

# Dynamic Analysis of Malicious Code: A Windows Operational System Approach

ADRIANO MAURO CANSIAN, THIAGO ALVES SIQUEIRA

Department of Computer Science and Statistics

São Paulo State University

Cristóvão Colombo Street, 2265 – São José do Rio Preto - SP

BRAZIL

{adriano, thiago}@acmesecurity.org <http://www.acmesecurity.org>

CÉSAR EDUARDO ATÍLIO

Technology Service Group

Kroll Brazil

Gomes de Carvalho Street, 1507 – São Paulo – SP

BRAZIL

catilio@krollworldwide.com <http://www.krollworldwide.com>

*Abstract*—The computers are increasingly more present in the people's way of life, being used for all sorts of activities, storing and transmitting their personal information in digital format. Following this tendency, the criminal activities evolved in the same way. Thus, appears the need for developing techniques to obtain and use electronic evidences, making use of concepts and methodologies of computer forensic expertise. Upon this context, this work will be focused on the dynamic analysis of artifacts, one of the techniques present in the computer forensic expertise, showing its concepts and methodologies. Finally, it is presented a case study using a malicious code distributed by electronic mail in the beginning of the year 2003 in the Brazilian Internet.

*Key-words:* - Internet, Computer Security, Forensic, Dynamic Analysis, Malicious Code, Keylogger

## 1 Introduction

The last decades were marked by the computer integration in the people's way of life. Nowadays it is common the use of the Internet for bank transactions and purchase purposes. All sorts of information started to be stored and transmitted in digital format. In the same way the criminal activities also evolved, aiming to obtain this information and make illicit use of it.

Presently, it is very common the use of mobile codes, which are executable programs copied from one computer to another through e-mails, web browsers etc., in order to execute a malicious attack against users. The Windows operational system is frequently target of attacks, partially because of its omnipresence and the large functionality that it provides. Some of these functionalities, such as executable codes attached to e-mails, provide opportunity for the malicious code to cause significant damage in a computer system. An obvious solution for this problem would be to disable these characteristics of the Windows. However, many users consider these characteristics convenient and a productive way through which they can guide their works. Thus, techniques that allow the protection against mobile codes without damaging the system functionality are necessary.

In this context is evident the expressive rise of the number of crisis involving computers. Thus, appears the need for a major understanding of how to obtain and use electronic evidences related to these crimes. This purpose can be reached through computer forensic expertise concepts and methodologies.

This work presents one of the techniques presented included in the forensic investigation process in a Windows computer system: the dynamic analysis of electronic evidences. It will be presented the concepts and methodologies used in this technique. The main objective of this work is to provide a detailed description of where, how and what search in a computer system affected by a malicious code. For such, several techniques, tools and procedures are presented, with practical examples that facilitate the comprehension. At the end, it will be presented a case study using a malicious code distributed by e-mail in the beginning of the year 2003 in the Brazilian Internet. This code, generically called "keylogger", has the capacity of capturing the presses of keys or the clicked area around the mouse. Its purpose is to capture personal information, such as bank accounts and passwords of users victimized by the actions of the code.

## 2 Computer forensic science

For the scope of this work, it is considered known that the Digital Forensic Science consists on the scientific methods of preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence. Its purpose is to facilitate or allow posterior reconstruction of criminal events or to help to anticipate non-authorized actions that show themselves anomalous to planned or expected operational behaviors [1]. In order to support the results of a forensic analysis are necessary the procedures and detailed protocols, revised and documented, accepted by the relevant scientific community [2]. For such, following are presented the pertinent procedures to a dynamic analysis of digital evidence.

### 2.1 Program analysis

There are several methods to study the behavior of a program. One of the ways is the dynamic study of the artifact when it is executed. For such, some tools can be used such as debuggers, function trackers, machine emulators, and logical analyzers and, in some cases, are used programs to monitor the net traffic. The advantage of the dynamic analysis in comparison with other methods is that it can be fast and precise. However, it has the disadvantage that the outlet of the obtained analysis is all what there is as result (“What you see is all you get”). It is hard to execute the program covering all the possible ways specified in its source code [3].

### 2.2 Dynamic analysis

The first concept to be defined is “malicious code” (artifact). In the context we are working, “malicious code” is a generic term which refers to all sorts of programs projected to effectuate some kind of non-authorized and undesired activity in computer systems. The dynamic analysis of artifacts is a precise process, based on the execution of the malicious code, observation and monitoring of all changes caused in the system in real time. This process is guided in order to be extensively documented and to obtain precise results. In order to this analysis be adequately guided, it is necessary a planning based on a police of analysis and the knowledge of the concepts and techniques to collect and analyze digital evidences. This planning involves firstly the analysis of the problem and the isolation of the artifact. Right away, begins the definition of tools for the search, collection and analysis of the digital evidences related to the execution of the malicious code.

## 3 Analysis methodology

### 3.1 Machine that will be used in the analysis

A reliable environment must be prepared for the use in all procedures comprehended by the analysis. Before the choice of the forensic machine operational system, of the set of tools that will be used and of the services this environment will have available, it must be determined which information is expected to obtain through the analysis, considering the characteristics of the artifact as well.

An essential aspect for the preparation of the forensic environment is to guarantee the isolation of this system in order to avoid possible malicious actions to damage other computers. One of the ways to get it is through traffic contention filters.

After determined that an artifact must be analyzed, it is necessary to collect it and generate a cryptographic hash [4]. This is an important tool for the analysis procedure of the file system information. The purpose of the hash of a file is to establish a signature of it in its reliable status, being a unicity property used to check the integrity. The hash is the one for the file from it was created, being different only if the file is changed. This guarantee is indispensable for, in future analysis by different members of the security community, being possible to check if the artifact obtained is the same one previously analyzed.

### 3.2 Environment preparation

For the preparation of the environment, it is relevant to determine the kind of equipment and operational system target of the malicious code. With this information it is possible to prepare a machine exclusively to perform this analysis. A machine used for dynamic analysis shall not to be reused, otherwise will be damaged after the execution of the code.

It is important the flexibility of the analysis environment since different network and system configurations can make the execution flow of the code actuates in several ways. If a request for a SMTP, DNS or IRC service is specified by the source code of the program, the network must provide these resources in a way to offer the configuration the code expects. Thus, the analyst can observe total performance of the malicious program.

The environment must have controlled access, that is, any changes executed in the system are detected. Any kind of change observed during the execution of the code must be taken into consideration. In order to reach this objective, correct tools must be used. Depending on the operational system, different behaviors must be

expected. In the Windows operational system, i.e., it can be expected in the system register. Thus, it becomes necessary a tool which monitors any change on it.

Apart of the local changes, can occur alterations remotely as well. Innumerable artifacts have the purpose of making remote attacks against other computers. It can simply disable the network connections of the analysis machine, which not always is the best solution. In some kind of artifacts, like the case study in the end of this article, it is important to define what the network sends. This way, it must be allowed for the machine to do external connections, but controlling them. For such, filters in the firewall [5] of this network can be used to block them.

An artifact analyst must have large knowledge on several tools for different operational systems. It must be known exactly how to use them in order to make the analysis an efficient and effective process. Many times the analysis of an artifact is a critical process. It must be quickly defined the consequences of it in order to solve them. Thus, as it is a quick process and does not need correct results, it is desirable the minor time to obtain it with the least mistake probability.

### 3.3 Digital evidences

The expression “digital evidence” refers to all digital information able to determine that an incident occurred or that establishes some relationship between the incident and the victims.

Dan Farmer and Wietse Venema introduced a concept called volatility order [6]. Such concept determines that the lifetime of electronic evidences varies according to the place where it is stored. The greater is the volatility of information, the harder becomes its extraction and there is less time to capture it. The simple fact of observing high volatile information can change them. The specification of each source of information, for the purposes of this article, as well as the evidences disclosed in each one and the techniques used for extracting the information, is presented as follows.

### 3.4 Evidence collection

The search for evidences in a computer system consists on precise sweepings in the information included on it, be them data in files or in memory, deleted or not, coded or possibly damaged [7].

When guiding an investigation on the behavior of a program, the investigator needs to collect information about the current status of the system where it is being executed. Some methods must be used in order to collect this kind of information and then save it in another forensic machine not to write or change the status of the

machine used to host the malicious code. For such, a good tool is *netcat*<sup>1</sup>. A process server of netcat can be configured in the forensic machine to collect this information and store it in a file through the following command:

```
C:\> nc -l -p 1234 > c:\archivo.dat
```

In this case, the forensic station is configured in order to receive the information through the door 1234 and store them in the file *archivo.dat*.

```
e:\> pslist | nc 12.34.56.78 1234
```

In the second step, the desired command is executed in the analysis system and its outlet is redirected to the netcat program, informing the IP address of the forensic station and the number of the door configured to receive the data where *e:* represents the CD-ROM drive in which are recorded all the tools that will be used in the information collection, including the native tools of the Windows system. It provides additional security in order to guarantee that their security versions will be used and also to guarantee that they will not be infected by the malicious code that is being studied.

## 4 Collecting volatile information

### 4.1 Processes

Volatile information can be described as the information that represents the current status of the system. The first item of interest for an investigator is the set of processes in execution in the system, since such information can disclose evidences of non-authorized activities. Information regarding the process can be obtained through the use of a combination of several tools. As example of such tools it can be mentioned the *Pslist.exe*<sup>2</sup>, which provides a list of processes in execution, the ID number of the process (PID), and the time for starting execution, besides the data about the change in process. *Fport.exe*<sup>3</sup>, lists the processes, the ID

<sup>1</sup> The netcat program and other details about it can be found in the URL

[http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/) (available in February 2004).

<sup>2</sup> The program pslist.exe and additional information about it can be found at

<http://www.sysinternals.com/ntw2k/freeware/pslist.shtml> (available in February 2004).

<sup>3</sup> The program Fport.exe and additional information about it can be found at <http://www.foundstone.com/index.htm?subnav=resources/navi>

number of the process (PID) and the way of the executed command, as well as the TCP and UDP opened doors relating to the process. *Listdlls.exe*<sup>4</sup> lists the DLLs<sup>5</sup> that are in use for each process. This tool also provides the ways for the DLLs that are charged.

## 4.2 Network connections

The study of the network connections provides valuable information on the connections in course and processes waiting for a connection [8]. From these information it is possible to determine if exists any non-authorized connection (or suspect) in course. It can be done using the tools *netstat*<sup>6</sup> and *nbtstat*<sup>7</sup>.

Another volatile information consists on the data related to the communication between victim system and other system through the network. For the Windows NT/2K platform there are numerous programs that use the libraries *winpcap*<sup>8</sup>, commonly called *sniffers*. Besides capturing the datagrams that pass through the network, the sniffers can decode and display them in a more legible format. The most common example of these programs is the *Ethereal*<sup>9</sup>. The *Ethereal* can be used to capture all sort of network traffic, decoding and displaying the datagrams as they are collected or store the datagrams in a binary file, allowing a posterior analysis.

## 5 Collecting non-volatile information

Non-volatile information consists on the system configurations that do not change all the time or when the system is reinitialized.

---

gation.htm&subcontent=/resources/overview.htm (available in February 2004).

<sup>4</sup> The program *Listdlls.exe* and additional information about it can be found at <http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml> (available in February 2004).

<sup>5</sup> More information about DLL at <http://support.microsoft.com/default.aspx?scid=kb;en-us;815065> (available in February 2004).

<sup>6</sup> The program *Netcat* and additional information about it can be found at [http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/) (available in February 2004).

<sup>7</sup> More information about *nbtstat* can be found at <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxp/proddocs/nbtstat.asp> (available in February 2004).

<sup>8</sup> More information about *Winpcap* can be obtained at <http://winpcap.polito.it/> (available in February 2004).

<sup>9</sup> The tool *Ethereal* and additional information about it can be found at <http://www.ethereal.com/> (available in February 2004).

## 5.1 Files

The source of information where the forensic analysis process often is concentrated is the file system. Like a database, the file system is the part of the operational system responsible for organizing the information of the disk in file format [9]. One of the information relating to the files that the investigator must collect is the time mark. Such marks correspond to the times of last change in the file, last access, and last change in the properties of the file (*MAC times*) [10]. An example of tool that can be used of this purpose is the *afind.exe*<sup>10</sup>.

ADSs<sup>11</sup> (*alternate data streams*) is a characteristic of the file system NTFS, which allows the storage of data and is executable in a way that these data are invisible for the tools provided by Windows, such as **dir**, **Windows Explorer** etc. Frank Heyne developed a tool called LADS<sup>12</sup>.

The investigator can decide to effectuate an analysis of the signature of the files. In Windows systems, the operational system relates files with executables based on the extension of the file. For example, files with the extension “.ppt” are related to Power Point. Files can be hidden in the system when their extension is changed, like when replacing “myphotograph.jpg” by “myphotograph.txt”. When a signature analysis is developed, the first 20 bytes of the file are analyzed, and then the check is done trying to relate this analysis with the correct extension of the file. For example, executable files with the extensions DLL, SYS, EXE and others, have the characters “MZ” in the first bytes of the file.

Audit the events occurred before and after the execution of the artifact is an important part of the analysis of its behavior. This process can evidence different behaviors of the patterns that characterize the normal use of the system and even observe tendencies in the use of resources of the computer system. *DumpEvt.exe*<sup>13</sup> is the tool projected for this purpose.

---

<sup>10</sup> The package that contains the *Afind* and additional information about it can be found at <http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/overview.htm> (available in February 2004)

<sup>11</sup> Other information about ADSs can be found at [http://www.chi-publishing.com/portal/backissues/pdfs/ISB\\_2001/ISB0601/ISB0601HC.pdf](http://www.chi-publishing.com/portal/backissues/pdfs/ISB_2001/ISB0601/ISB0601HC.pdf) (available in February 2004).

<sup>12</sup> The program LADS and additional information about it can be found at [http://www.heysoft.net/Frames/f\\_sw\\_la\\_en.htm](http://www.heysoft.net/Frames/f_sw_la_en.htm) (available in February 2004).

<sup>13</sup> Other information about the program *DumpEvt.exe* can be obtained at <http://www.systemtools.com/somarsoft/> (available in February 2004).

## 5.2 Register

The register of the system is defined as a hierarchical database used by Microsoft Windows 9x, Windows CE, Windows NT and Windows 2000 used to store the necessary information to configure the system for one or more users, applications and hardware devices. The data of the register are stored in binary files. Thus, these databases can contain valuable evidences regarding the behavior of the malicious code. A native tool of the system used to collect information from the inlet of the register is **Regedt32.exe** for Windows NT and the **Regedit.exe** for the other Windows versions.

## 5.3 Documentation

Simultaneously to the preliminary considerations pointed out in the last section, to the preparation process and to the analysis itself, it must be started the documentation process.

The basic information that must consist in the documentation of a dynamic analysis are the data of the analyst, data of the responsible by the collection, and date and time of the action execution. The importance of these items is to specify in time the occurrence of the attack and, if there is any question about the analysis and the collection, to know who to refer. The date when the hash was generated must be specified.

It must be described the way the artifact was obtained (e-mail, *honeypot* etc) and why it was concluded that the artifact could be a malicious code. This action can work as a characterization of how such artifact is normally used.

After the initial descriptions, it must be precisely reported all the steps used during the analysis. Firstly, it is mentioned how it was decided which tools should be used. After choosing the necessary tools, it must be described the function of each one in the analysis, what results are expected by using them, besides their versions. Right away, it must be carefully described the production of the appropriate environment for the analysis of the artifact. In this description, it also must be described the reason why that environment was regarded as ideal. All the steps, as well as functions and commands of the desired tools selected and used in their implantation, shall be mentioned.

After the environment is constructed, it begins the analysis process. All the details of analysis, as well as the reasons why certain action is being adopted, must be recorded. It must be evidenced in the documentation relevant information of the alternative provided by the tool used and what conclusion comes from this result, as well as the need for reaching such result. Invariably, the result obtained through the use of a tool will be used for

the choice of another tool. In such case, it is necessary to come back to the beginning of the documentation and specify any new decision taken in the course of the analysis.

Finally, a conclusion about the analysis, disclosing what is the objective of the malicious code, what are the consequences it can cause to the system and the characteristics after its installation. It is important, if possible, to point out a solution to solve the problem and reestablish the system to its original status.

After presenting the necessary items for a good documentation, each analyst of artifacts can create a personal pattern. The important aspect is that all documentation contains the aspects mentioned up to now. This way, the exchange of information among the analysts becomes complete, helping the development of the security community.

## 6 Case study

Below, it is presented a case study consisted by a dynamic analysis of a malicious code. This program was distributed through e-mail in the Brazilian Internet in May 2003. This e-mail contains a persuasive message of cognitive nature and an attached program. This program (*certificado\_digital.exe*) installs the real artifact *MSNBC32.EXE* in the system in which it was executed. For the purpose of this article, the analysis data were sanitized.

### 6.1 Dynamic analysis of the risk artifact

Analysis: 12/2003

#### **certificado\_digital.exe / Desejos.exe (variação)**

Nature of the Artifact: Borland Delphi

Affected Systems: Systems using Microsoft Windows

Codename: false-verisign

Hash: 7e5776f9c965307d8033433cbdbc6bde

File Size: 334567 bytes

Data of the Analyst: xxxx

Data of the Responsible by the Analysis: xxxx

Data and Time of the Analysis: 05/13/2003 – 10 p.m.

### 6.2 Motivations for the analysis

- Malicious e-mail received by several people with information apparently unauthentic;
- Such attitude does not fit with the behavior of the company regarding the e-mail sending not requested;  
A file with a persuasive message of non-confirmed origin;

- Messages of discussion list describing the partial behavior of the artifact.

Cognitive message distributed through e-mail, translated to English:

*"Get your digital certificate now*

*This is a VOS message instructing for a better On Line Security.*

*The major exigency of the Internet users when doing online transactions is the reliance in doing it safely. One of the ways to afford this reliance is through the Digital Identification (Digital ID), based on security technologic measures, called Digital Certificate.*

*These digital certificates show that the sending of information to a site was done through a nearly inviolable way, since it is cryptographed. Because of that the sites that use this security mechanism in the network are in the elite of Internet companies that assure to the user absolute tranquility.*

*VeriSign® visa:*

*\* Protect or codify a message in order to it can only be received and deciphered by the addressee through a public password that contributes to activate the protection.*

*\* Create sophisticated access permissions indispensable for the transactions.*

*\* Compare the digital signature of the user who has the certificate in order to all the sending of messages and all the transactions effectuated by it cannot be subject for plagiarism.*

*Get your digital certificate now*

*Install the VeriSign® Site Security Certificate now*

*Install a VeriSign® Site Digital Certificate and have all the advantages of an exclusive and secure access. Just click the icon VeriSign® in any point of the screen, effectuate the Download and execute the Digital Certificate Program.*

*Click here and effectuate the Download of your Digital Certificate.*

*Verisign Inc. S/A. Security Team"*

### 6.3 Analysis environment

Forensic Machine:

Operational System: Windows 98 SE

Processor: AMD Athlon 800MHZ

RAM: 128MB

Disk: 2 GB

This machine is part of a network that has more two machines using the Linux operational system. This network is limited by a *firewall* (Zwicky and Cooper, 2000) using IPTABLES (Stephens, 2002). The objective of these machines is to monitor and capture the intention of the forensic machine with the environment when the program is executed. Strategically, the contention system has the purpose of repressing any kind of malicious traffic addressed to any forensic environment external computer.

### 6.4 Artifact conclusion

21:54:14 The program is executed in the forensic machine described above.

The outlet of the program pslist.exe provided the following information:

Name (MSNBC32), Pid (1700), Elapsed Time (0:00:12.031)

Files created or changed from the execution of the artifact:

C:\WINDOWS\MSNBC32.EXE

C:\WINDOWS\Debug32

C:\faxsetup32.txt

C:\WINDOWS\Debug32\Clickxxx.jpg

C:\WINDOWS\SERVONE32.DLL

C:\WINDOWS\SERVTW032.DLL

C:\WINDOWS\ZIPDLL.DLL

C:\WINDOWS\SYSTEM.CB

(File that contains information of the keyboard).

It was reported the creation or modification of these files according to their time attributes, corresponding to the time of execution of the program. This verification was performed many times in order to guarantee that the

modifications can be attributed to the execution of the artifact.

After the program is executed once, modifications in the system register occur in order to execute the program automatically when the system is reinitialized. Such modification is notified as follows:

It is added a new *string* value with the name MSNDC32.EXE, referring C:\WINDOWS\MSNBC32.EXE in MyComputer\HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

When Windows reinitializes the program, the file C:\faxsetup32.txt is created. Other possible locations of MSNBC32.EXE in the register are RunServices, Run-, RunServices-, RunOnce.

### 6.5 Artifact behavior

When starting its execution, the file MSNBC32.EXE creates the file C:\faxsetup32.txt, which will store a summary with the system specification, the presses on the keys and the titles of the active windows. Besides, it temporarily correlates the key presses with photographs of the area clicked by the mouse.

The files Clickxxx.jpg, created in C:\WINDOWS\Debug32, store 25 x 25 pixel images in 16 million colors around the cursor, when the left button of the mouse is clicked. There are variations of the artifact behavior that record the images obtained from the system in C:\WINDOWS\Control32.

The programs capture images and the behavior of the system for about 400 seconds. Ended this time, the program sends by e-mail the files faxsetup32.txt and all Clickxxx.jpg, being zzz@zzz.zzz its addresser and xxx@xxx.xxx.xx and yy@yyy.yyy.yy its addresse. Below is the file faxsetup32.txt created (Sanitized data):

```

-----
ID Equipment Data
-----
Operational System : Windows 95/98/Me
-----
Computer Name : VWwin98
-----
System Version : xxx
-----
System Compilation : xxxxxxxx
-----
File System : FAT32
-----
Local Disk Name : Disco Local
-----
HD Serial: xxxxxxxx
-----

```

```

System Date : 5/13/03
-----
System Time : 10:07:55 PM
-----
Local Computer IP: 192.168.0.98
-----
Active Window - Servant Salamander
< Click1 >
< Click2 >
< Click3 >
< Click4 >
<Tab> <Down> <F3> <Esc> <Down>
Active Window - C:\WINDOWS\Debug32\Click2.jpg
(25 x 25 x 16777216 colors - 100%) - PictView
<F3> <Esc> <Down>
Active Window - C:\WINDOWS\Debug32\Click3.jpg
(25 x 25 x 16777216 colors - 100%) - PictView
<F3> <Esc> <Down> <F3>
Active Window - C:\WINDOWS\Debug32\Click4.jpg
(25 x 25 x 16777216 colors - 100%) - PictView
Active Window - Servant Salamander
<Esc> <Ctrl> <Alt>
< Click5 >
< Click6 >
< Click7 >
.
.
.
< Click13 >
Active Window - Microsoft Internet Explorer
< Click14 >
Active Window - C:\WINDOWS\Debug32\Click9.jpg -
Microsoft Internet Explorer
< Click15 >
.
.
.
< Click298 >
< Click299 >
<Tab> c <Shift> <Enter>

```

<Down> de  
<Down> <Down> <Down> <Down> <Down> <Down>  
<Down> <Up> <Down>  
<Up> <Up> <Up> <Up> <Up> <Up> <Down>  
<Down> <Down> <Down> <Up>  
<Up>  
<Up> <Up> <Up> <Up>

## 6.6 Anomalous behavior of the system obtained through the artifact execution

- System presents loss of performance;
- Sending of messages accusing lack of resources or of RAM memory;
- Mouse cursor blinks intermittently.

## 6.7 Recommendations for cleaning damaged machines

The following steps must be followed for cleaning damaged machines:

If the machine can be formatted:

1. Disconnect the damaged machine of the network;
2. Make security copy of the personal files;
3. Format the damaged machine;
4. Replace all the passwords inserted in the system since the moment the artifact was executed, including passwords and numbers of credit cards;

If the machine cannot be formatted – Risk option (not recommended!):

1. Remove the damaged machine of the network;
2. Exclude through Prompt DOS the files:  
C:\WINDOWS\MSNBC32.EXE  
C:\WINDOWS\Debug32\\* ou  
C:\WINDOWS\Control32\\*  
C:\WINDOWS\faxsetup32.txt ou  
C:\faxsetup32.txt
3. Remove the access of the register:  
MyComputer\HKEY\_LOCAL\_MACHINE\  
SoftwAre\Microsoft\Windows\CurrentVersi  
on\Run\MSNBC32.EXE;
4. Replace all the passwords inserted in the system since the moment the artifact was executed, including passwords and numbers of credit cards;

When the information is sent to the addressees mentioned above, many particular information of the

system are sent attached. Because of that is highly recommendable to reinstall the system, modifying some of its characteristics, such as the version of the operational system, name of the machine and its IP address in the network.

## 6.8 Analysis conclusion

Considering this behavior, it was presumed that one the purposes of the program is to capture passwords or personal information through an interface of the type virtual keyboard, nowadays very used in Netbanking systems. The virtual keyboards are nothing but a graphic interface that allows simulating pressing keys through using mouse clicks. Several bank institutions developed a virtual keyboard considering it the ideal solution against trojans and hackers.

Another purpose of the program is to capture presses of system keys in order to obtain information typed from the keyboard.

## 7 Conclusion

The study presented in this work represents an effort directed to supply the necessity of a better understanding of how to obtain and to use electronic evidences stored in computers. The discussion about the numerous procedures involved in a dynamic analysis, specifying methodically each step of an analysis, provides a practical guide for those who are initiating in the forensic computer area. Also, the presentation of the case study allows to guide the investigator in the choice and manipulation of the tools of analysis, as well as in the conduction of an analysis and mainly in the behavior of a malicious code.

## 8 References

- [1] G. Palmer, A Road Map for Digital Forensic Research, Digital Forensic Research Workshop (Dfrws), *Report* 2001.
- [2] M. Noblett, M. Pollitt, L. Presley, Recovering and Examining Computer Forensic Evidence, Forensic Science Communications, Number 4, Volume 2, U.S. Department of Justice, FBI.
- [3] W. Venema, Finding the purpose of an unknown program, Strangers In the Night, Dr. Dobb's Journal, 2000.
- [4] B. Scheneier, Applied Cryptography, John Wiley & Sons, New York, 1996.
- [5] J. C. Stephens, Iptables. Available: <http://www.sns.ias.edu/~jns/security/iptables/index.html>

- [6] D. Farmer, W. Venema, Computer forensics analysis class handouts. Available: <http://www.fish.com/forensics/class.html>
- [7] M. Abdalla, P. Geus, Forense Computacional: Procedimentos e Padrões, Information Security Symposium, 2001, São José dos Campos, Annais SSI2001.
- [8] W. G. Kruse II, J. G. Heiser. Computer Forensics: Incident Response Essentials, Addison-Wesley, Reading, Massachusetts, 2002.
- [9] A. Silberschatz, P. Galvin, Operating System Concepts, John Wiley & Sons, New York, 5<sup>th</sup> Edition.
- [10] E. Casey, Handbook of Computer Crime Investigation, Academic Press, San Diego, California, 2000.