatically when system starts. This kind of behaviour is not presented to users. An exclusive way to stop WinAV process is to uncheck the identified process in the msconfig toolbar, where different processes that start automatically are listed.

3.3 Analysis of the Results

Following observations arise from the current experiment:

- A great many software applications are bundled with additive components that serve suspicious background activities. These background activities evade users' privacy and expose system stability and security.
- Behaviours of software applications are implicitly notified to users through the EULA. In fact EULA texts are full of legal terms and references to third partner policy websites. In addition, EULA texts are so

¹³ Wengo Phone, <http://www.wengophone.com/index.php/homePage>

long that users feel discouraged to go through. Worst, vendors' statements are not negotiable; users are required to agree on giving away certain rights if they are to install a desired software application.

- EULA analyser tools scan the EULA text and report suspicious key terms, relieving users from the big effort required to read lengthy EULA. But a single term seems insufficient to describe a behaviour. Users are required to read the lines containing flagged terms in order to understand the behaviour referenced.

- Background activities revealed by anti-spyware tools are not clearly reported in EULA texts. This could explain why in Fig. 3.1, the number of bad behaviours found is different from the number of bad behaviours notified to users. Both message notification and performance of EULA analyser tools are questioned here. In fact, the usage of truncated words or non-common words and expressions in EULA text could negatively affect the performance of EULA analyser tools. Vendors can voluntarily truncate important key terms to comply with the law while deceiving current EULA analyser tools. Next sections investigate the issue very well.

- EULA text seems vendor proprietary. Vendors decide both on the content and the presentation of notification messages. No template reference structure is in use. Blank cells in Table 3.1 refer to behaviours not detected by anti-spyware and not mentioned in EULA text. Such a status is regarded as unreliable from a customer point of view. Vendors should also state clearly bad behaviours that their software product does not allow. To provide sufficient guaranty to users, mainly an equal understanding of any EULA, a current trend actively requests the standardisation of EULA. We examine and better describe that approach in the discussion part.

- In general, the law effectively addresses deceptive behaviours of software products; the activity is considered as illegal and is ranged in the category of unfair trade practices. Governments' trade committee require vendors of software applications to fully notify users about all the behaviours of their products [13]. Notices should be readable and understandable. Many users forum do help in protecting the users' privacy during trade and exchange with software vendors. They collect users complaints, proceed to forensic inquiries and with the support of all the members, they present the case to the Federal Trade Commission (FTC). Severe legal sanctions are taken against any vendor that deceives users. Such cases are treated as crime and therefore, addressed by the criminal law. Resulting penalties are ranged up to five years in prison, and

suits by Internet Service Providers (ISP) that could lead to a ban of the vendor and its product¹⁴ [13]. Current trend in trade and exchange of software focuses more on the presentation of notification messages to users.

¹⁴ <http://www.stopbadware.org/> , “FTC shut down Team Taylor Made”

Chapter 4

Development of the Proof of Concept Tool

As stated before, the purpose of this thesis is to design an effective software preference tool that integrates the following functionalities: EULA Analyser, Software Preference Editor and Software Deed. This chapter presents some important considerations that should be addressed during the development of the tool.

4.1 State-of-the-Art Regarding EULA Analyser Tools

To lessen the difficulty for users in reading lengthy EULA, EULA analyser tools were designed. These tools scan the EULA text of a software application searching for specific suspicious key terms and expressions. EULA analyser tools output a list of suspicious key words found in the text and the corresponding sentences they refer to. Instead of reading the whole EULA text, users are recommended to read and understand the flagged lines of text displayed only, in order to take a good decision. By reducing importantly the size of the text to be read, the tools address the complaint of users regarding lengthy EULA. Two examples of tools freely available are EULalyzer¹⁵ and online EULA analyser¹⁶. Commercial versions of the above mentioned EULA analyser tools do exist also. They offer more features as: automatic analysis of the EULA text triggered by the installation process, better performance of the scanner conditioned by a large set of suspicious key words and

¹⁵ <http://www.javacoolsoftware.com/eulalyzer.html>

¹⁶ <http://www.spywareguide.com/analyze/index.php>

expression. Performances between different EULA analyser tools lay on semantic considerations, and suspicious key words set size. However, users still have a lot of personal effort to put in order to take a good decision regarding the installation of software applications. In fact, the understanding of legal terms referenced in flagged sentences is not common to all users; the risk of misinterpretation is high. Therefore, the risk of taking a bad decision even after reading the EULA remains high. There is a room for improvement here, mostly in the design of a more knowledgeable EULA analyser tool. Knowledge enables consumers to shorten the time needed to make decisions and reduce the cognitive effort to perform the tasks [14].

4.2 Design of the Tool

4.2.1 EULA Analyser

This module features the consumers' knowledge and control over the software applications to be installed. EULA analyser module is a core component of the tool that provides inputs to subsequent modules. A well-designed EULA analyser module is a guarantee for the efficiency of our tool. EULA analyser tools behave as text analyser tools. They scan the text looking for compound words and expressions. But text analyser tools go deeper in semantic considerations to get the meaning of a word in a sentence. They also get advantage from a large compound words set. As mentioned before, the performance of a EULA analyser depends also on the size of the suspicious keys words set.

We are concerned about developing an effective, knowledgeable and convivial tool. One approach is to get the knowledge from the scanned EULA text. Instead of referring users to multiple lines of text containing the suspicious key word detected, the tool gets the knowledge from these lines and just display the bad behaviours mentioned in the first layer. Our tool also takes the user software preferences into account. Suspicious behaviours that do not comply with users preferences are notified to the user in the second layer. We are therefore considering a multi-layer presentation of the analysis of the EULA text: the first layer reports all the bad functionalities detected in the EULA irrespective of the user preferences. Suspicious behaviours mentioned in the first layer are checked in regard to users' preferences. More details about the behaviours

that do not comply with the users preferences are presented in the second layer. The two layers representations are illustrated in Appendix A.

4.2.2 User Software Preferences Editor

Fighting against PIS potentially results in legal conflicts between users' interests and vendors' interests. What is complex is that the users' interests are never clearly defined and are very dynamical.

To give users more control over the installation of software in their system, the user software preference module has been developed at the application level. This module provides an interface where the user can select and save his software preferences. It presents in a simple and understandable language, an exhaustive list of bad behaviours that can be found in many software applications. Users are required to select the behaviours they strongly reject. The user preference module interface is illustrated in Appendix A.

The selection of the user is mapped on a scale that presents the security level (High, Medium and Low). Suspicious behaviours reported in the EULA are checked against the user preferences. Behaviours that do not comply with the user preferences are explicitly notified to the user. Such notifications are easily understood by users and effectively help them in taking a good decision about the installation of software applications.

User can statically change the security preferences; such action affects the future software applications to be installed. Table 4.1 below presents the available software preferences with their corresponding weight.

Table 4.1: Available software preferences and their corresponding weight.

Software Preference	Weight (W_i)	Information
Spyware	5	Highly dangerous behaviour that affects user privacy
Adware	4	Provides targeted ads
Third party components	4	Install components that are not trusted
Upgrade without notice	3	Dangerous, may exploit user privileges to upgrade itself
Objectionable objects	2	Not highly dangerous but unethical
Age limit	1	Like parental control
Benchmark testing and reverse engineering	1	No benchmark test and reverse engineering without vendor

		consent
Warranty	1	No warranty if the software damages the system

4.2.2.1 Grading of User Preferences and Security Levels

To provide a feedback to users regarding their selection of software preferences, the grading system is designed. It helps in assigning a security level that determines how safe their system will be. After selecting his software preferences, the user is alerted about the level of security of his system. Three levels of security are used: High, Medium and Low that are displayed on a coloured scale (See Appendix A).

Every software preference is given a weight ranging from 1 to 5, with 5 representing highly dangerous behaviour (see Table 4.1). User selected preferences are added to get the final score that determines the security level of his system as shown in Equation 4.1 below.

$$\text{Sum} = \sum_{i=1}^8 w_i * x_i \quad \text{Equation 4.1}$$

Where $x_i = 1$ if user has selected i^{th} preference
 $x_i = 0$ otherwise.
 $w_i =$ weight of the i^{th} preference.
Sum = total score.

If Sum \in [0 10], then security level is Low
If Sum \in [11 16], then security level is Medium
If Sum \in [17 21], then security level is High

Three main aspects that affect user privacy are spyware, adware, and installation of third party components. So these are given more weight in user preference module. Here is a scenario: a user selects all the preferences except spyware. The total weight will be 16, which means security level of his system will be set to Medium. Such a system is still exposed to spyware threat. A perfectly safe system should address all the potential threats. In this case, total weight will be 21 thereby setting the security level to High.

4.2.3 Software History

During the installation of a software application, users agree to have certain behaviour in the software. We find it interesting for users to keep track on the software applications they installed along with their security levels and preferences. So, we designed a software history module that stores all the installation decisions the user agrees on. It displays a list of software installed and the corresponding level of security of the system at the moment of their installation. By selecting a software from the history list, more details about the installation decisions are displayed, specifically whether the software complied or not with the security preferences stored at that time.

4.3 Design Choices

We follow the general design outlined in previous sections during the development of the tool. We present here some additional settings required in building the system.

4.3.1 Black List of Words

Analysing a EULA text is more of a semantic task than a syntactic one. Software vendors can cleverly convey the behaviours of their product in different manner. For instance, a vendor can use several sentence styles to notify that its product contains spyware. The EULA text may contain a simple sentence as “This product contains spyware“. Any EULA analyser will search for the black listed word spyware and may warn the user. A more cleverly written EULA may contain “This product may track your activities“, or “This product may monitor your activities“ and therefore can stay unnoticed by EULA analyser. This confirms that the performance of the EULA analyser is partially dependent on the size of the black list of words. The larger the list, the more suspicious words are likely to be detected and conversly, the percentage of false positive increases. Commercial versions of EULA analyser tools propose online updates of the black list. We build the black list for our tool by reading and analysing the bad EULAs published and available in some discussion forums and online communities¹⁷.

¹⁷ <http://www.stopbadware.org/>

4.3.2 Overcoming False Positive Warnings

The drawback of searching for specific key words is that we may get some false positive warnings. A vendor can disclaim by writing in the EULA text "This product does not contain any spyware". EULA analysers will scan for the black listed word spyware and issue a warning to the user without understanding the actual meaning of the sentence. We consider this as a false positive. It misleads the user in making a good decision regarding the installation of the software. One approach in overcoming the false positive warnings is to scan for negative words as "No, Not, Never, does not ...", in the neighbourhood of the black listed terms detected.

Chapter 5

Assessment of the Tool

In this chapter we are testing the tool designed on this study, and we are comparing the results with other tools freely available on Internet. We focus on the EULA analyser module since it is the core module of the tool. Apart from LimeWire, which did not present its EULA, all the remaining software applications used in Experiment were scanned with EULalyser 1.2, Online Analyser and our tool. We present and analyse the results obtained in following sub-sections.

5.1 Results

The output of the first notification layer of our prototype is added to table 5.1 to 5.5 for a slight comparison. User security preferences are set to their default values. Kazaa presented three different EULAs during the installation and are correspondingly named Kazaa1, Kazaa2 and Kazaa3 as seen in the following tables.

Table 5.1: Comparative result for Kazaa1

Terms found in kazaal	Eulalyser 1.2	Online Analyser	Prototype
Advertising	19	15 (Reference to advertising)	The software displays advertisement pop-up
Promotional message	3	3 (online promotion)	
Third party software	33	0	
Without notice	1	0	The software installs third party components
Privacy Search terms	1	0	

Web links	6	0	The software tracks your activity The software delivers questionable or objectionable object
Reference to tracking and monitoring	0	6	
Objectionable objects	0	1	

Table 5.2: Comparative result for Kazaa2

Terms found in kazaa2	Eulalyser 1.2	Online Analyser	Prototype
Advertising	1	1 (Reference to advertising)	The software displays advertisement
Without notice	1	0	
No warranty	0	1	The software downloads additional component without notice to the user Do not reverse engineer

Table 5.3: Comparative result for Kazaa3

Terms found in kazaa3	Eulalyser 1.2	Online Analyser	Prototype
Advertising	0	2 (Reference to advertisement)	The software downloads, installs or upgrade without any notice to the user
Third party	4	0	
Reference to tracking and monitoring	0	2	
Without notice	2	0	
Privacy Search terms	1	0	The software installs third party component
Address (Web links)	1	0	
No warranty	0	1	The software
Website visit	0	1	

Performance of software	0	1	displays advertisement pop-up The software delivers questionable or objectionable object
--------------------------------	---	---	---

Table 5.4: Comparative result for Wengo

Terms found in Wengo	Eulalyser 1.2	Online Analyser	Prototype
Third party	1	0	The software has no warranty
Reference to tracking and monitoring	0	1	The software downloads, installs or upgrade without any notice to the user The software installs third party component

Table 5.5: Comparative result for ABC Scrabble

Terms found in ABC	Eulalyser 1.2	Online Analyser	Prototype
Third party	2	0	This software links to a third party website
Website address	4	0	
Reference to tracking and monitoring	0	3	This software contains spyware
Tracking: Reference to tracking or monitoring of usage.	0	1	This software has no warranty

Table 5.6: Comparative result for WinAntiVirus

Terms found in Win AntiVirus	Eulalyser 1.2	Online Analyser	Prototype
Reference to tracking and monitoring	(Nothing detected)	3	<p>The software forbids any reverse engineering</p> <p>The software has no warranty</p> <p>The software downloads, installs or upgrades without any notice to the user</p>

5.2 Analysis

1) It is obvious that different tools use different black listed words. For example the term ‘third party’ is black listed in the EULalyzer 1.2 but not in the online analyser. As another example, the terms ‘ad’, ‘advertising’, ‘advertisements’, ‘ads’, ‘advertisers’ refer to the same behaviour. EULalyzer will scan for all of these key words whereas online analyser will scan for only some of them. Hence there is a difference in number of time the advertisement behaviour is referred.

2) While others tools report the number of appearance of some key words, our tool makes no mention of the quantity of the bad words detected. The output of our tool is more abstract at this first layer but sufficient to explicitly notify and inform the user about the behaviour of the software. In addition, some analyser tools present the level of complexity regarding the readability of the EULA. Such information is scientifically impressive but not helpful to the user installing the software. We present some screen shot of the output of the tools in Appendix A.

3) Result presented by our tool is more concise, easy to read, and likely to provoke a quick and good decision from the user. Other tools do not achieve this goal. For example in Table 5.1, the output of the tool EULalyzer refers the user to 63 short sentences when analysing Kazaa1; the output of the online analyser tool refers the user to 25 lines whereas our tool refers the user to three short informative lines.

4) The output of our tool does not contradict the others tools, better it presents their result in a more readable and meaningful format that helps the user in making a good decision.

Chapter 6

Survey

The present survey is conducted to test the impact the tool may receive from potential users. This survey does not intend to compare our tool with other available tools. Further, directions of improvements are revealed and presented in the next chapter.

6.1 Methodology

We designed a web page for the thesis¹⁸. The link to the thesis web page was sent to a sample of targeted users through a standard email explaining the purpose of the survey. A link to download the software was also inserted along with a sample of EULA texts. Nine questions covering the following areas: personal experience about software installation, opinion and interaction with EULA, opinion and interaction with the tool were explicitly written and made available online¹⁹. Users were firstly required to download and install the tool on their local personal computer compatible with .NET framework. Secondly, they were invited to run the tool and analyse few EULA texts provided. Finally, they could fill the survey online, free from any pressure.

Participants in this survey are university students, those are experienced computer users frequently involved in software applications download and installation. A total number of 34 users were selected in regard to their availability during the survey period that is week 34 of the year 2007. Percentages collected from individual questions are reported in next section.

¹⁸ <http://www.student.bth.se/~arda05/>

¹⁹ <http://freeonlinesurveys.com/rendersurvey.asp?sid=i2dks5mc01k2laz330136>

6.2 Percentages Collected From Individual Questions

Questions were asked on the order provided below and users could check on one answer he agrees on among the multiple presented. Question 9 requires a complete detailed answer from users.

- 1) **Is our tool helpful in making a decision in the installation of a software**

	Percentage	Responses	
Very helpful	41.2	14	
Helpful	55.9	19	
Not at All	2.9	1	
Total responses			34

- 2) **Is the Graphical User Interface (GUI) simple to use**

	Percentage	Responses	
Yes	38.2	13	
Yes, But it can be improved	58.8	20	
Not at All	2.9	1	
Total responses			34

- 3) **What is your expertise on PIS before you used this tool ?**

	Percentage	Responses	
Expert	8.8	3	
Pretty good	29.4	10	
Little	41.2	14	
No idea	20.6	7	
Total responses			34

- 4) **What is your expertise on EULA before you used this tool ?**

	Percentage	Responses	
Expert	11.8	4	
Pretty good	26.5	9	
Little	52.9	18	
No Idea	8.8	3	
Total responses			34

- 5) **Do you read EULA, ToS and other agreements prior to the installation of any software ?**

	Percentage	Responses	
Always	8.8	3	
Sometimes	41.2	14	
Not at All	50.0	17	
Total responses			34

- 6) **How much time do you spend on reading one agreement in general ?**

	Percentage	Responses	
None	41.2	14	
1 minute	26.5	9	
5 minutes	29.4	10	
10 minutes	2.9	1	
More than 10 minutes	0.0	0	
Total responses			34

- 7) **Are you able to understand all the terms and conditions in an EULA, ToS or any other agreement ?**

	Percentage	Responses	
Yes, Very well	26.5	9	
Some terms and conditions are not understandable	41.2	14	
Not at all	32.4	11	
Total responses			34

- 8) **What is your overall rating of the tool ?**

	Percentage	Responses	
5 (Very good)	32.4	11	
4 (Good)	44.1	15	
3 (Satisfactory)	23.5	8	
2 (Not good)	0.0	0	
1 (Very bad)	0.0	0	
Total responses			34

9) Your suggestion for future work on the tool

(The last five responses are given)

- You can also rate a software according to its features. So when I see the rating, i can decide if it is good or bad

- To include more information about software behaviour

- To make modification in displaying results.

- Hi, Good work. Suggestion would be to work in analysis area and GUI

- Why don't it run in taskbar and when i want to install a new software, it has to pop up and ask me. this will be a nice feature

Chapter 7

Discussion

The experiment results reveal that important software behaviours are reported in the EULA, they are written in a very formal and legal language that is hard for ordinary people to understand: 25 out of 34 users do not understand the EULA. Generally, the EULA text will be very large, around 5000 words and will contain many legal terms that are difficult to understand. To support this point, we present the EULA of wengo software in Appendix B. Even though this EULA is the shortest among the selected software applications, it has many pages. Users lack the patience to read such lengthy agreements before using the application: 17 out of 34 users do not read the EULA at all, 14 out of 34 users read EULA sometimes. We can also note that 23 out of 34 users take less than a minute to agree on EULA terms which is highly insufficient to understand the legal terms presented. Many software applications have more than one license agreement like EULA, Terms of Service (ToS) and privacy policy. A good example of this is Kazaa, which had three different license agreements.

Software vendors do not take any responsibility regarding third-party component bundling within their product. Instead, they propose a separate EULA or refer users to their partner privacy policy. Such practices are legally acceptable due to the license considerations, but they contribute in increasing or compacting the EULA text making it more confusing to users and EULA analyser tools. Analyser tools seldom scan the several EULA texts on one run. We recommend that software vendors adopt explicit privacy policy strategies. Software vendors, with strong concern about user privacy policy should associate with third parties that have the same privacy policy strategy. In addition, some names of software applications are misleading like WinAntiVirus 2006. It performs well as anti-spyware tool but carries some unethical background activities.

It falsely warns about threats in a clean system in order to force the user to purchase a commercial version of the product. Another behaviour that could lead to wrong decision is the absence of any license agreements during the installation. We noted this with LimeWire. It did not present any license agreement to agree on during the installation. It does not mean that the software is cleared from any PIS suspicion. Instead, the vendor does not let users know about the software. Such business practice is considered as unethical because it fouls the customers. Therefore, such cases should be severely addressed by the law.

A recent study shows that interactive multimedia applications can be achieved by monitoring real time audio input from television sets [15]. This method can also be used in delivering targeted advertisements, where user's television sets can be monitored to know what kind of programs the user is interested in. Based on this information, targeted ads can be delivered. It is obvious that PIS could explore new channels nowadays in order to get targeted users. The voice could also be a great target since it may carry users needs, feelings and desires. This was one of the reasons to investigate a VoIP tool. Wengo application was used to check if it contained any PIS. We found out that Wengo did not contain any PIS and also provided complete uninstallation feature. It could also mean that new investigation strategies are required to address the PIS problem nowadays since, there is an arms race between PIS developers and anti-spyware developers. It is very important that software vendors conform to the law about trade and exchange that abstractly define both consumers and vendors rights and duties.

Conversely, some software applications lay in the gray zone and cannot be branded as spyware or adware since their behaviours are reported in EULA. Anti-spyware tools are inappropriate to address those applications; they do not have legal right to record them in their black list. Our tool seems to effectively address such issue: 33 out of 34 users find our tool helpful, 14 find it very helpful. We avoid legal conflicts that may arise from any classification by focusing on message notification instead.

All the users were satisfied by our tool: 11 users rated it as very good, 15 users rated it as good and remaining 8 rated it as satisfactory. Still the tool could be improved. Impressive comments and suggestions were made by users. We summarize and present them as follows.

- a. Improvement in GUI area and in particular use of pictogram and multimedia in the result window.
- b. Standardize notification of software behaviours in EULA.

- c. Using several abstraction layers while displaying the result.
- d. Automatic software analysis without user initiation.
- e. Community networks where users rate a software according to its behaviour and functionality.

We detail and discuss each of these in the next section.

7.1 Improvements to the Usage of EULA

From this study we can draw a conclusion that there is a huge scope for improvement in the usage of EULA. Software vendors can state more clearly the behaviours of their application and these notifications can also be displayed to users in a proper way. In this section, we state some important improvements for the EULA.

7.1.1 EULA Framework

Notification about ethical information collection should be adequate and transparent. EULAs are the most appropriate way to achieve transparency and obtain users consent. Users should have the power of deciding if they want to install a software or not. Their knowledge also has a significant impact on their decision-making. So we believe in providing as much information as possible to users. Vendors in specific countries develop software applications and their major concern is to respect the local governmental laws. But these applications are widely in use. Some conflict may arise due to different government privacy policy framework. In fact, legal documents do not have the same presentation and the impact of the terms used vary from one country to another. Hence, there is a need to standardize EULA. Attempts to do this are ongoing²⁰. The main advantage is that the EULA will be standardized, well structured, using commonly known legal terms and thereby improving the performance of EULA analyser tools. A disadvantage to this could be an increase monopoly of few vendors that lead the standards and that are more concerned about profit as opposed to the informal activity that fuels the production of free software applications. More, software developers consider such initiative as a serious threat to their creativity that will be monitored by the law. It could lead to a situation where the law imposes certain functionalities in any software application [16].

²⁰ <http://www.lawdepot.com/contracts/softwarelicense/>

7.1.2 Using Multimedia in EULA

As stated previously, available EULA analyser tools need to be improved. One method of achieving this is to make use of pictograms and icons in the presentation of analyser results. Pictograms are more meaningful, attract user focus and improve his understanding. This idea is successfully applied in the traffic where drivers can easily understand the meaning by just looking at the picture. Audio can also be used in notifying the users in much more effective way. Pictograms could be used to create a summary of the EULA but the whole text is still available. Pictograms can be recognized internationally irrespective of languages, countries and users experience. A better approach might be to encourage vendors in using pictograms and multimedia in their EULA. Hence, analyser tools are no more required given that the vendors respect trade law. If it is the case, simple software applications may require additional storage space; such requirements may become redundant and excessive for users that install many software applications in their system. Approaches that can assure optimal usage of memory storage seem more realistic: machine-readable deeds are good examples.

7.1.3 Machine Readable Deeds

Installing a software application requires a lot of patience from users. Much time is spent in the pre-installation phase where users are requested to give their consent prior to the final installation of any software application. Analyser tools advantageously reduce the decision time for users; the tool developed in this study for instance, offers better performance in that way. An approach to automate the decision making process should be envisaged. Two considerations can sustain such initiative: the need to reduce the decision time during the installation of software applications and the need to present useful information regarding software to be installed. User agents that read specific EULAs available in XML format in specific vendors' web sites, and check their compliance with preferences stored locally in users system, can be developed. The installation process triggers user agents; message box windows that alert users about eventual information mismatch could be displayed prior to the final installation. Users' intervention during the installation of software applications is importantly reduced, their decision time is fractioned in CPU scale, and more security is achieved. A machine-readable deed requires a live connection to the public Internet, which could be unavailable in many cases. Community networks could be of

great value here, they provide pertinent information searched in vendors' web sites. In addition, they guarantee a central source of information in a local administrative network.

7.1.4 Reputation System

As mentioned above, reputation systems are collaborative networks where many users share their experience about software. Vendors' web sites have the drawback to present information that is subjected to marketing practices. Reputation systems present less twisted information, resulting from users' individual experiences about software applications. Statements from users are graded by others users depending on their pertinence [17]. Reputation systems are rich source of information that reduces the need to connect to different vendors web sites. Software deeds from many vendors could also be made available in XML format for operating system communication.

7.1.5 Deeds at Several Abstraction Layers

Deeds can also exist at several layers offering the opportunities to different classes of users, from the lowest to the most experienced ones to understand the behaviour of software applications in order to take a good decision. Our tool partially achieves this by using two layers for the presentation. A third layer could be easily added to our tool that displays pertinent comment about the usage of the software application to be installed. Such a pertinent comment could be issued from a reputation system, and may positively affect users' decision. Several layers of information require a good management, one specific layer having impact on the user's decision; why not direct the user to that layer?

All the above-mentioned improvements are not exclusive and could be integrated in one package like our tool. This guarantees better usage of our tool.

7.2 Research Questions

In this section we answer the research questions based on previous developed sections.

Q.1 How are the behaviours of existing programs notified to users through EULA?

We mentioned that important software behaviours are reported in EULA, but they are written in a very formal and legal language that is hard for users to understand: 25 out of 34 users do not understand the EULA. In addition the EULAs are very large, around 5000 words and may refer to other license agreements. This is a reason why many users do not read the EULA at all.

Q.2 How can the usage of EULA be improved by new mechanisms?

EULA analysers reduce the time spent on reading a EULA. Still users have to go through lengthy flagged sentences of EULA. Our tool is knowledge based and more convivial to users. This helps users in taking an effective and good decision in short period of time. Further improvement could be done by integrating a more semantic based analysis approach.

Q.3 How well could the activities of the software, contained in EULA, be provided to users in an effective way?

As discussed in section 7.1, we can use multimedia approach, usage of pictograms, multi-layer notification, EULA framework, reputation systems, machine readable deeds to effectively notify the users about the activities of a software contained in EULA. These mechanisms are not exclusive, they could be integrated to form a larger security solution.

Chapter 8

Conclusion

The main goal of this thesis was to see how the users could be helped in making a good and quick decision about the installation of any software application. An attempt was also made to see if the usage of EULA could be improved with respect to users' need. To achieve this, a user software preference module was constructed, where users could store their preferences about the behaviours of software applications they could tolerate in their system. Using this, the EULA was scanned to see if it complied with the preferences set by the user. A two layer format was used to present the result: the first layer is abstract and reveals all suspicious behaviours found in the EULA irrespective of the preferences stored and the second layer is more informative and in accordance with the preferences. A comparison with existing EULA analyser tools reveals that our tool was more convivial and more effective. It helps the user to take a quick and efficient decision. Further the feedback from users regarding our tool was positive. 11 out of 34 users found our tool very good, 15 found it good and remaining 8 found it satisfactory.

A number of issues to be addressed while developing a system like ours include the black listing of key words to be searched, rating of users' selection and also the presentation of the analysis result. During the development of EULA analyser we noted that it is more of a semantic problem than a syntactic one. We also proposed improvements of EULA, which includes software deeds, pictograms, multi-media utilities and abstraction layers.

If the software vendor does not report all the behaviours of his application, then the result from our tool may mislead the user. Government laws that protect consumers' interests could overcome such defections. Users have their own role in protecting themselves in today's digital environment. We believe our tool will make the role of users much more active, responsible and help them in fighting PIS by stopping them at the gate. Of course, this tool will not be a complete security solution, but rather it can be used as a part of large security solution suite including anti-spyware/virus tools, reputation systems and firewalls.

Chapter 9

Future Work

In this section, we discuss how the present study could have been carried out provided unlimited resources. We focus on the activities previously described that could be improved.

During the experiment, we could have scanned the selected software applications using more than one anti-spyware tool. By doing so, additional bad behaviours not mentioned in our study might have been detected. An identical good result could have been obtained by using the available commercial version of the anti-spyware used. Professional version of anti-spyware tool provides more details about the installed software.

The case study on existing EULA analyser tools is limited to free version of tools exclusively. Usage of professional tools, the commercial version of EULAnalyzer for instance, could result in more flagged sentences and suspicious key terms. This in turn could have been of good help in designing our tool by giving more information about suspicious key terms.

Improvements could be done in the core module of our tool, that is the EULA analyser module. Semantics-aware detection could be deeply implemented in this module, so that proper meaning of every sentence is obtained which helps in achieving better result. Considerations as automatic scanning of EULA triggered by the installation process, minimisation of the decision time, availability of the source of information may guide in selecting the best combination. A proper integration with

the improvements presented in the discussion part should also be envisaged as future work.

Given unlimited resources, the survey could have been carried out in a better way: time, space and location. Number of participants taking the survey could increase advantageously, so that the results obtained are more reliable.

Nevertheless, the study reveals a strong need of services in the area of preventive mechanisms against PIS.

Bibliography

- [1] M. Boldt, and B. Carlsson, "*Privacy-Invasive Software and Preventive Mechanisms*", in *Proceedings of the IEEE International Conference on Systems and Network Communications (ICSNC'06)*, Papeete French Polynesia, 2006.
- [2] N. Good et al., "*Stopping Spyware at the Gate: A User Study of Privacy, Notice and Spyware*", in *Proceedings of Symposium on Usable Privacy and Security (SOUPS) 2005*, Pittsburgh, PA, USA.
- [3] Data Protection Working Party, "*Opinion on More Harmonized Information Provisions*", Nov 2004, www.europa.eu.int/comm/privacy/ , 2006-10-20.
- [4] Centre for Democracy and Technology, "*Ghosts in Our Machines: Background and Policy Proposals on the Spyware Problem*", November 2003.
- [5] Platform for Privacy Preferences Project (P3P). <http://www.w3.org/TR/P3P11/> , 2007-01-15.
- [6] European Privacy Officer Forum, "*Comments on Review of the EU Data Protection Directive (Directive 95/46/EC)*".
- [7] J.W. Creswell, "*Research Design – Qualitative, Quantitative and Mixed Method Approaches*", 2nd Edition, Sage Publications, Thousand Oaks CA, 2002.
- [8] 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.

- [9] P2P Networking, "*FastTrack Architecture Overview*",
<http://www.cs.umu.se/~bergner/thesis/html/node64.html> ,
2006-11-19.
- [10] Skoudis Ed, "*Malware: Fighting Malicious Code*", Prentice Hall, Upper
Saddle River NJ, 2004.
- [11] StopBadware.org, "*WinAntiVirus 2006 Report*",
<http://stopbadware.org/reports/container?reportname=winantivirus.com%2F> , 2007-01-05.
- [12] Electronic Frontier Foundation (EFF), "*Dangerous terms: A user's
Guide to EULAs*", <http://www.eff.org/wp/eula.php> , 2007-01-05.
- [13] Susan P. Crawford, "*First do no Harm: The Problem of Spyware*",
Berkeley Technology Law Journal, Vol. 20, p. 1433, 2005, 55 Fifth Ave.
New York, NY 10003 United States.
- [14] Alba, J.W., and Hutchinson, J.W. "*Dimensions of Consumer
Expertise*", J. Consumer Research 13, 4 (1987), 411-454.
- [15] M. Flink, M. Covell, S. Baluja, "*Social- and Interactive-Television
Applications Based on Real-Time Ambient-Audio Identification*",
Google Research, Google Inc, 1600 Amphitheatre Parkway, Mountain
View CA, 94043, United States.
- [16] Patricia L. Bellia, "*Spyware And The Limit Of Surveillance Law*", Notre
Dame Law School, Legal Studies Research Paper No. 05-15
- [17] Tobias Larsson, Niklas Linden, "*Blocking Privacy-Invasive Software
Using a Specialized Reputation System* ", BTH, February 2007
- [18] Federal Trade Commission, "*Summary of Presentation by the Center for
Information Policy Leadership (CIPL)* ", GLB Interagency Meeting on the
ANPR on Privacy Notices, January, 2004.

Appendix A

Software GUI

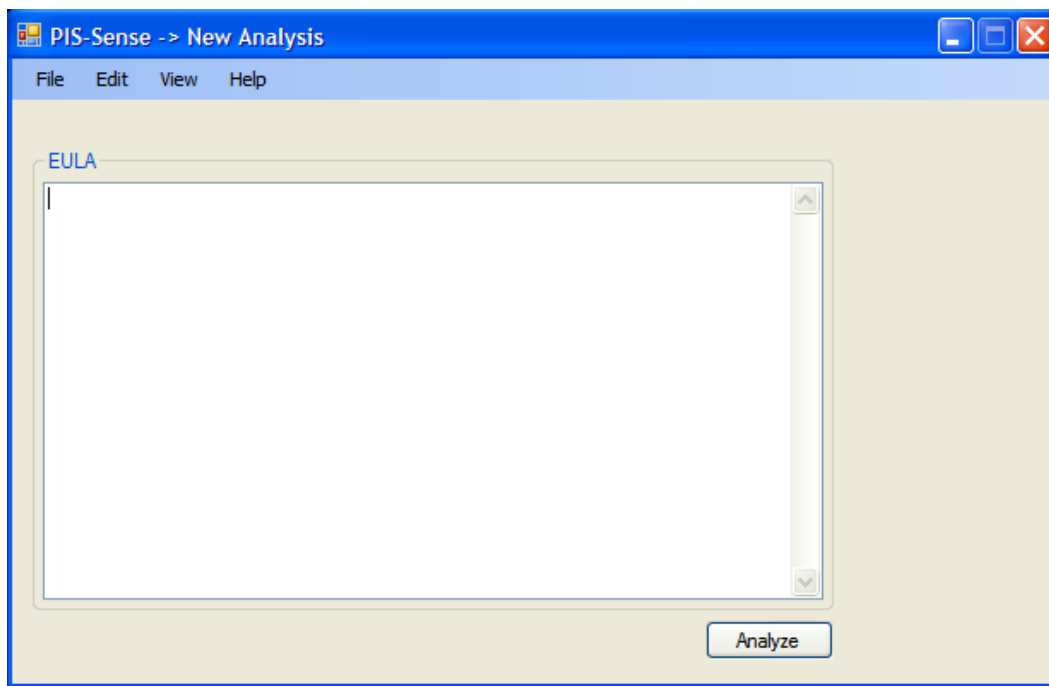


Fig A.1: The dialog box used by users to analyze a new EULA.

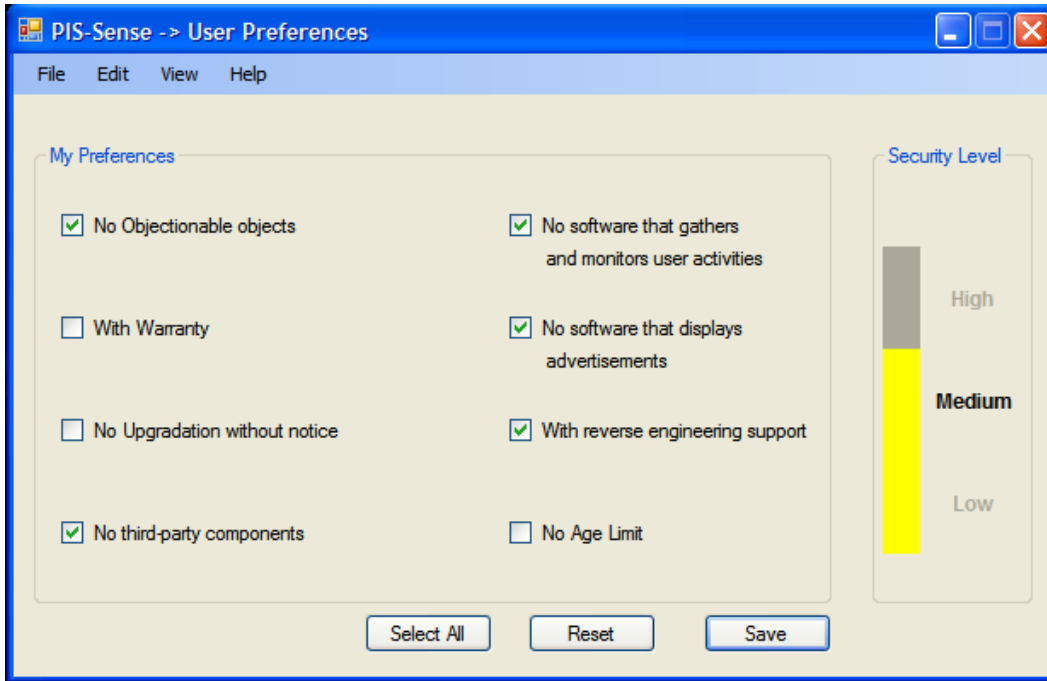


Fig A.2: The User Preferences interface where users can save their preferences. The corresponding security level according to the user preferences is also shown.

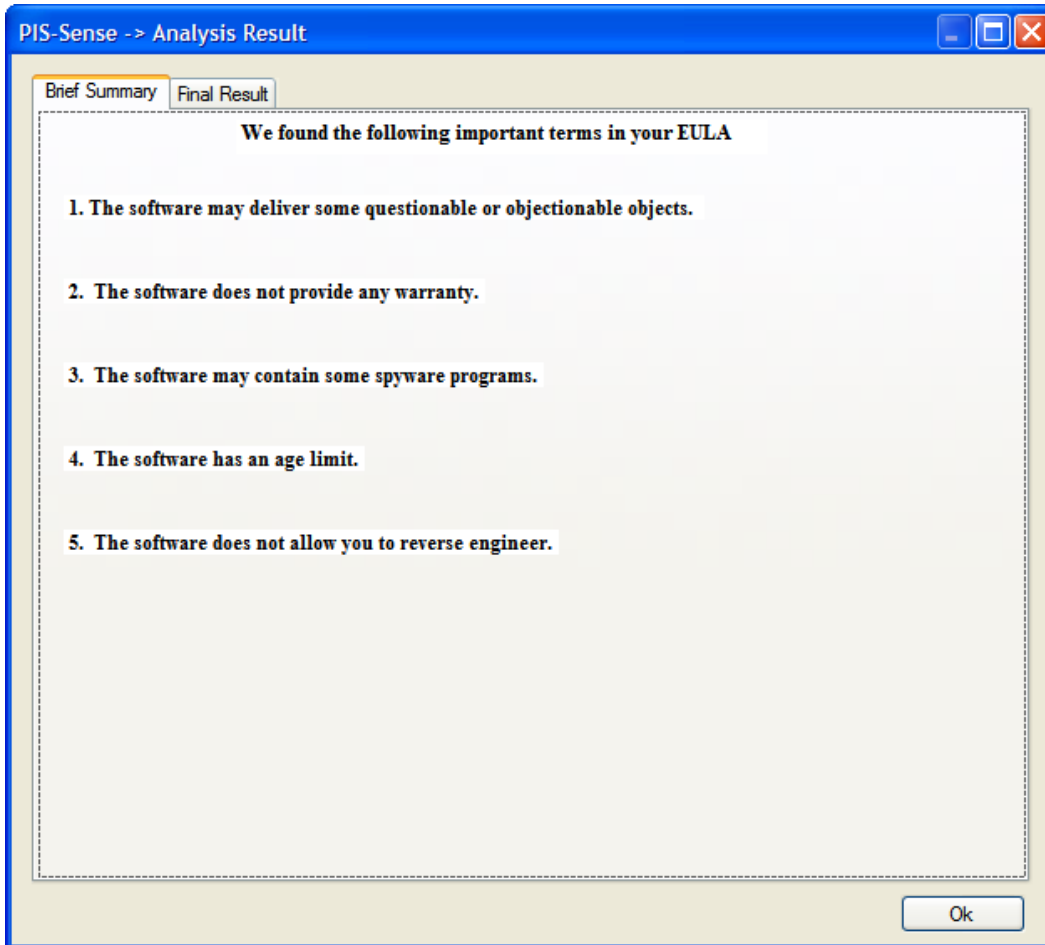


Fig A.3: The first layer which shows the results dialog displayed after analyzing the EULA.

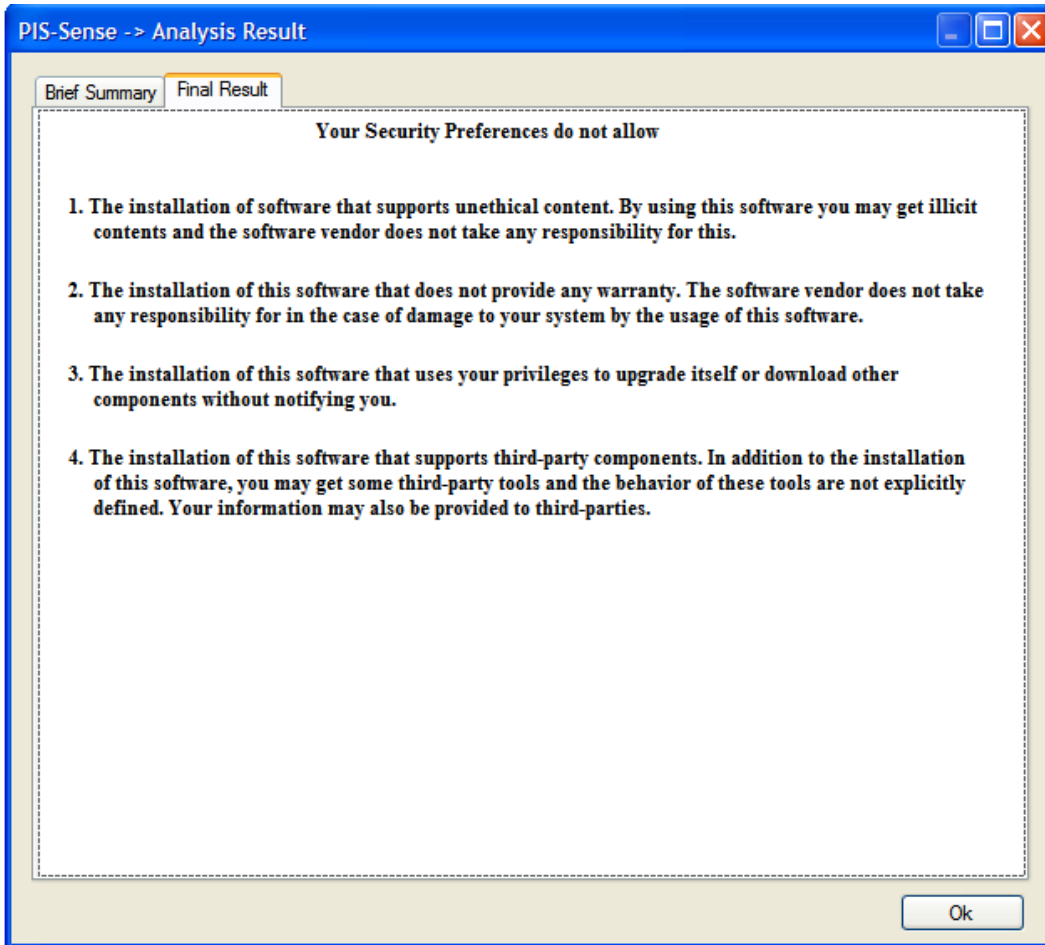


Fig A.3: The second layer which shows the results dialog displayed after analyzing the EULA with respect to stored user preferences.

Appendix B

EULA of Wengo

As a special exception, Wengo gives permission to link this program with the Qt Library (commercial or non-commercial edition), and distribute the resulting executable, without including the source code for the Qt library in the source distribution. As a special exception, Wengo gives permission to link this program with the OpenSSL Library.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that

you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights. We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software. Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations. Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you". Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice

and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original

licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any

later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. To do so, attach the following notices to the program. It is safest to attach

them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Also add information on how to contact you by electronic and paper mail. If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details. The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program. You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program

`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.