Fully Automated and Hidden System for Wiping Sensitive Data

GEORGE PECHERLE, CORNELIA GYŐRÖDI, ROBERT GYŐRÖDI, BOGDAN ANDRONIC
Department of Computer Science
Faculty of Electrical Engineering and Information Technology, University of Oradea
Str. Universitatii 1, 410087, Oradea
ROMANIA
gpecherle@uoradea.ro, cgyorodi@uoradea.ro, rgyorodi@uoradea.ro, andronic_bogdan@yahoo.com

Abstract: - In this article, we will describe a method to securely erase sensitive data in fully automated and hidden mode. Compared to other similar technologies, our method has two main advantages. The first one is the ability to run in a fully automated mode, in other words the system is configured once and the computer is protected without requiring any user intervention. The second advantage is the ability to run in a so-called secret mode, in which the system looks like another software, for the main purpose of confusing other users. Our paper will describe the structure and functionality of our system, and some of the most important technologies and algorithms that we have used.

Key-Words: - security, data wiping, data recovery, automation, scheduling, patterns, overwrite data

1 Introduction

Deleting a file using the operating system functions is not a secure operation. When a file is deleted, the operating system marks the disk areas previously occupied by the file as available for new data. Therefore, the old information is still on the hard drive, until new files happen to be saved in exactly the same locations. This information can be easily recovered by any basic software recovery tool [3].

Files can be deleted by:

1. The Windows user: for example, when the user deliberately removes one or more files that he no longer needs;
2. The Windows operating system or installed applications: during their normal operation, most applications create and remove temporary data, without the user's knowledge or approval.

In addition to the free disk space that can store a lot of previously deleted data, almost all applications save information on the hard drive that is meant to improve the user's experience. For example, web browsers save web pages, images and videos for quick access in the future (the web browser's cache). They also store a list of previously visited websites to enable the user to locate them faster (the web browser's history). A side effect of storing all this information is that it offers anyone a real portrait of the user's activity on the computer. And this is not always desirable.

To permanently wipe all traces left by Windows or applications, two important steps must be followed:

1. Finding out what information (files, registry keys, etc.) contain activity traces.
2. Securely erasing this information by using repeated overwrite operations to make it completely unrecoverable [3].

Normal users do not know where to look and find this information and even if they knew, securely erasing it in a continuous way would be a difficult and time consuming task. That's why, it is necessary that a software application (initially configured what to do) will do this job automatically and permanently. Almost all users have the experience and knowledge of running an antivirus on their system. The main advantage of a powerful antivirus is to let it do its job in the background and automatically eliminate threats (viruses, spyware, etc.). Our software acts on the same principles, the only difference is that it won't eliminate viruses or spyware, but it will eliminate sensitive data from the computer, such as traces left by other applications.

2 Advantages over similar technologies

There are a lot of data wiping software products that can destroy the traces left behind by the operating system or other applications. However, our research [1] indicates they have two main problems:

- Not completely automated: the user has to configure and run the wiping process periodically. Between these wiping processes, private data is saved and in danger of being discovered.
- Not completely hidden: even if some products hide the application while wiping data, there are still ways to determine that some hidden processes are running. Also, such hidden processes can be detected as possible threats by anti-spyware software [2].

The system we propose can address both of the problems above. The purpose is to develop a wiping system that ensures that at any moment, no sensitive data

is left behind (the system is always clean), without requiring any user intervention. The model we propose will lead to a new concept in data wiping: "just install and let it do its job". The main idea is to detect changes at specific locations (files/folders/registry keys), considered to be private locations, and erase new or updated data immediately.

Also, our application is easy to use even by novice computer users. If they trust the default configuration, they can simply accept it and just run the program.

Our idea of developing a fully automated wiping system came from a similar solution for a different problem, data backup. Genie Timeline from Genie-Soft [6] makes backup copies of user's data in an automated way, following the same principle, "set up and forget about it". We thought that something like this would apply very well for secure data wiping, not only data backup, so we developed the current solution, presented in this paper.

3 Implementation of our system in a software application

In order to show its advantages, we have implemented our fully automated and hidden system in a software application, that we have developed in Visual Studio 2008 using the .NET Framework and C# language. We tried to design it as simple as possible, so that it does not consume a lot of system resources and to be as fast as possible when running in the background to avoid slowing down the entire system [8].

The erasing mechanism is based on a high level erasing. This is done by opening the file that needs to be erased, replacing everything with random data and then saving it. This process is repeated for a number of times before the file is finally deleted [9].

We have chosen this method of high level wiping and not the low level wiping method (hard drive sector based) because of the nature of the file system: the fragmentation problem. Thus, the problem is avoided and the result is basically the same as low level wiping.

The following will describe the architecture of our application and the main functionalities we have implemented, to demonstrate the advantages of the fully automated and hidden system.

When the Main Application starts, the user will be asked if he wants to configure the application or run directly with pre-configured options. This way, we can identify two main modules:

1. The Configuration Module
2. The Execution Module

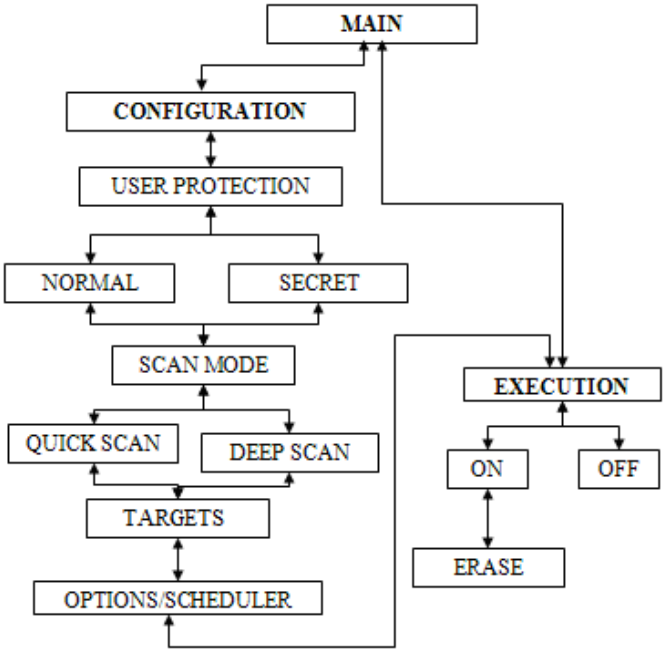

Fig. 1. Our application structure

1. The Configuration Module: when we designed the configuration module, we started from the idea that it has to be as simple as possible. That's why we divided it in 3 steps:

STEP 1 - The selection of the running mode: There are two categories of running modes.

The User Protection defines how the user is protected against someone who wants to see what this program is used for. This mode can have two values:

a) Normal Mode: a mode in which the program runs with an interface of a real data wiping system. This mode is recommended if you don't want to hide you are erasing data.

b) Secret Mode: a mode in which the program runs with an interface of a fake data wiping system. One example is to run it with an interface of an antivirus application. Everyone would think you are removing viruses and no one would suspect that you are actually erasing your secret data. Therefore, this mode is recommended if you want to hide you are erasing data.

The Scan mode defines what domains are scanned for sensitive data. This mode can have two values:

a) Quick Scan: a mode in which the program scans only pre-defined locations. These locations can be altered in the next step (Select What To Erase). The data is wiped immediately without user confirmation because the locations are considered to contain only private data. This mode is recommended for most users as it is a good compromise between performance and security.

b) Deep Scan: this will search globally (on selected disk drives) and determine new/updated data. Because not all data is sensitive, we can apply certain filters to

select only sensitive data. These filters can be: files that contain specific keywords, files of a certain type (MS Word, JPEG, etc.). The user can choose to trust these filters or he can select the user confirmation mode. The confirmation mode can also be optimized by choosing to erase immediately those files that are located in the pre-defined paths. Confirmation may seem like a non-automated way, however this is only until the system "learns" what data is sensitive or not. Little by little, the system will be able to take its own decisions and erase data without user confirmation. This is similar to the way anti-spam enabled applications work, such as Mozilla Thunderbird's Junk Mail feature [4]. They will ask for confirmation only until they "learn" what spam (in user's own opinion) is.

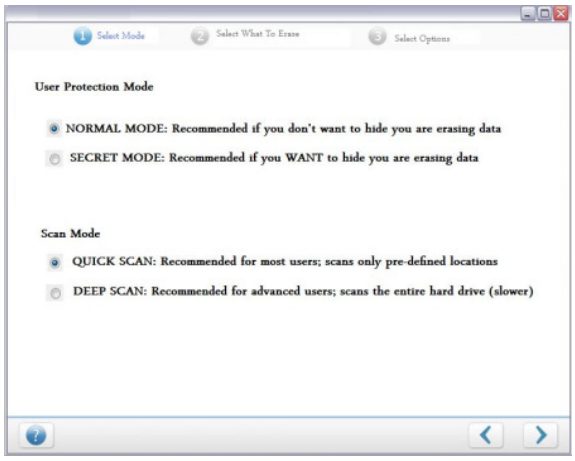

Fig. 2. Select Mode window

STEP 2 - The selection of what to erase: At this step, the user can select three types of sensitive areas. The first two are available both in the Quick Scan and Deep Scan modes. The third one is only available in the Deep Scan mode:

a) Traces Left By Applications: the user can select which application(s) he wants to erase usage traces for (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Microsoft Office, Yahoo Messenger, Skype, etc.). To define these options, we use a sensitive area definition language, called XSAD (eXtended Sensitive Area Definition), developed by us, as an extension to the OSAD language that we proposed at [2]. This new version of our language is based on XML and has a cleaner structure and better support for web based wiping systems. We use this language to define what sensitive areas (files/folders/registry keys) should be monitored for sensitive data. These sensitive areas can be places where applications leave their traces (such as the web browser's cache, history, cookies) or user defined sensitive areas.

b) Specific Files and Folders: the user can define individual files or folders to target. File masks, such as *.txt, are allowed to be used. At the implementation level, we use the same XSAD structure to save these files/folders locations.

c) File Filtering Rules (available only in the Deep Scan mode): it is a tool described above at step 1 that allows the definition of filtering rules based on file types or file contents. Based on these criteria, a file can be categorized as being sensitive or not. For example, someone could define a rule that all Microsoft Word files containing the term "financial plan" should be considered private.

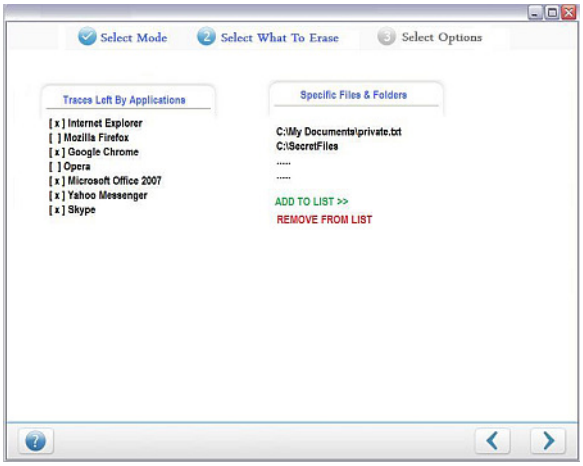

Fig. 3. Select What To Erase window (Quick Scan)

STEP 3 - The selection of options: At this step, the user can select various options, such as the overwrite method, whether he wants to confirm files before erasing or let it run in automated mode (default), the scheduling options (by default, the scanning process takes place continuously in the background, however the user can select a less frequent scanning, for performance reasons).

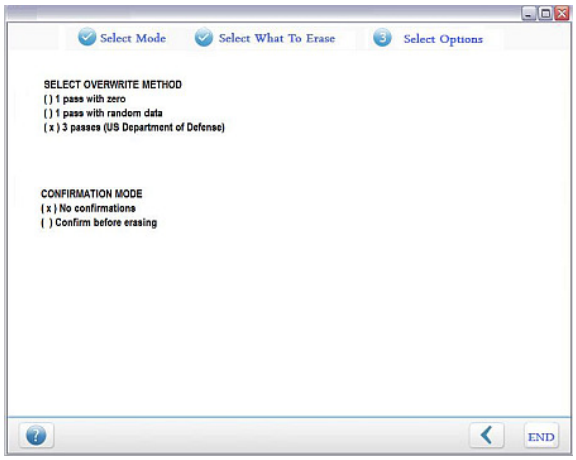

Fig. 4. Select Options window

2. The Execution Module: after following the steps above, the system is configured and ready to be run in an automated way. The last window is the main interface of the program that is displayed when the program is running. From this window, the user can switch

protection ON or OFF, he can view the status of the protection (what is being erased, when the last wiping process was run, etc.), or he can go back to the configuration module to change settings. A progress of the wiping operation is also displayed.

The execution module can also be accessed by double clicking its system tray icon. This icon will change its appearance based on the protection status (green – protection ON, red – protection OFF).

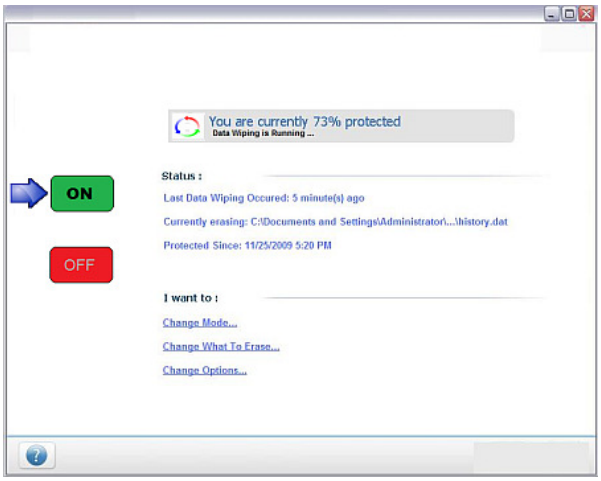

Fig. 5. The Execution Module

4 Hidden System Design

Our second purpose was to prevent other users from discovering there is a data wiping system installed on a computer. For this, we designed our wiping system to look like an anti-virus software. An anti-virus software is something that everyone has, so it is not suspicious at all.

Here is how some data wiping tasks are "transformed" into anti-virus tasks:

- Graphical user interface: all texts and images are updated to be related to an antivirus
- Scan process: when the system is searching for sensitive data, we rename this to something related to anti-virus scanning
- Wiping process: when the system is wiping sensitive data, we rename this to something related to virus removal. We will not show the names of all files that are removed, because there are usually thousands of files that are wiped, and it is usually more difficult to have so many infected files.
- Update process: when XSAD files are updated, we inform the user that virus definition files are updated
- Virus names: we can use a public virus list [5] and assign virus names randomly to files that we wipe

An option to switch to the normal (unhidden) mode, where everything looks like in the reality, is also offered to the user. This can be done from the Configuration module, in the Select Mode window (see Fig. 2).

5 Sensitive Areas Configuration Files. The XSAD Language

The configuration module has a panel where the user can select the applications he wants to erase sensitive traces from. Each of these applications (Internet Explorer, Mozilla Firefox, etc.) store their history traces in different places. In order to prevent hard coding these locations in our product, we developed a file structure that uses a definition language which we called XSAD (eXtended Sensitive Area Definition) [3]. This is actually pure XML language, customized for our own needs [7]. We tried to make it as simple as possible.

Here is an example of an XSAD file for Internet Explorer that shows the most important elements that can be used:

```
<?xml version="1.0" encoding="utf-8"?>
<SensitiveAreas>
<SensitiveArea name="Internet Explorer">
<Location name="Cache">
<Item detection="dynamic" type="folder">
<Detection>
<RegistryKey>
HKEY_CURRENT_USER\Software\Microsoft\Windows
\CurrentVersion\Explorer\Shell Folders
</RegistryKey>
<RegistryValue>
Cache
</RegistryValue>
</Detection>
</Item>
</Location>
<Location name="Most Recently Used">
<Item detection="static" type="regkey">
HKCU\Software\Microsoft\Windows\CurrentVersion
\Explorer\ComDlg32\LastVisitedMRU
</Item>
<Item detection="static" type="regkey">
HKCU\Software\Microsoft\Windows\CurrentVersion
\Explorer\ComDlg32\OpenSaveMRU
</Item>
</Location>
</SensitiveArea>
<FilesFolders>
<Item type="folder">
C:\Documents and Settings\Administrator\My Documents\Private Data
</Item>
<Item type="files">
C:\Documents and Settings\Administrator\My Documents\Company
Plans\*.doc
</Item>
<Item type="file">
C:\Documents and Settings\Administrator\My Documents\Company
Plans\SecretEmail.eml
</Item>
</FilesFolders>
</SensitiveAreas>
```

Fig. 6. An XSAD file for Internet Explorer

The XSAD file has two types of elements inside the root element <SensitiveAreas>. These two elements can be included in a single XSAD file (like in Fig. 6 above) or split in two distinct files for a cleaner structure:

- one or more <SensitiveArea> elements which contain what needs to be erased for a specific application. For example, a <SensitiveArea> element can be for Internet Explorer.
- one <FilesFolders> element that contains the files and/or folders defined by the user.

The <SensitiveArea> element has a name attribute containing the name of the application as it appears in the list. One or more nested elements, called <Location>, each with a separate name attribute, represent the various types of locations that can be erased from an application. For example, Internet Explorer has Cookies, Cache and History that have to be erased.

For each <Location>, we can have one or more <Item> elements which can be detected in two ways (this is specified in the detection attribute):

- static detection, in which the location is specified as it is (for example C:\My Secret Data\December.txt).
- dynamic detection, in which the location is determined following some rules (for example, the location of the Cookies folder for Internet Explorer is the value of a specific registry key).

If the detection is dynamic, there is a nested element called <Detection> containing other sub-elements that specify how the detection is done. In the example above, the Cache location is determined from a registry value. If the detection is static, the path can be specified directly as the value of the <Item> element.

Also, each <Item> can be of several types, and this is specified in the type attribute, which can be file, folder, regkey or regvalue.

The <FilesFolders> element can have several <Item> nested elements that describe what should be erased. In the type attribute, we specify what type of <Item> we refer to, and then in the value of the element, the path to the <Item> is provided, as in Fig. 6 above.

For increased security, we are using the following protection methods for XSAD files:

- the name of the XSAD file is not a descriptive one. It can be a random name and with a random or confusing extension (for example A5FB2U3C.DAT).
- the contents of the XSAD file are encrypted using an encryption method that can be specified by the user.

6 Conclusions and Future Work

There are a lot of data wiping solutions in the market and we wanted to bring out something unique, because we have determined that users find these solutions difficult to use and not providing continuous and complete protection. Our solution is mainly designed for home users however it can be integrated very well in a business environment, in small or large companies who want continuous data protection and to keep competitors away from their private information.

As future work, we want to bring more options to the user, implement an update module to keep all applications and XSAD files up-to-date and to improve our user interface.

References:

[1] „Privacy Software Review 2009 – TopTenREVIEWS" -

[2] Spyware Removers, by CNET Download.com

[3] „Detection of Confidential Data Using the Open Sensitive Area Definition (OSAD) Language" – George Pecherle, Cornelia Gyorodi, Robert Gyorodi – IEEE ICCP 2009 – Cluj Napoca, Romania

[4] Mozilla Thunderbird Junk Mail - http://www.mozilla-europe.org/

[5] VirusList.com – http://www.viruslist.com

[6] Genie Timeline by Genie-Soft – http://www.genie-soft.com/products/genie_timeline/default.html

[7] XML Path Language (XPath) Version 1.0 - http://www.w3.org/TR/xpath

[8] MSDN - http://msdn.microsoft.com/

[9] National Industrial Security Program Operating Manual, reissued February 28, 2006

[10] "Secure Deletion of Data from Magnetic and Solid-State Memory", Peter Gutmann, Department of Computer Science, University of Auckland. Published in the Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996

[11] The IEEE Computer Society – "Remembrance of Data Passed: A Study of Disk Sanitization Practices" - SIMSON L.GARFINKE, ABHI SHELAT (Massachusetts Institute of Technology) – January/February 2003

[12] "How to Forget a Secret", G. Di Crescenzo et al., Symposium Theoretical Aspects in Computer Science (STACS 99), Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1999, pp. 500–509

[13] NIST Special Publication 800-88: "Guidelines for Media Sanitization", Recommendations of the National Institute of Standards and Technology, Richard Kissel, Matthew Scholl, Steven Skolochenko, Xing Li, September 2006