Forensic Analysis of a Compromised Intranet Server

GCFA Gold Certification

Author: Roberto Obialero; roberto.obialero@poste.it

Adviser: John C.A. Bambenek

Accepted: March, 3rd 2006

Abstract

Nowadays companies are geographically distributed, thanks to the globalisation and the pervasive presence of the Internet. This is a great business and organizational opportunity: this way employees can be very collaborative despite their location and distance, but this could have some drawbacks.

One issue may be related to organization hardware and software inventory managed by system administrators located in the company headquarters. If they don't have expensive management and configuration software available or they aren't timely updated about hardware and software changes by branch office personnel they would not be aware of the resources they are managing and the security vulnerability risks could increase.

It could sometimes happen that some minor company servers become completely forgotten so they run unpatched and without antivirus signature update, paving the way to the spreading of viruses and worms.

Without countermeasures in place, driven by system administrators, employees may be unaware of such risks and they may continue to use those servers for daily operations, contributing to the spreading of malware infections.

This document details the forensic analysis process of a compromised Intranet server, from the verification stage to the dissection of malware code, supported by an explanation of the followed methodology.

Some information, about the server and the company it belongs to, have been sanitized in order to preserve the privacy of the business.

Table of Contents

Table of Figures

Index of Tables

1 Forensic investigation methodology

Before entering the analysis in detail, it is important to take an overall look at the investigation process, described in the flowchart depicted below:

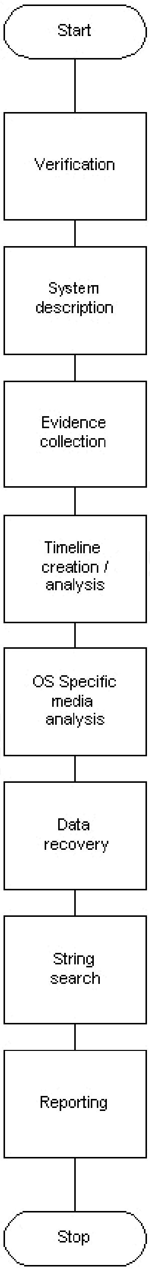


Figure 1 - Forensic investigation methodology

The following paragraphs explore in detail the steps of the process[1].

1.1 Stage 1: Verification

The first phase of the investigation process is the task called verification: during this stage the forensic examiner called on duty takes a careful look at the information logged by the system, by the antivirus applications and by the network devices (firewalls, IDS, routers) to be sure the incident effectively occurred.

During the verification stage, the Incident Response Team (IRT for short) members encounter two typical situations:

1. Dead system with the power unplugged (computer system off) and the media frozen.
2. Live system with the power and operations on (processes running, disks being accessed and active network connections).

In the latter condition the forensic analyst must be very careful to avoid the volatile information's destruction (processes, memory, network connections).

During this phase the forensic examiner makes use of a set of simple and trusted tools to check the presence of abnormal network connections, rootkits, strange directories, and binary files recently installed.

1.2 Stage 2: System Description

Once the verification task is successfully completed and the IRT is certain that the security incident occurred, the forensic examiner has to fill in a detailed system description. The information ranges from the hardware and software system characteristics, the hard disks geometry (useful in the following media analysis phase), what the computer is used for, the list of users and many other useful information.

Part of these data will be pulled out from the system using trusted media software tools: the investigators cannot rely on the system target of the attack tools, since they could be trojanized. A set of these data could also be used for the chain of custody, in case of computer seizure by Law Enforcement officials.

1.3 Stage 3: Evidence collection

This stage is crucial to the forensic analysis process: all of the available computer information must be transferred to an external media or to a forensic workstation in order to perform the next analysis tasks. This operation is critical because the examiner needs to be sure that only the original data is transferred and taken into account.

All of the collected system data (memory, processes, network connections, disk partitions) must be ear-marked using the cryptographic MD5 hash technique to ensure the data integrity.

The MD5 hash is a 32 bit long string created computing the data belonging to files, partitions, disks by a mathematical algorithm; such operation is irreversible (it is not possible to recover the original file starting from its hash value) and it is mathematically impossible for two different files to produce the same MD5 sum.

A relevant part of such data must be obtained using automated tools in order to avoid mistakes, perform the task in timely fashion, collect data in the right order of volatility and safeguarding the data integrity validation by hash signatures.

Before starting this task the forensic examiner has to verify the integrity of the system used for the analysis: it is mandatory to have at disposal a sterile image media to avoid previous installations data tracks left on the media that can cheat the next analysis phase.

This stage terminates the data acquisition process tasks.

1.4 Stage 4: Timeline creation and analysis

Once the evidence collection process is completed and all of the data of the system under analysis are stored in the forensic workstation, the first task to perform is the timeline file creation. This is a complete image files and inodes[1] listing associated with the MAC times[2] info; it is very useful to trace back the system activity; (the timeline file prints out the last time an executable file was run, the last time file or directory were created/deleted and it could also prove the presence of scripting activity).

The main reason for performing the task before any other is that every command run on mounted images could accidentally rewrite the last file access time.

The timeline creation process consists of two parts: the first one generates the intermediate body file with all the information (data and metadata) pulled out from the image file and the latter sorts such data chronologically in ascending order.

1.5 Stage 5: OS-specific media analysis

[1] File descriptors, attributes, MAC times

[2] Modify, Access and Creation times

Starting from the findings obtained by the timeline analysis, the forensic examiner can begin the media analysis searching for clues behind compromise of the system. The toolset available to the forensic analyst depends on several variables:

- the software platform used in the forensic workstation;
- the software platform used in the system target of the analysis;
- if the analysis has to be performed on a live system;
- the network configuration.

In this stage, the forensic analyst has to examine thoroughly the media layers (physical, data, metadata, file system and file name) searching for evidence of suspicious binary installations, files and directories added, removed but open files and so on.

From the forensic examiner point of view, Linux is the more flexible operating system supporting most of the currently used file systems and having a more complete toolset than Windows; it must be underlined though, that a set of Linux forensic tools have been successfully ported to the Windows platform.

1.6 Stage 6: Data recovery

After the media analysis phase, the forensic examiner can take a thorough look at the media, extracting the unallocated data in order to recover any deleted files.

Searching for the slack space (in the Windows environment) or looking in the unallocated data space could reveal many file fragments that picked together could represent the clue of the file deletion activity; the knowledge of the file deletion time is also an important information to correlate the attacker activity.

1.7 Stage 7: Strings search

With this deep knowledge of the system, the analyst can now begin searching for specific strings contained inside files to reveal useful information like IP and email addresses tracing back to the attacker. A list of standard “dirty words” (e.g. hack, rootkit, mail, victim and any other word correlated to the previous findings) could be very useful to pull out relevant information about the compromise of the system.

1.8 Stage 8: Reporting

All the above stages need to be reported in a detailed fashion with a language that can be easily understood by non-technical people. The forensic analyst must make sure he explains the evidence he found very clearly, together with the techniques and the methodology he used.

Some tools feature a reporting function that could be very useful to keep track of the actions done during the forensic analysis process.

2 Compromised server forensic analysis process

This document section details the whole forensic analysis process of the system which is the target of the investigation. Some computer screenshots help to better illustrate the operating context. It is assumed the forensic analyst had been previously notified by computer management personnel, which experienced an abnormal system behaviour.

Although several commercial software applications are available today for forensic purposes, the use of Open Source software tools is still preferred for its flexibility and wide community support.

2.1 Verification phase

Before starting to operate, the forensic examiner collects as much information as he can by interviewing system administrators team and users. The present scenario reflects a situation in which the sysadmins may have had a lot of hard work to do with headquartered central servers and became less involved with peripheral servers, originally supplied to a small number of users belonging to a branch office.

The information gathered from the sysadmins reveals the computer under analysis currently acts as a file server to a limited user group; it is domain networked in a branch office subnet and grew unpatched without the latest updates at the antivirus software. Then some users complained about loosing important files: the clues of the computer compromise may be found on the following screenshots.

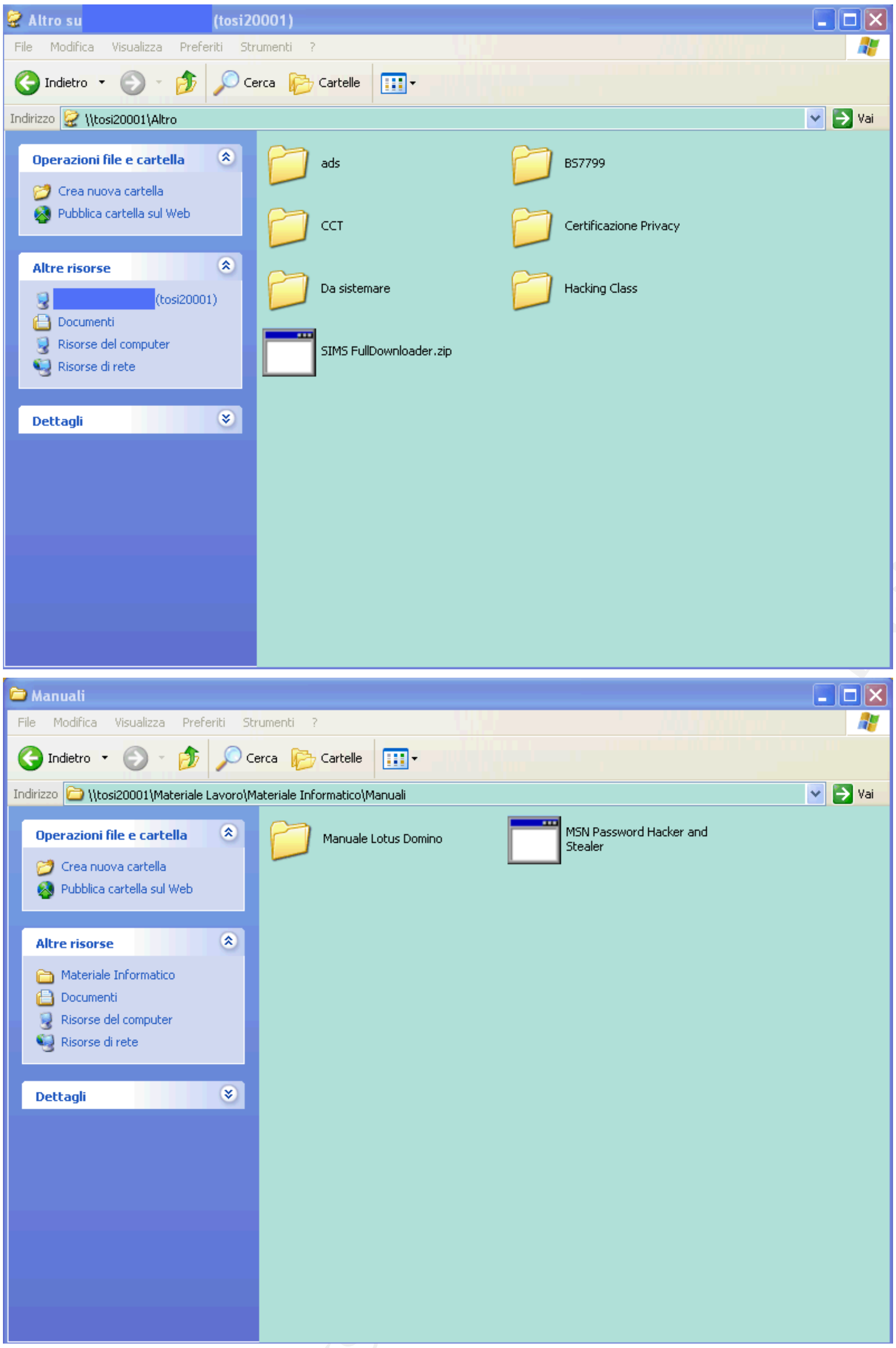


Figure 2 - Compromised system clues

It is quite normal to be doubtful of files named SIMS FullDownloader.zip and MSN Password Hacker and Stealer into the shared users area.

Taking a quick search on the Internet for such file names reveals they are file dropped on the system from LovGate worm family; a detailed technical description of the worm characteristics can be found on the document appendix[3].

To have a confirmation of the worm infection, the forensic analyst could scan the system against an updated antivirus software but

[3] Lovgate_g_description.rtf

this would modify the files access times, therefore losing some important findings.

The forensic examiner could perform an audit to the system log files to get more clues but the problem would be the same as above, it is therefore imperative to begin the data acquisition phase in order to get the clues by analysing directly the system image on a dedicated forensic workstation.

In this case, the forensic analyst has to work on a live system: this is the best situation, so he could take advantage of the volatile information (memory, processes, pagefile, network connections) but he need to be very careful pulling out the data without changing the system state.

2.2 System description phase

Once the forensic examiner has got some evidences of the system compromise he needs to get as much computer information as possible: this can be accomplished by observing the computer in its physical site (system identification), asking questions to users and system administrator and picking up system information by running forensic tools (previously recorded on a trusted CDROM).

This last step prevents the execution of system binaries that could be trojanized getting out false information or making the system instable.

2.2.1 Beginning the analysis

The analysis activities started at 16:15 GMT+1, on December, 13th 2005

The server is a Compaq Deskpro Tower, serial number # NL21112293.

The server FQDN is TOSI20001.it.XXXX.YYYY.com.

The server fixed IP Address is in the private class range 10.8.20.200/24.

The system date and time are right: no time zone and clock skew adjustment required.

The last two information were pulled out by executing a cmd.exe shell (run as administrator) from the forensic analyst trusted CDROM; all of the next data acquisition commands have to be executed in the same way.

The best and quickest method to acquire the data from the compromised server would be connect an external hard disk drive by a communication device port (USB, fire wire, SCSI).

The forensic examiner didn't have it available so he pulled out the data by network shares on the forensic workstation (E:\WFT\ for the WFT output, E:\memdump\ for the memory image in Windows mode and /images/windowsforensics/ in Linux mode for the hard disk partition images).

The forensic workstation FQDN is TOS20005.it.XXXX.YYYY.com and the DHCP assigned IP address is 10.8.11.186 in Windows Mode (Running XP Pro SP2); the IP address in Linux Mode (Fedora Core 3 - hostname Linuxforensics) was 10.8.11.144.

The data acquisition environment network diagram is depicted in the following page picture.

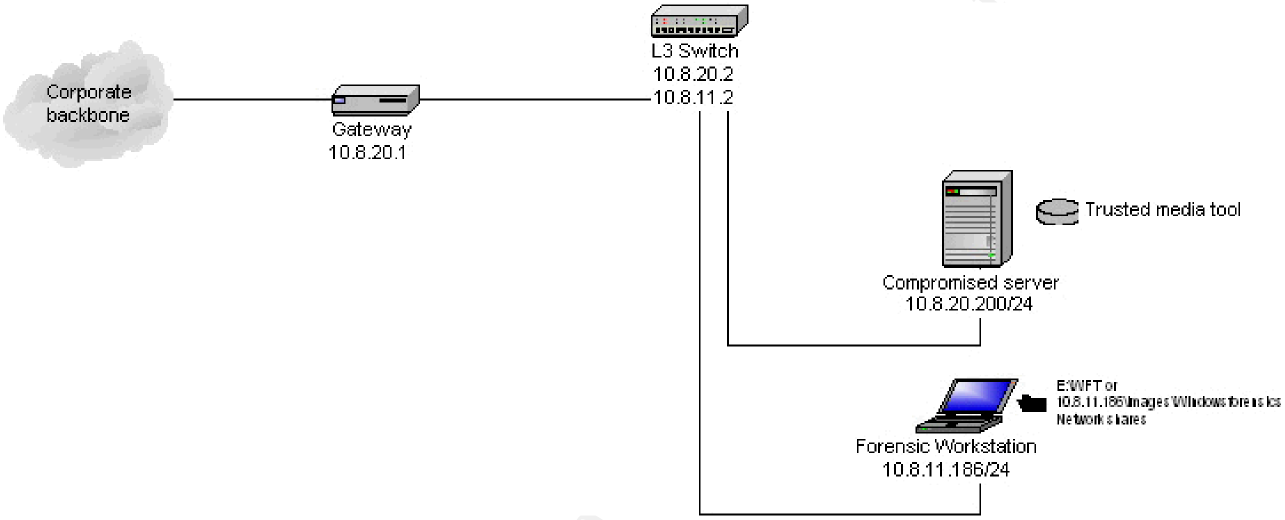


Figure 3 - Forensic analysis data acquisition diagram

2.2.2 Collecting system info by WFT

This task is mainly enumeration: during this activity the forensic examiner acquires a lot of system information, this could be partially superimposed to the next phase (evidence collection of processes, network connections etc).

The Windows Forensic Toolchest[2] (WFT) is an open source program written by Monty McDougal; it a is very useful tool to collect in a automated fashion valuable information from the compromised system.

It executes many commands in a forensic sound manner, checking them by calculating their MD5 value before any execution and logging all the activities without altering the evidence media.

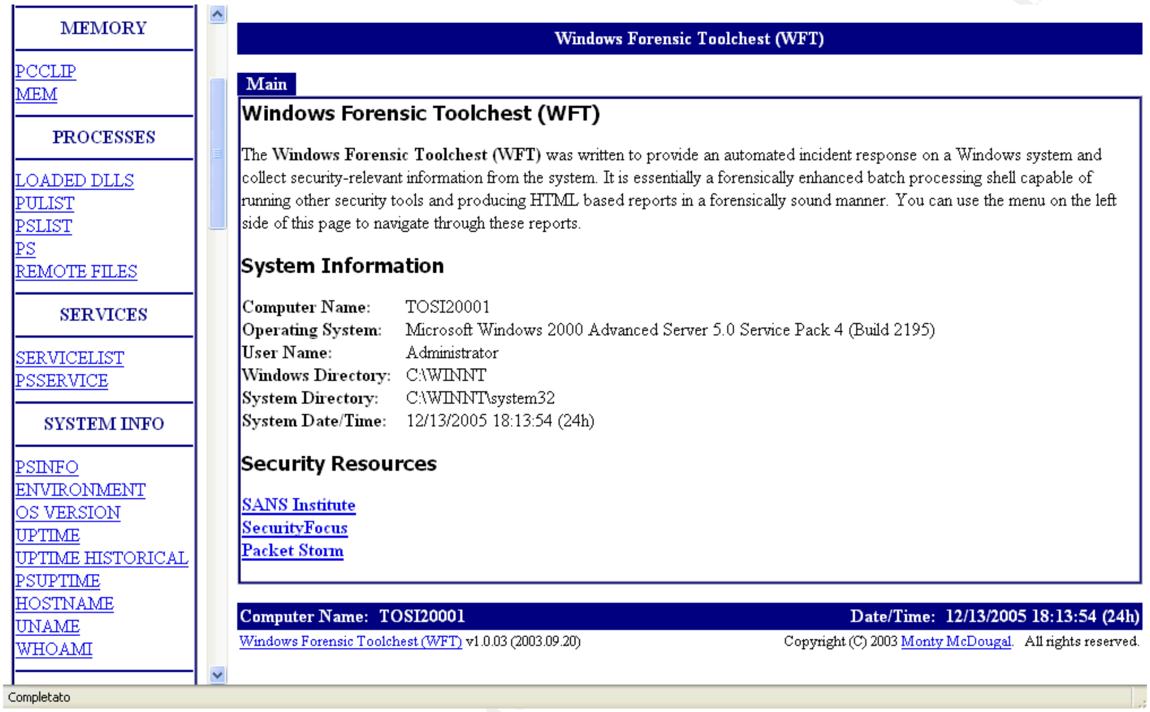
The output report is produced either in html pages or in text mode and could give the examiner a lot of useful information (system info, processes running, network connections, system users and groups, last used files, last changed registry keys and many others).

The forensic analyst run that tool, on the compromised server with the command:

```
Dos Prompt >  D:\wft -dst \\tos20005\wft\
```

The WFT started his execution on December, 13th at 17:53:13: it produced the wft log file[4] which is available in the document appendix.

The index page of the html report follows:



Windows Forensic Toolchest (WFT)

Windows Forensic Toolchest (WFT)

The Windows Forensic Toolchest (WFT) was written to provide an automated incident response on a Windows system and collect security-relevant information from the system. It is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner. You can use the menu on the left side of this page to navigate through these reports.

System Information

Computer Name:	TOSI20001
Operating System:	Microsoft Windows 2000 Advanced Server 5.0 Service Pack 4 (Build 2195)
User Name:	Administrator
Windows Directory:	C:\WINNT
System Directory:	C:\WINNT\system32
System Date/Time:	12/13/2005 18:13:54 (24h)

Security Resources

SANS Institute
SecurityFocus
Packet Storm

Computer Name: TOSI20001 Date/Time: 12/13/2005 18:13:54 (24h)

Windows Forensic Toolchest (WFT) v1.0.03 (2003.09.20) Copyright (C) 2003 Monty McDougal. All rights reserved.

Figure 4 - Index page of WFT

A thorough read at the WFT report pages confirms the previous suspects: the system event viewer logged many files which were infected by the W32/Lovgate_g@M worm; a look at the registry search history gave more clues: presence of virus dropped file names in the Software\Microsoft\Internet Explorer\Explorer Bars - FilesNamedMRU section.

Another suspicious system trait is the fact that currently the file server has the network TCP ports 25 and 80 open to any external connections (from the WFT netstat report). As gathered during the interviews with users and sysadmins, the system sole purpose was to serve files, not to send emails or publishing webpages.

[4] wft.log

2.3 Evidence collection phase

The process collecting evidence from a machine running the Windows OS is depicted in the following flow-chart:

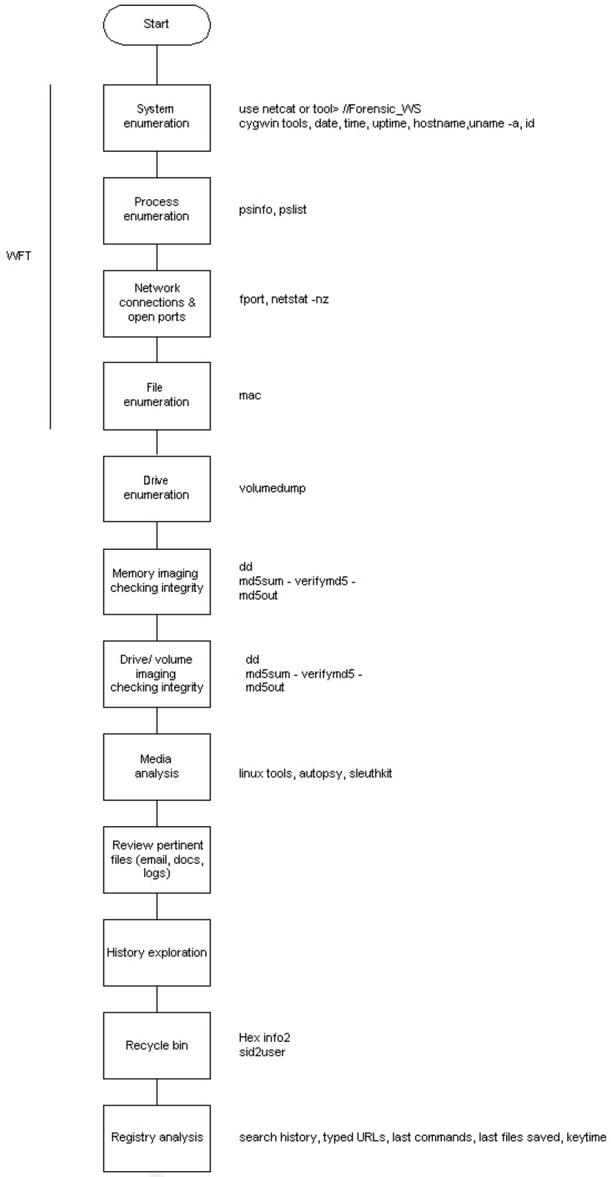

Start
System enumeration
use netcat or tool> //Forensic_WS
cygwin tools, date, time, uptime, hostname,uname -a, id
Process enumeration
psinfo, pslist
WFT
Network connections & open ports
fport, netstat -nz
File enumeration
mac
Drive enumeration
volumedump
Memory imaging checking integrity
dd
md5sum - verifymd5 - md5out
Drive/ volume imaging checking integrity
dd
md5sum - verifymd5 - md5out
Media analysis
linux tools, autopsy, sleuthkit
Review pertinent files (email, docs, logs)
History exploration
Recycle bin
Hex info2
sid2user
Registry analysis
search history, typed URLs, last commands, last files saved, keytime

Figure 5 - Evidence collection process

The first section of the diagram has already been covered in the previous paragraph: the forensic analyst copied, via the WFT tool, the more volatile information to the forensic workstation.

In order to collect the whole memory dump the examiner has to use another Unix tool, ported to the Windows environment; it is the dd tool able to copy memory, files, directory, disk partitions or physical devices in a bit by bit mode[5] regardless of the contents.

2.3.1 Collecting memory dump

The memory size of the compromised server is 256 MB; it was dumped to the forensic workstation share E:\memdump\ entering the

[5] Really the tool copies block by block, depending on the specified options

command:

```
Dos Prompt > dd if= \\.\PhysicalMemory[6] of= \\tos20005\Memdump\mem.img --md5sum --verifymd5 --
md5out= \\tos20005\Memdump\mem.img.md5 conv=noerror
```

To verify the integrity of the data memory dumped, the md5 commands were added: the memory md5 hash is written on the standard output, then it is written on a file on the target machine and finally the MD5 hash is computed on the memory dumped file and checked backwards with the one calculated at the end of the memory dump process.

The hash calculated and verified was the following:

```
\7be085618a6d68c76f09883fd56a58a2
[\\\\.\\PhysicalMemory]*\\\\tos20005\\Memdump\\mem.img
```

The data is now available onto the forensic workstation to better understand what happened on the compromised server.

In order to have more clues, it was run the (Unix ported to Windows) strings command against the memory image file: the command, in his default execution mode extracts from that raw file all of the ASCII strings whose length is more than 4 characters. Finally it is possible to parse the string output file with the grep command against the words tipically used by hackers.

The "dirty words list" contained these strings: 163.com, attack, badmail, bomb, fake, flood, ftp, hack, I-WORM, mail, mail-ru, passw, root, shellcode, smtp, Trojan, victim, virus, worm, xploit.

The search results were inserted in a spreadsheet[7] in order to track the scripts, virus names, strings, email and IP addresses in support for further examination. Such file can be found in the document appendix.

2.3.2 Logical volumes information

To have a more precise idea on how the hard disks media were organized at the installation time, in order to plan the next disk imaging activities, it was run the volume_dump utility written by George Garner[3].

The program output prints out some useful media information: volume name, volume label (if exists), mount points, drive type, serial number, volume characteristics, file system and disk number.

The utility was run entering the following command:

```
Dos Prompt > Volume_dump > \\TOS20005\Memdump\volume_dump.txt
```

[6] The \\.\ string identifies the current machine

[7] Dirty_words.xls

The Volume_dump output file[8] is in the appendix.

Analysing the volume_dump output file it is now possible to fill the following table with the disk media organization info:

Disk #	Serial #	Mount point	File system	Starting offset (Byte #)	Extent Length (Bytes)	End of Partition (Byte #)	Partion Size (Gbytes)
0	140041511	C:\	NTFS	32256	3700376064	3700408320	3,45
0	1355578721	D:\	NTFS	3700440576	14716445184	18416885760	13,71
0	2978877375	E:\	FAT	18416918016	1602445824	20019363840	1,49
1	2017617590	F:\	NTFS	32256	15358984704	15359016960	14,30
2	224860127	G:\	FAT32	32256	5765889024	5765921280	5,37
2	1004013017	I:\	FAT	5848206336	2113864704	7962071040	1,96
2	807475203	(field service reserved)	FAT32	7962071040	2146798080	10108869120	2,04

Table 1 - Hard disks partitions

2.3.3 Logical drives images acquisition

Starting from the information in the previous table, it's now possible to plan the acquisition of the disk logical volume images; even tough it could be possible to collect the whole physical disks images, the forensic analyst chooses to collect the logical volumes separately and combine them later on.

The logical images acquisition took advantage of some Linux based commands – dd and md5; the partition images data were sent to a network share of the Linux forensic workstation with the following command:

```
Dos Prompt > dd if=\\.\C: of=\\10.8.11.186\images\windowsforensics\disco_c.img --md5sum --
verifymd5 --md5out=\\10.8.11.144\images\windowsforensics\disco_c.img.md5
```

The same collection process has been repeated for all of the other hard disks logical partitions.

The hash calculated and verified were the following:

```
\58765abb9a2274824a5f846b4df745eb [\\\\.\\c:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_c.img

\1ff40fc272158b7e0add565a3a33d500 [\\\\.\\d:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_d.img

\7ac93b47f6f03cc04d11111e0e7846e7 [\\\\.\\e:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_e.img
\bb62fdc57174d48552a04d39eee584ae [\\\\.\\f:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_f.img
```

[8] Table 4 – Volume_dump utility output

```
\5cce25f2c0648d46d4930bda51532844 [\\\\.\\g:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_g.img

\3bf5de59ef693a6f12d19416da5ace24 [\\\\.\\i:]
*\\\\10.8.11.144\\images\\windowsforensics\\disco_i.img
```

Such activity finally terminated the data acquisition process.

2.4 Media analysis phase

The media analysis task starts once all of the compromised server information is available in the forensic workstation.

As stated in the previous chapter Linux was used as favourite operating system to analyse the media. Such activity relies mainly on the Sleuth Kit[4] tool written by Brian Carrer, a very specialized forensic suite made up of sixteen tools able to analyse thoroughly the disk media (according to the Unix style based on simple programs doing their task very well).

Handling large images (several Gbytes of data and hundred thousands of files) could be a daunting task with the command line interface provided by the Sleuth Kit; to solve the situation this tool was combined with the web based Graphical User Interface supplied from the Autopsy Forensic Browser, coming from the same author.

The Sleuth Kit version was the 1.72.

The Autopsy Forensic Browser version was the 2.03.

2.4.1 Autopsy Forensic Browser analysis process

The following figure depicts the analysis process performed by the Autopsy Forensic Browser tool:

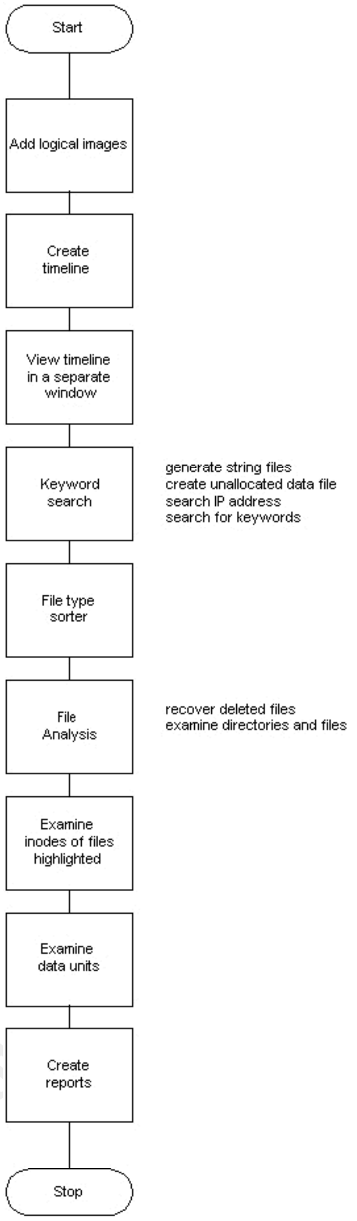


Figure 6 - Autopsy Forensic Browser verification process

Once the Autopsy application start up is completed, a few more steps must be followed to create a basic analysis information:

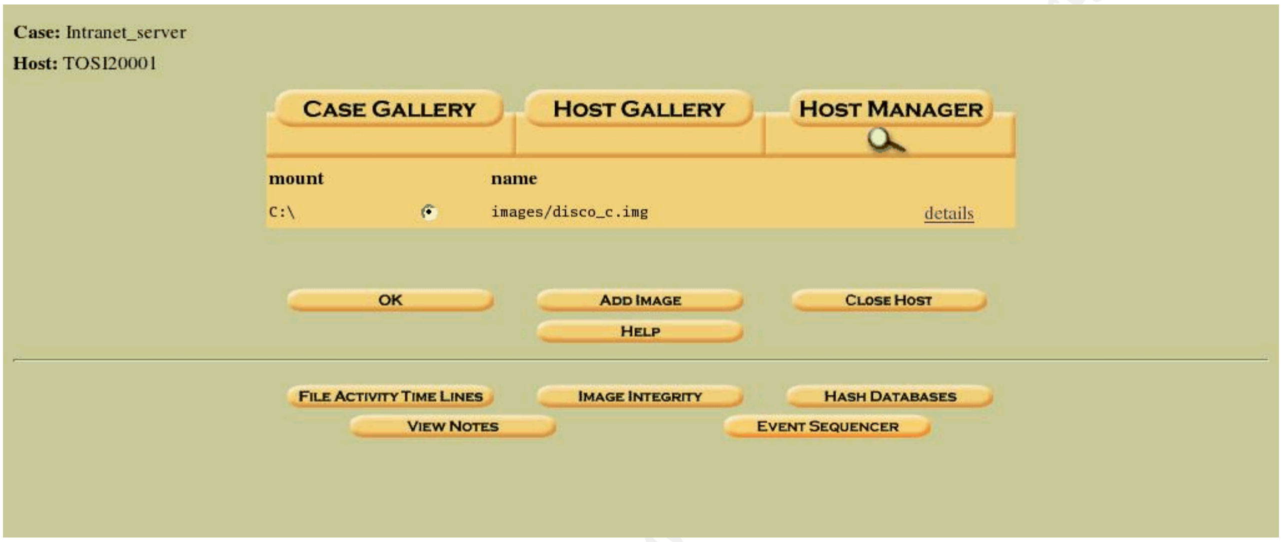
- a new case has to be opened;
- the investigation details have to be filled in (investigator's name, case description);
- the host gallery is now opened: the investigator can add the host target of the analysis and fill in pertinent

information;

- finally, the examiner must add (with symbolic links) the volume images previously copied to the forensic workstation checking their MD5 hash value.

The program creates the configuration files and directories: the media analysis task can now begin.

An Autopsy Forensic Browser screenshot about the "Intranet Server" case is depicted in the following picture:



Case: Intranet_server

Host: TOSI20001

mount		name	
C:\	●	images/disco_c.img	details

Figure 6 - Autopsy Forensic Browser screenshot

The program GUI interface is very user-friendly, the magnifier lens points to the actual program's feature; it is now possible by many pushbuttons to activate specific tools.

The "File Activity Time Lines" button, placed in the left bottom area, was used to obtain the timeline file described in the following paragraph.

2.5 Timeline creation and analysis phase

This activity is very useful to understand what happened on the compromised system from a temporal perspective.

The process is divided in two phases: in the first one all of the MAC times (belonging to either the image data or the metadata structures of the image) are pulled out and stored to an intermediate body file; note it could be possible now to merge data coming from different images or from external security devices like firewalls or IDSes.

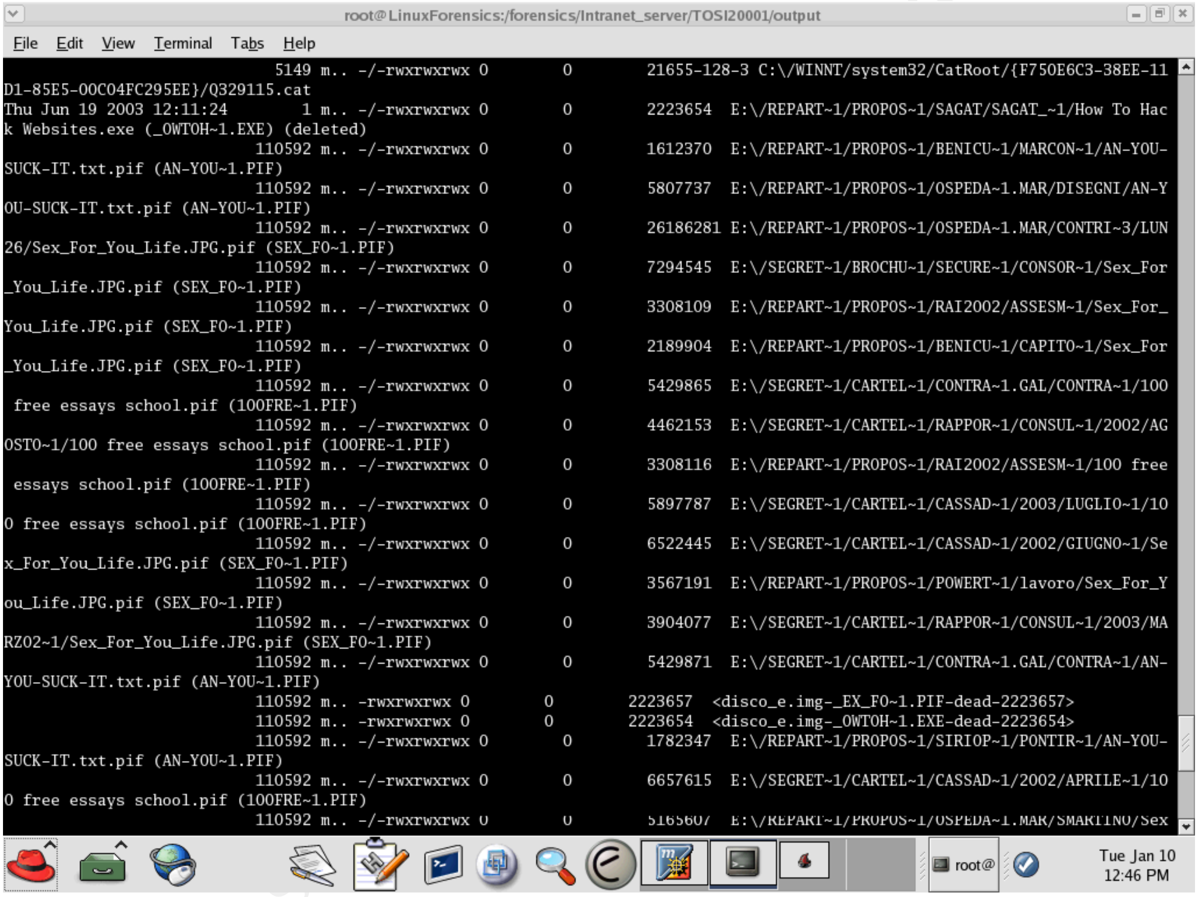
The examiner produced the body file of the timeline putting together the data coming from different images.

The Sleuth Kit commands run in this task were:

```
fls -r -m     ->     list both the allocated and the deleted files, recurse on directories
(-r), display the output in timeline import format (-m)
ils -m        ->     list inodes information in mactime mode (-m)
```

The second part of the timeline creation sorts the information in date ascending order: it is also possible to select a starting and ending date and other parameters to better refine the analysis task. Into the timeline analysis phase, it is a good idea to correlate such data with the "uptime historical" output supplied by the WFT tool.

An extract of the timeline file is depicted in the following picture:



root@LinuxForensics:/forensics/Intranet_server/TOSI20001/output
File Edit View Terminal Tabs Help
root@
Tue Jan 10
12:46 PM

```
                           5149 m.. -/-rwxrwxrwx 0        0        21655-128-3 C:\/WINNT/system32/CatRoot/{F750E6C3-38EE-11
D1-85E5-00C04FC295EE}/Q329115.cat
Thu Jun 19 2003 12:11:24        1 m.. -/-rwxrwxrwx 0        0        2223654  E:\/REPART~1/PROPOS~1/SAGAT/SAGAT_~1/How To Hac
k Websites.exe (_OWTOH~1.EXE) (deleted)
                         110592 m.. -/-rwxrwxrwx 0        0        1612370  E:\/REPART~1/PROPOS~1/BENICU~1/MARCON~1/AN-YOU-
SUCK-IT.txt.pif (AN-YOU~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        5807737  E:\/REPART~1/PROPOS~1/OSPEDA~1.MAR/DISEGNI/AN-Y
OU-SUCK-IT.txt.pif (AN-YOU~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        26186281 E:\/REPART~1/PROPOS~1/OSPEDA~1.MAR/CONTRI~3/LUN
26/Sex_For_You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        7294545  E:\/SEGRET~1/BROCHU~1/SECURE~1/CONSOR~1/Sex_For
_You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        3308109  E:\/REPART~1/PROPOS~1/RAI2002/ASSESM~1/Sex_For_
You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        2189904  E:\/REPART~1/PROPOS~1/BENICU~1/CAPITO~1/Sex_For
_You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        5429865  E:\/SEGRET~1/CARTEL~1/CONTRA~1.GAL/CONTRA~1/100
 free essays school.pif (100FRE~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        4462153  E:\/SEGRET~1/CARTEL~1/RAPPOR~1/CONSUL~1/2002/AG
OSTO~1/100 free essays school.pif (100FRE~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        3308116  E:\/REPART~1/PROPOS~1/RAI2002/ASSESM~1/100 free
 essays school.pif (100FRE~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        5897787  E:\/SEGRET~1/CARTEL~1/CASSAD~1/2003/LUGLIO~1/10
0 free essays school.pif (100FRE~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        6522445  E:\/SEGRET~1/CARTEL~1/CASSAD~1/2002/GIUGNO~1/Se
x_For_You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        3567191  E:\/REPART~1/PROPOS~1/POWERT~1/lavoro/Sex_For_Y
ou_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        3904077  E:\/SEGRET~1/CARTEL~1/RAPPOR~1/CONSUL~1/2003/MA
RZO2~1/Sex_For_You_Life.JPG.pif (SEX_FO~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        5429871  E:\/SEGRET~1/CARTEL~1/CONTRA~1.GAL/CONTRA~1/AN-
YOU-SUCK-IT.txt.pif (AN-YOU~1.PIF)
                         110592 m.. -rwxrwxrwx 0        0        2223657  <disco_e.img-_EX_FO~1.PIF-dead-2223657>
                         110592 m.. -rwxrwxrwx 0        0        2223654  <disco_e.img-_OWTOH~1.EXE-dead-2223654>
                         110592 m.. -/-rwxrwxrwx 0        0         1782347  E:\/REPART~1/PROPOS~1/SIRIOP~1/PONTIR~1/AN-YOU-
SUCK-IT.txt.pif (AN-YOU~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        6657615  E:\/SEGRET~1/CARTEL~1/CASSAD~1/2002/APRILE~1/10
0 free essays school.pif (100FRE~1.PIF)
                         110592 m.. -/-rwxrwxrwx 0        0        5165607  E:\/REPART~1/PROPOS~1/OSPEDA~1.MAR/SMARTINO/Sex
```

Figure 7 - Timeline format file

The timeline lists information in the following format, from the leftmost column to the right direction: File size (Bytes); last mac action (m=saved, a=read,executed, c=created,inode allocated); file permissions, GID, UID, inode number, file or inode specification.

While examining the timeline, the forensic analyst had the confirmation the system had been infected by the **W32/Lovgate.g@M**

worm. The evidences were many file entries named as follows:

- MSN Password Hacker & Stealer
- Winrar + crack.exe
- The world of lovers.txt.exe
- Sex for your life
- Panda Titanium
- Age of Empires
- Are you looking for love.doc.exe
- Mafia Trainer!!!.exe
- MoviezChannelsIsInstaller.exe
- Star Wars II Movie Full Downloader.exe
- How To Hack Websites.exe
- CloneCD +Crack.exe
- Autoexec.bat
- 100 Free Essays Schools.pif

The larger number of these files was found on the system shared logical drives as confirmation of the spreading method stated by the antivirus software researchers.

A common trait of such files is they were written/executed and then immediately deleted; recording the inode number that appears in the timeline makes it possible to further analyse the content of the file.

The first evidence of the infected file dates back on May, 23rd 2003 on partition C, then the virus spreading activity was very heavy, until the end of 2003; on December, 10th 2003 (as reported from WFT) a new antivirus software installation was performed (McAfee instead of Norton). Starting from this date the number of infected files started to decrease, but some worm activity was still observed; such events last about one day, then the infection hushed and offered a period of quietness.

A summary timeline table[9] follows in the next page:

Date	**Time**	**Event**
14/05/2002	15:16:35	OS installation

[9] The whole timeline file is available separately

23/05/2003	**16:18:20**	**First LovGate_G evidence (Starwars II ….dropped) on C partition**
18/06/2003	12:14:09	Hotfix KB823182 installed
19/06/2003	**12:11:24**	**First LovGate_G files (*.pif 110592 bytes length) dropped on E partition**
19/06/2003	21:05:04	Windows Service Pack files copied
27/06/2003	13:55:15	Hotfix KB823559 installed
05/07/2003	10:16:54	Hotfix KB823980 installed
16/07/2003	**15:25:40**	**Executed the above Starwars II file**[10]
05/08/2003	15:14:23	Hotfix KB824146 installed
23/08/2003	14:44:56	Hotfix KB824146 installed
27/08/2003	15:11:23	Hotfix KB825119 installed
02/09/2003	11:48:32	Windows SP Installation
07/10/2003	16:05:53	Hotfix KB826232 installed
23/10/2003	14:21:38	Hotfix KB828035 installed
23/10/2003	15:40:56	Hotfix KB928749 installed
06/11/2003	**00:00:00**	**Worm flood on shared partitions**
21/11/2003	**00:00:00**	**Worm flood on shared partitions**
01/12/2003	**00:00:00**	**Worm flood on shared partitions**
10/12/2003	12:03:30	Last Norton AV scan
10/12/2003	14:48:25	McAFee Netshield AV installation
15/12/2003	**00:00:00**	**Worm flood on shared partitions**
23/12/2003	**00:00:00**	**Worm flood on shared partitions**
07/01/2004	09:31:03	System memory dump
12/02/2004	**17:29:16**	**Worm flood on shared partitions**
14/05/2004	**09:28**	**Worm flood on shared partitions**
17/05/2004	**10:18:34**	**Worm flood on shared partitions**
07/06/2004	10:25:31	Updated AV; daily scan enabled
09/03/2005	**00:00:00**	**Worm flood on shared partitions**
22/04/2005	**14:29:56**	**Worm flood on shared partitions**
01/08/2005	**10:13:08**	**Worm flood on shared partitions**
24/11/2005	**15:13:32**	**Worm flood on shared partitions**

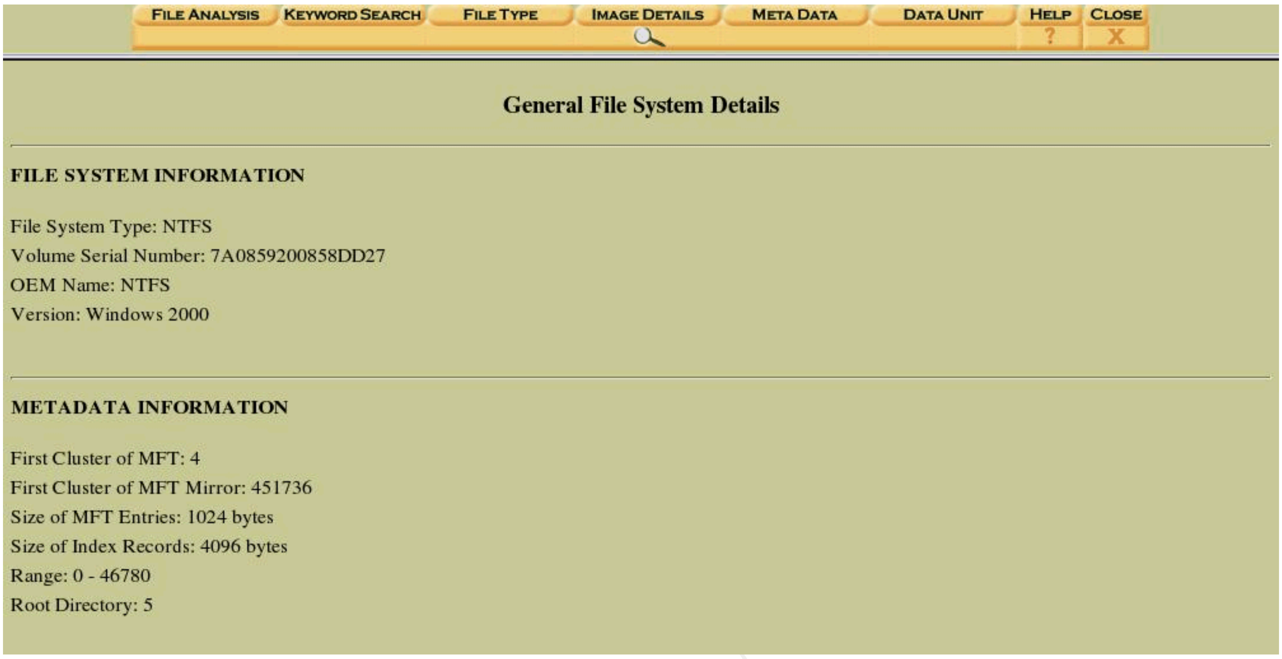
Table 2 - Summary timeline

Once the timeline examination is over, the media analysis task can start, looking for more evidence of the server's compromise.

2.6 File system analysis

[10] Beginning of the virus spread

The forensic examiner has to choose the image he wants to analyse (by selecting the "details" button); in the next screenshot he chose the file system tab and the Sleuth Kit fsstat tool run. The command output is partially depicted in the following picture:



General File System Details

FILE SYSTEM INFORMATION

File System Type: NTFS
Volume Serial Number: 7A0859200858DD27
OEM Name: NTFS
Version: Windows 2000

METADATA INFORMATION

First Cluster of MFT: 4
First Cluster of MFT Mirror: 451736
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 46780
Root Directory: 5

Figure 8 - File system details

Some important information about the file system disk partitions organization are displayed:

- C partition
  - NTFS, Windows 2000 type,
  - The data cluster is made of 8 sectors so it is 4096 bytes long, the sector ranges from 0 to 7227295, the cluster ranges from 0 to 903411,
  - The metadata range from 0 to 46780, MFT0 start from cluster 4, MFT1 start from cluster 451736.
- D partition
  - NTFS, Windows 2000 type,
  - The data cluster is made of 8 sectors so it is 4096 bytes long, the sector ranges from 0 to 28743056, the cluster ranges from 0 to 3592882,
  - MFT0 start from cluster 4, MFT1 start from cluster 1796442.
- E partition

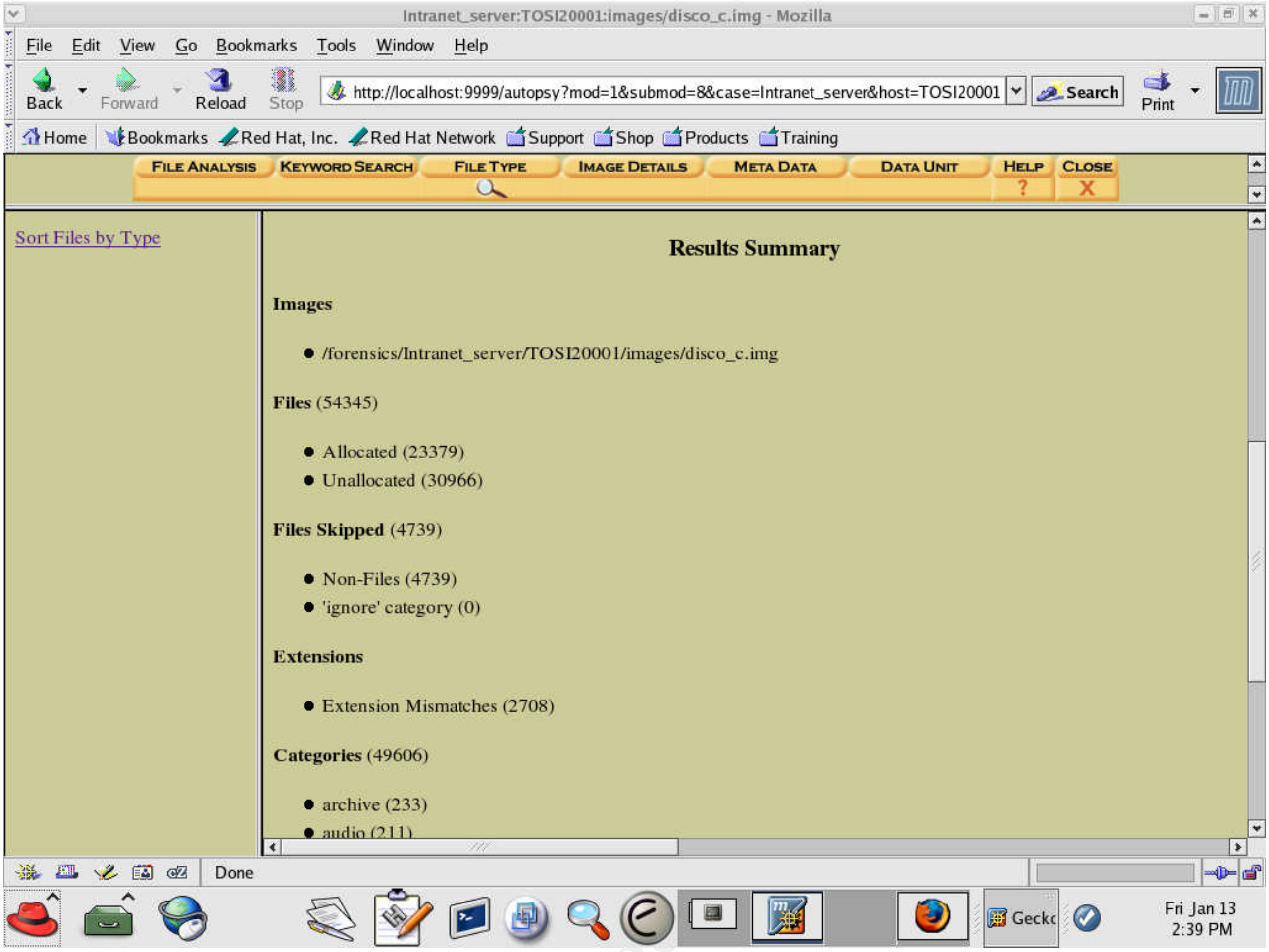
    - FAT16 type,
    - The data cluster is made of 64 sectors so it is 32768 bytes long, the sector range from 0 to 3129776, the cluster range from 2 to 48896,
    - The metadata range from 2 to 50068482, FAT0 from 1 to 222, FAT1 from 223 to 444
- F partition
    - NTFS, Windows 2000 type,
    - The data cluster is made of 8 sectors so it is 4096 bytes long, the sector ranges from 0 to 29998016, the cluster ranges from 0 to 3749752,
    - MFT0 start from cluster 4, MFT1 start from cluster 1874876.
- G partition
    - FAT32 type,
    - The data cluster is made of 8 sectors so it is 4096 bytes long, the sector range from 0 to 11261501, the cluster range from 2 to 1404940,
    - The metadata range from 2 to 179832194, FAT0 from 32 to 11008, FAT1 from 11009 to 21895
- I partition
    - FAT16 type,
    - The data cluster is made of 64 sectors so it is 32768 bytes long, the sector range from 0 to 4128641, the cluster range from 2 to 64502,
    - The metadata range from 2 to 66049026, FAT0 from 1 to 255, FAT1 from 256 to 510

Starting from the previous screenshot, the Autopsy program gives a few options for conducting the investigation in more detail, based on the appropriate file system layer (file name, metadata, cluster).

Selecting the File Type tab the Sleuthkit file sorter tool runs, testing the file header; the output yields a more precise picture about:

- media files number and categories;
- the number of unallocated files (link to the MFT or to the FAT deleted);
- the number of files with the wrong name extension respect to the file header section;
- the files are grouped in the summary according to predefined categories.

The screenshot with the output of this task is depicted in the picture on the following page:


Intranet_server:TOSI20001:images/disco_c.img - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:9999/autopsy?mod=1&submod=8&case=Intranet_server&host=TOSI20001
Search Print
Home Bookmarks Red Hat, Inc. Red Hat Network Support Shop Products Training
FILE ANALYSIS KEYWORD SEARCH FILE TYPE IMAGE DETAILS META DATA DATA UNIT HELP CLOSE
Sort Files by Type
Done
Fri Jan 13
2:39 PM

Results Summary

Images

- /forensics/Intranet_server/TOSI20001/images/disco_c.img

Files (54345)

- Allocated (23379)
- Unallocated (30966)

Files Skipped (4739)

- Non-Files (4739)
- 'ignore' category (0)

Extensions

- Extension Mismatches (2708)

Categories (49606)

- archive (233)
- audio (211)

Figure 9 - File sorter screenshot

The large number of unallocated files (30966 versus 23379 allocated) depends on the virus activity (files created/deleted), on the antivirus task (files quarantined) and from the long time elapsed (more than 3 years) since the system software installation. It is further present a large number of files (2708) to check for extension mismatches.

Having a “known good file hash database” fulfilled when the host was created could have been a good idea to reduce the analysis task effort, but it was currently unavailable.

2.7 Strings search

Several keyword searches during this step were performed either in the unallocated space or in the whole images to collect findings of the system compromise. The Autopsy search tool features are based both on ASCII and Unicode patterns, some keywords from previous memory searches were used.

2.8 Extract unallocated data space from the image

Before starting the string search phase the unallocated space extraction from the image has to be performed: this is very important because hidden information not accessible by the standard file system tools can be searched for.

The extract unallocated pushbutton was selected from search string tab; when the extract unallocated process is terminated, it is possible to conduct string searches both on the original image and on the unallocated space.

2.9 Data units, inodes and file analysis

The reverse process beginning from the string search results to recover the file contents could be summarized as follows (clicking on Autopsy hyperlink launch the specialized Sleuth Kit tool):

- in the search strings results html page (unallocated space) a data unit number was found;
- pressing the data unit tab displayed its content on the screen (via the dcat tool);
- if any interesting evidences were found the program calculated the original data cluster (via the dcalc tool) pressing the load original pushbutton;
- upon cluster content verify (dcat & strings) it was picked out the Autopsy Hex Report;
- the program run the ifind tool to search for the inode number;
- the istat tool (inode details) was run against the inode number to obtain the inode statistics;
- then the ffind tool was run to map from inode to file name
- finally the icat tool was run to recover, if available, the whole deleted file.

Some results of the file system searches[11] are summarized in the table 5 in the document appendix:

Looking at the table some information can be deducted:

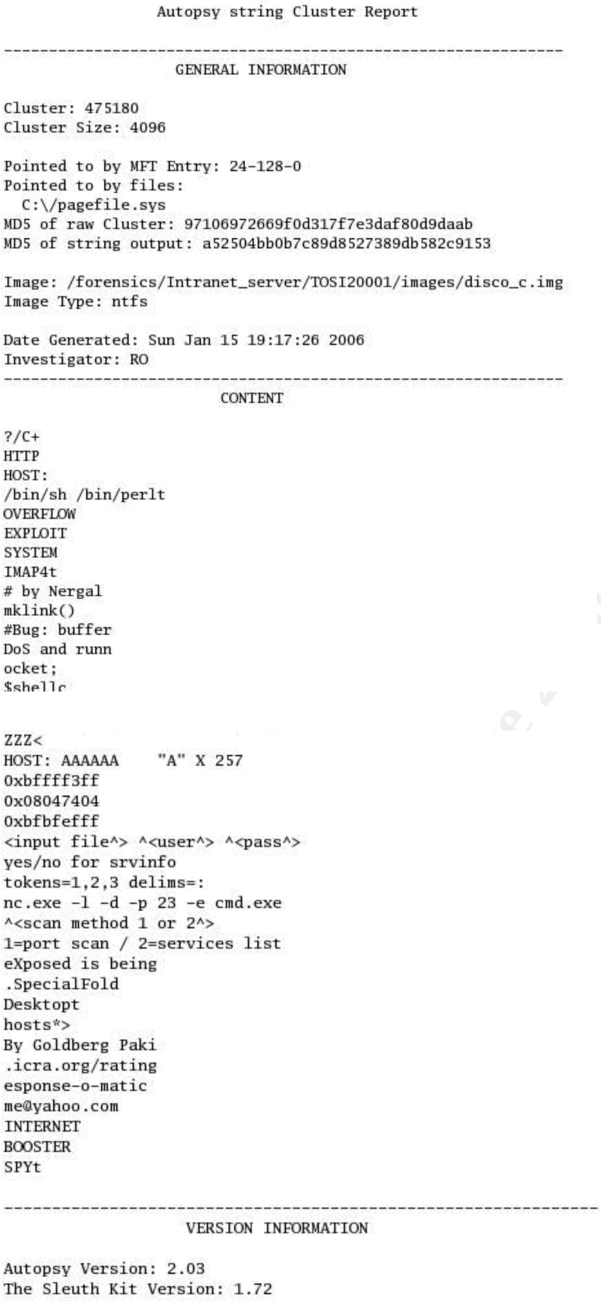
- on January, 7th 2004 there was a system crash and the memory content (with a lot of hacking scripts) was copied to the memory dump file;
- the virus activity was heavy from December, 1st (virus files

[11] This table includes only a subset of possible string searches due to the large spread of the worm files.

dropping) to the next day (virus file blocked from execution by the antivirus software marked with the VBN extension);

- despite the presence of the antivirus software (updated with the latest definitions on June, 7th 2004) no effective action was taken to eliminate the virus spreading from the system; this is highlighted from the presence of hacking script code in the pagefile.sys file created a few hours before performing the media images.

A sample screenshot with detailed information about the cluster # 475180 strings content is depicted in the following picture:

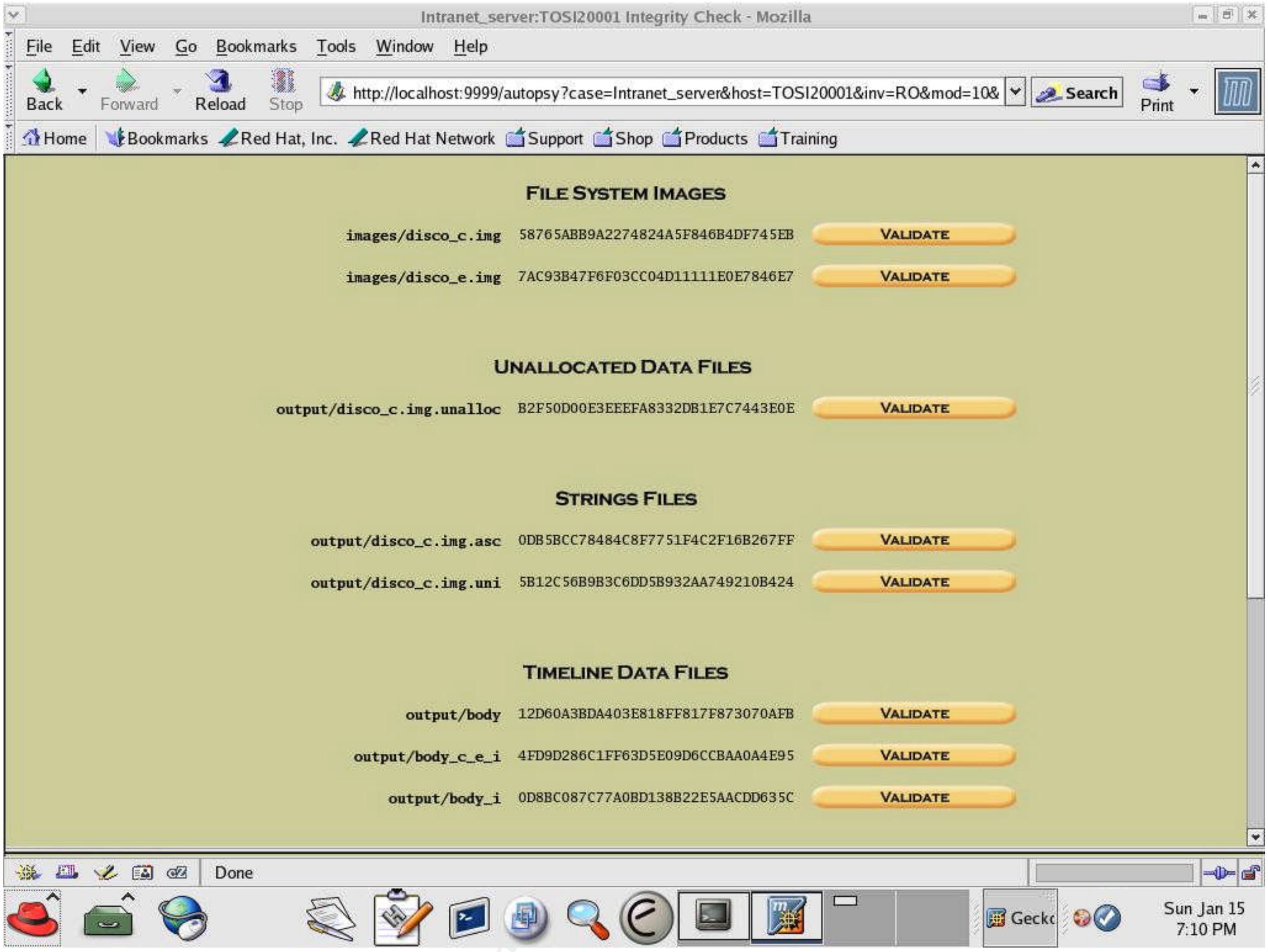


```
              Autopsy string Cluster Report

---------------------------------------------------------------
                  GENERAL INFORMATION

Cluster: 475180
Cluster Size: 4096

Pointed to by MFT Entry: 24-128-0
Pointed to by files:
  C:\/pagefile.sys
MD5 of raw Cluster: 97106972669f0d317f7e3daf80d9daab
MD5 of string output: a52504bb0b7c89d8527389db582c9153

Image: /forensics/Intranet_server/TOSI20001/images/disco_c.img
Image Type: ntfs

Date Generated: Sun Jan 15 19:17:26 2006
Investigator: RO
---------------------------------------------------------------
                        CONTENT

?/C+
HTTP
HOST:
/bin/sh /bin/perlt
OVERFLOW
EXPLOIT
SYSTEM
IMAP4t
# by Nergal
mklink()
#Bug: buffer
DoS and runn
ocket;
$shellc


ZZZ<
HOST: AAAAAA    "A" X 257
0xbffff3ff
0x08047404
0xbfbfefff
<input file^> ^<user^> ^<pass^>
yes/no for srvinfo
tokens=1,2,3 delims=:
nc.exe -l -d -p 23 -e cmd.exe
^<scan method 1 or 2^>
1=port scan / 2=services list
eXposed is being
.SpecialFold
Desktopt
hosts*>
By Goldberg Paki
.icra.org/rating
esponse-o-matic
me@yahoo.com
INTERNET
BOOSTER
SPYt

------------------------------------------------------------------
                   VERSION INFORMATION

Autopsy Version: 2.03
The Sleuth Kit Version: 1.72
```

Figure 10 - Cluster 475180 search strings detail

A partial list of unallocated space string search results is in table 6 in the document appendix.

As stated in the chapter 2.4, Autopsy executes its activities in a forensic sound manner performing the MD5 calculation in all of the tasks it completes; a sample of such checksums is depicted in the following screenshot:



FILE SYSTEM IMAGES

images/disco_c.img 58765ABB9A2274824A5F846B4DF745EB VALIDATE

images/disco_e.img 7AC93B47F6F03CC04D11111E0E7846E7 VALIDATE

UNALLOCATED DATA FILES

output/disco_c.img.unalloc B2F50D00E3EEEFA8332DB1E7C7443E0E VALIDATE

STRINGS FILES

output/disco_c.img.asc 0DB5BCC78484C8F7751F4C2F16B267FF VALIDATE

output/disco_c.img.uni 5B12C56B9B3C6DD5B932AA749210B424 VALIDATE

TIMELINE DATA FILES

output/body 12D60A3BDA403E818FF817F873070AFB VALIDATE

output/body_c_e_i 4FD9D286C1FF63D5E09D6CCBAA0A4E95 VALIDATE

output/body_i 0D8BC087C77A0BD138B22E5AACDD635C VALIDATE

Figure 11 - MD5 sums calculated by forensic tools

It was a little unexpected to find nothing but the presence of the email address **@yahoo.com.cn** that was used from the hacker as recipient to be notified about the system compromise; this keyword string search on the C drive image revealed 54 hits and it took to the string:

configserversmtp.163.com#ab89d@yahoo.com.cn#yf23668@163.com

This is part of the Snort IDS signature for the LovGate worm that follows:

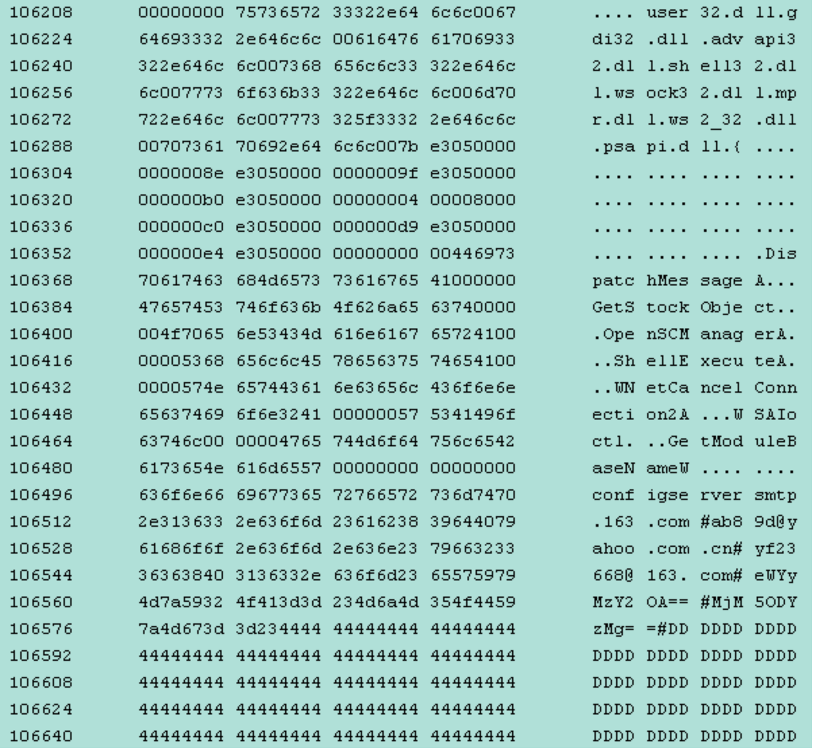
```
alert tcp any any -> any 139 ( msg: "lovgate virus"; content:
"configserversmtp.163.com#ab89d@yahoo.com"; flow:to_serve
r,established; classtype:misc-activity;)

alert tcp any any -> any 445 ( msg: "lovgate virus"; content:
"configserversmtp.163.com#ab89d@yahoo.com"; flow:to_serve
r,established; classtype:misc-activity;)

alert tcp any any -> any 445 ( msg: "lovgate virus netservices.exe";
content: "s.y.s.t.e.m.3.2.\.N.e.t.S.e.r.v.i.c.e.s.
..e.x.e"; flow:to_server,established; classtype:misc-activity;)

alert tcp any any -> any 445 ( msg: "lovgate virus netservices.exe";
content: "s|00|y|00|s|00|t|00|e|00|m|00|3|00|2|00 5C
00|N|00|e|00|t|00|S|00|e|00|r|00|v|00|i|00|c|00|e|00|s|00|.|00|e|00|x|00|e";
flow:to_server,established; classtype:misc-activity;)
```

Several searches were performed for clues about some “live file” on a C partition containing such string but all of them conducted to a *.VBN file already detected by the antivirus software.

Analysing thoroughly the virus files dropped on the E partition at the beginning of the infection process were found some of them unchecked from the antivirus software; the FAT entries #1612370 and #5807737 were analysed respectively.

- Metadata 1612370, last written on 19/06/03 12:11:24, last accessed on 23/12/03 0:00:00; sector range from 990557 to 990812 (256); filename E:\repart_1\propos_1\marcom_1\AN-YOU-SUCK.txt
- Metadata 5807737, last written on 19/06/03 12:11:24, last accessed on 23/12/03 0:00:00; sector range from 1057885 to 1058140 (256); filename E:\repart_1\propos_1\ospeda_1.mar\disegni\AN-YOU-SUCK.pif

The whole hexdump of the second file[12] ca be found in the document appendix, a partial view of the virus fingerprint is depicted in the following picture:



```
106208      00000000 75736572 33322e64 6c6c0067      .... user 32.d ll.g
106224      64693332 2e646c6c 00616476 61706933      di32 .dll .adv api3
106240      322e646c 6c007368 656c6c33 322e646c      2.dl l.sh ell3 2.dl
106256      6c007773 6f636b33 322e646c 6c006d70      l.ws ock3 2.dl l.mp
106272      722e646c 6c007773 325f3332 2e646c6c      r.dl l.ws 2_32 .dll
106288      00707361 70692e64 6c6c007b e3050000      .psa pi.d ll.{ ....
106304      0000008e e3050000 0000009f e3050000      .... .... .... ....
106320      000000b0 e3050000 00000004 00008000      .... .... .... ....
106336      000000c0 e3050000 000000d9 e3050000      .... .... .... ....
106352      000000e4 e3050000 00000000 00446973      .... .... .... .Dis
106368      70617463 684d6573 73616765 41000000      patc hMes sage A...
106384      47657453 746f636b 4f626a65 63740000      GetS tock Obje ct..
106400      004f7065 6e53434d 616e6167 65724100      .Ope nSCM anag erA.
106416      00005368 656c6c45 78656375 74654100      ..Sh ellE xecu teA.
106432      0000574e 65744361 6e63656c 436f6e6e      ..WN etCa ncel Conn
106448      65637469 6f6e3241 00000057 5341496f      ecti on2A ...W SAIo
106464      63746c00 00004765 744d6f64 756c6542      ctl. ..Ge tMod uleB
106480      6173654e 616d6557 00000000 00000000      aseN ameW .... ....
106496      636f6e66 69677365 72766572 736d7470      conf igse rver smtp
106512      2e313633 2e636f6d 23616238 39644079      .163 .com #ab8 9d@y
106528      61686f6f 2e636f6d 2e636e23 79663233      ahoo .com .cn# yf23
106544      36363840 3136332e 636f6d23 65575979      668@ 163. com# eWYy
106560      4d7a5932 4f413d3d 234d6a4d 354f4459      MzY2 OA== #MjM 5ODY
106576      7a4d673d 3d234444 44444444 44444444      zMg= =#DD DDDD DDDD
106592      44444444 44444444 44444444 44444444      DDDD DDDD DDDD DDDD
106608      44444444 44444444 44444444 44444444      DDDD DDDD DDDD DDDD
106624      44444444 44444444 44444444 44444444      DDDD DDDD DDDD DDDD
106640      44444444 44444444 44444444 44444444      DDDD DDDD DDDD DDDD
```

Figure 12 - LovGate virus fingerprint extracted from the file hexdump

3 Malware analysis

[12] Lovgate_Hexdump_File.rtf

Once the malware code has been isolated, it is possible to execute it against a virtualized host in order to have more information about the virus behaviour; to accomplish such task the malware must be run in a sterile environment while dedicated tools are deployed to monitor all the program activities.

Since such code widely spreads the effects all over the system a machine able to revert its state to a "pre-infection" configuration has to be set up.

The system virtualization application took advantage of a VMWare 5.0 virtual machine installed in an air-gapped[13] Linux based hardware with Win2K installed guest operating system; the following process steps have to be strictly followed to monitor the malware behaviour:

- Winalysis[5] was run to take a system snapshot
- Nestat tool was run to display the current TCP/UDP connections and open ports
- Regmon[6] utility was run to monitor the interactions with the system registry
- Filemon[7] utility was run to monitor the interactions with the file system
- LovGate malware code was run for about 20 seconds
- Then the LovGate process was stopped with Windows Task Manager
- Both the Regmon and the Filemon utilities were stopped
- Nestat tool was run again to display the updated TCP/UDP connections and open ports
- Winalysis was run again to take an updated system snapshot

Once completed the data collection activity the infection tracks can be deleted reverting the virtual machine to a previously saved configuration.

3.1 Code execution summary results

The Filemon analysis indicates that the LovGate.exe process loads the following system dynamically linked libraries:

wsock32.dll, ws2_32.dll, ws2help.dll, psapi.dll

Then it creates the brand new file **WinDriver.exe** in the system directory (c:\winnt\system32), then it deletes the c:\winnt\system32\config\software.log, finally it creates the hostile files **WinHelp.exe**, **winrpc.exe**, **WinGate.exe** and **RAVMOND.exe** into the system directory.

[13] No external network connections available

The Regmon analysys indicates the LovGate executable open, creates and set the value of a lot of registry keys, the most important ones are summarized below:

HKLM\System\CurrentControlSet\Services\Windows Management Instrumentation Driver Extension\ImagePath with the value "C:\WINNT\System32\WinDriver.exe –start_server"

HKLM\Software\Microsoft\Windows\CurrentVersion\Run with the value"C:\WINNT\System32\winhelp.exe…"

HKCR\txtfile\shell\open\command\(Default) with the value "**winrpc.exe %1**…"

HKLM\Software\Microsoft\Windows\CurrentVersion\Run\WinGate initialize with the value "C:\WINNT\System32\WinGate.exe -remoteshell…"

HKCU\Software\Microsoft\Windows NT\CurrentVersion\run with the value "**RAVMOND.exe**"

From the networking point of view a lot of network connections were verified established (TCP ports 139 and 445) between the Windows infected "virtualized host" and the Linux system hosting the VMWare application.

Infected machine behaviour notes: when the LovGate executable was stopped the presence of dropped files named as in the list on page 21 was observed in many directories these are the proof of the auto-copy behaviour of the worm; taking another view at the SAMBA shares on the VMWare host computer, worm replicas were found too.

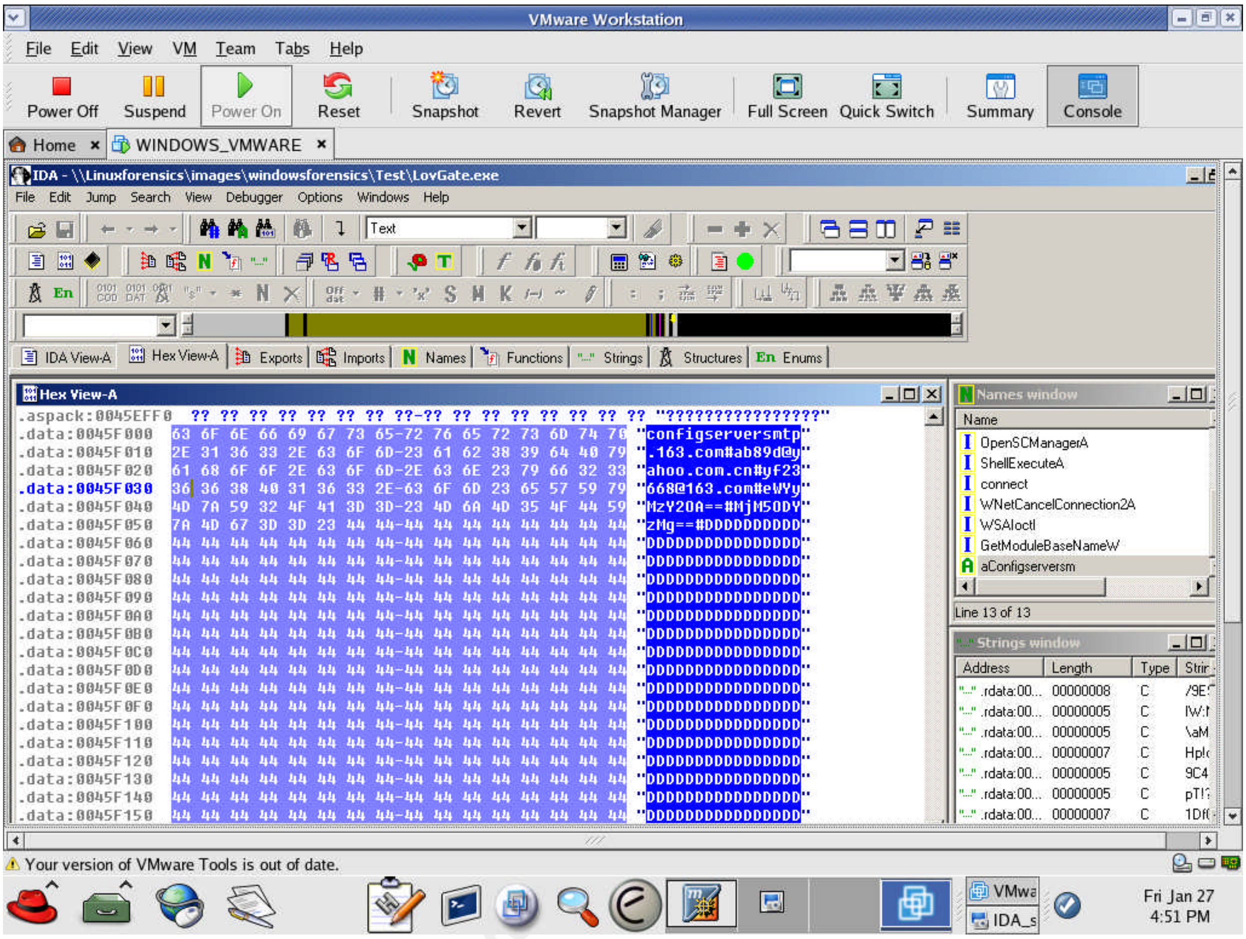
Less than one minute after killing the LovGate process a heavy activity was noticed in the system: taking a closer look at the Windows Task Manager program this activity was related to the brand new system processes: winrpc.exe, wingate.exe and so on.

The analysis reference files[14], excerpts from filemon and regmon logfiles, and the netstat views are located in the document appendix.

3.2 Malware advanced analysis

[14] Lovgate_filemon.LOG; Lovgate_Regmon.LOG; Table 7 – Network connection before LovGate execution; Table 8 – Network connection after LovGate execution

In order to discover some hidden malware features, the code analysis was tried by a disassembler program; the favourite software application was the IDA Software from Datarescue[8] a LovGate examination screenshot is depicted in the following picture:



```
.aspack:0045EFF0  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  "????????????????"
.data:0045F000    63 6F 6E 66 69 67 73 65-72 76 65 72 73 6D 74 70  "configserversmtp"
.data:0045F010    2E 31 36 33 2E 63 6F 6D-23 61 62 38 39 64 40 79  ".163.com#ab89d@y"
.data:0045F020    61 68 6F 6F 2E 63 6F 6D-2E 63 6E 23 79 66 32 33  "ahoo.com.cn#yf23"
.data:0045F030    36 36 38 40 31 36 33 2E-63 6F 6D 23 65 57 59 79  "668@163.com#eWYy"
.data:0045F040    4D 7A 59 32 4F 41 3D 3D-23 4D 6A 4D 35 4F 44 59  "MzY2OA==#MjM5ODY"
.data:0045F050    7A 4D 67 3D 3D 23 44 44-44 44 44 44 44 44 44 44  "zMg==#DDDDDDDDDD"
.data:0045F060    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F070    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F080    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F090    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0A0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0B0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0C0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0D0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0E0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F0F0    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F100    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F110    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F120    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F130    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F140    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
.data:0045F150    44 44 44 44 44 44 44 44-44 44 44 44 44 44 44 44  "DDDDDDDDDDDDDDDD"
```

Name
OpenSCManagerA
ShellExecuteA
connect
WNetCancelConnection2A
WSAIoctl
GetModuleBaseNameW
aConfigserversm

Line 13 of 13

Address	Length	Type	Strir
"..." .rdata:00...	00000008	C	/9E
"..." .rdata:00...	00000005	C	IW:!
"..." .rdata:00...	00000005	C	\aM
"..." .rdata:00...	00000007	C	Hp!(
"..." .rdata:00...	00000005	C	9C4
"..." .rdata:00...	00000005	C	pT!?
"..." .rdata:00...	00000007	C	1Df(

Figure 13 - IDA Disassembler screenshot

The program analysis reveals it is a portable executable for 386 file type; it is packed by aspack[9] to reduce the size and to protect the program from non professional hackers reverse engineering.

The malware program makes use of some Windows standard system functions reported in the following page table:

Imported Windows Library	Function	Entry Point[Hex]

	Start	0045 D001
Kernel32.dll	GetProcAddress	0045 E1CC
Kernel32.dll	GetModuleHandleA	0045 E1D0
Kernel32.dll	LoadLibraryA	0045 E1D4
User32	DispatchMessageA	0045 E33B
Gdi32	GetStockObject	0045 E343
Advapi32	OpenSCManagerA	0045 E34B
Shell32	ShellExecuteA	0045 E353
Wsock32	connect	0045 E35B
Mpr	WnetCancelConnection2A	0045 E363
Ws2_32	WSAloctl	0045 E36B
Psapi	GetModuleBaseNameW	0045 E373
	Configserversm	0045 F000

Table 3 - LovGate software functions

4 Legal implications

As stated in the document introduction section the intranet server is currently out of either the company security policy or the incident response policy; it lacks from the security management point of view.

4.1 United States Code

In accordance with the United States Code the incident suffered by such server can be classified as Network Crime; it is ruled by the statute 18 U.S.C. § 1030 known as Computer Fraud and Abuse Act.

It criminalizes the hacking actions against the "protected computers": the term refers to computers belonging to government agencies, financial institutions or affecting commerce and communications - 18 U.S.C. § 1030 (e)(2).

Such activity produced a "damage" – any impairment to the integrity or availability of data, a program, a system or information – 18 U.S.C. § 1030 (e)(8) worth more than $ 5000 in one year period.

The expenses the damage refers to are assessment, data loss and restore costs; they can be aggregated with other networked computers suffering the same attack - 18 U.S.C. § 1030 (e)(11).

If the crime author acted by intentional conduct – knowingly transmitting a "program, information, code or command" – he committed a felony violation of the law. 18 U.S.C. § 1030 (a)(5)(A)(i).

This may apply both to insiders (employees) and to outsiders (hackers): the maximum penalty is a fine and 10 years of imprisonment - 18 U.S.C. § 1030 (c)(4)(A)&(C).

4.2 Italian Criminal Law

According to the Italian Criminal Law – Computer Crime Statute, the incident caused a computer system unauthorized access punished with 3 years of imprisonment maximum penalty. Art. 615/ter.

Another issue is the spreading of programs that can damage or deny the access to a computer system (worms and viruses). Art. 615/quinques.

Some more privacy issues are ruled by the law n.675/1996 and its recent updates.

5 Lesson learned

Nowadays the system information security has become much more important than just a few years ago. Virus threats are frequently reported in the news and many companies have been hit directly. Such companies are therefore becoming aware of the money and image losses caused by security incidents and they begin to invest a small IT budget slice in security technologies.

Sometimes the task of security management becomes a problem for several reasons, such as the lack of resources and low sensitivity both from technical and managerial personnel.

For example, the incident described in this document could have been avoided easily. In fact, to prevent the spreading of the malware program through the enterprise network, an automatic antivirus signature update and a scheduled daily virus scan activities could have been enough.

A more exhaustive preventive solution could have been to couple the measure above with a recent IDS/IPS technology deployed in the strategic computer network points.

From the company organizational point of view, it would be very important to distribute the right task load to the system administrators and force them to update the hardware/software inventory in order to ensure no system lies forsaken and forgotten.

From perspective of the forensic analyst some points are success keys to improve the overall analysis process:

- The adoption of the external hard disk drive in order to speed up the evidence data gathering phase: in such a way it is possible to avoid the limitations of the hard disk on the forensic workstation and copying the whole physical disk instead of separate partitions. Using this method the examiner is not tied to network constraints. Furthermore, there would be no need to carry the forensic workstation to the incident site.
- Grouping commands and using the regular expressions in the program scripts could be very useful to save time in the media analysis phase especially when analysing tons of material like in the case described in the current document.
- It is also important to be very careful in the data correlation task to avoid following false tracks.
- Another issue in the forensic analysis described in the document is the long time passed between the initial virus infection and the time of the forensic examination. For example tracks of the virus activity (MAC times, processes, network connections) could have been covered by normal system activities.

6 Appendix – References

wft.log   Dirty_words.xls   Lovgate_g_description.rtf   Lovgate_Hexdump_File.rtf   Lovgate_filemon.LOG   Lovgate_Regmon.LOG

Table 4 - Volume_dump utility output

```
H:\response_kit\win2k_xp\volume_dump.EXE
Volume Dump Utility, 1. 0. 0. 330
Copyright © 2002 George M. Garner Jr.

Volume Name:          \\?\Volume{9acab338-674e-11d6-bfe8-806d6172696f}
Volume Label:
Drive Type:           Fixed
Serial Number:               140041511
Maximum Component Length:    255
Volume Characteristics:
                      File system preserves case
                      File system supports case sensitive file names
                      File system supports Unicode file names
                      File system preserves and supports persistent ACL's
                      File system supports file level compression
                      File system supports named streams
                      File system supports encryption
                      File system supports object identifiers
                      File system supports reparse points
                      File system supports sparse files
                      File system supports quotas
File System:          NTFS
Volume Extents:
       Disk Number:   0
       Starting Offset: 0x0000000000007e00
       Extent Length:   0x00000000dc8f4200
Volume Name:          \\?\Volume{5200890c-7214-11d6-acae-806d6172696f}
Volume Label:
Drive Type:           Fixed
Serial Number:               1355578721
Maximum Component Length:    255
Volume Characteristics:
                      File system preserves case
                      File system supports case sensitive file names
                      File system supports Unicode file names
                      File system preserves and supports persistent ACL's
                      File system supports file level compression
                      File system supports named streams
                      File system supports encryption
                      File system supports object identifiers
                      File system supports reparse points
                      File system supports sparse files
                      File system supports quotas
File System:         NTFS
Volume Extents:
       Disk Number:   0
       Starting Offset: 0x00000000dc903e00
       Extent Length:   0x000000036d2b2200
Volume Name:         \\?\Volume{5200890d-7214-11d6-acae-806d6172696f}
Volume Label:
Drive Type:          Fixed
Serial Number:              2978877375
Maximum Component Length:   255
Volume Characteristics:
                     File system preserves case
                     File system supports Unicode file names
File System:         FAT
Volume Extents:
       Disk Number:   0
       Starting Offset: 0x0000000449bbde00
       Extent Length:   0x000000005f836200
Volume Name:         \\?\Volume{3df0557a-7221-11d6-9b52-806d6172696f}
Volume Label:        Dati2
Drive Type:          Fixed
Serial Number:              2017617590
Maximum Component Length:   255
Volume Characteristics:
                     File system preserves case
                     File system supports case sensitive file names
                     File system supports Unicode file names
                     File system preserves and supports persistent ACL's
                     File system supports file level compression
                     File system supports named streams
                     File system supports encryption
                     File system supports object identifiers
                     File system supports reparse points
                     File system supports sparse files
                     File system supports quotas
File System:         NTFS
Volume Extents:
       Disk Number:   1
       Starting Offset: 0x0000000000007e00
       Extent Length:   0x0000000393778200
Volume Name:         \\?\Volume{ba430000-8313-11d8-9d50-806d6172696f}
Volume Label:
Drive Type:          Fixed
Serial Number:              224860127
Maximum Component Length:   255
Volume Characteristics:
                     File system preserves case
                     File system supports Unicode file names
File System:         FAT32
Volume Extents:
       Disk Number:   2
       Starting Offset: 0x0000000000007e00
       Extent Length:   0x0000000157ac7c00
Volume Name:         \\?\Volume{ba430001-8313-11d8-9d50-806d6172696f}
Volume Label:        IBM_SERVICE
```

```
Drive Type:         Fixed
Serial Number:             807475203
Maximum Component Length:  255
Volume Characteristics:
                    File system preserves case
                    File system supports Unicode file names
File System:        FAT32
Volume Extents:
       Disk Number:   2
       Starting Offset: 0x00000001da939000
       Extent Length:   0x000000007ff58a00
Volume Name:        \\?\Volume{ba430002-8313-11d8-9d50-806d6172696f}
Volume Label:
Drive Type:         Fixed
Serial Number:             1004013017
Maximum Component Length:  255
Volume Characteristics:
                    File system preserves case
                    File system supports Unicode file names
File System:        FAT
Volume Extents:
       Disk Number:   2
       Starting Offset: 0x000000015c948c00
       Extent Length:   0x000000007dff0400
Volume Name:        \\?\Volume{955bcc7f-7224-11d6-852d-806d6172696f}
Volume Label:       ForensicTrack8
Drive Type:         CDROM
Serial Number:             2034888187
Maximum Component Length:  110
Volume Characteristics:
                    File system supports case sensitive file names
                    File system supports Unicode file names
File System:        CDFS
Volume Name:        \\?\Volume{9acab336-674e-11d6-bfe8-806d6172696f}
Volume Label:
Drive Type:         Removable
Serial Number:             0
Maximum Component Length:  0
Volume Characteristics:
File System:
```

Table 5 - String search sample results in the C partition

Byte #	Cluster #	Metadata #	Filename	String	Creation Date	Access date
1.945.769.655	475041	24	c:\pagefile.sys	vxload.vxd	12/14/05 15.46	
1.945.908.960	475075	24	c:\pagefile.sys	username.log	12/14/05 15.46	
1.945.918.184	475077	24	c:\pagefile.sys	c:\Paslog.txt	12/14/05 15.46	
1.945.935.087	475081	24	c:\pagefile.sys	PaSsWoRd	12/14/05 15.46	
1.945.988.024	475094	24	c:\pagefile.sys	www.lsky lion	12/14/05 15.46	
1.945.995.196	475096	24	c:\pagefile.sys	\\donkey.log	12/14/05 15.46	
1.945.997.398	475097	24	c:\pagefile.sys	smtp.hotbox.ru	12/14/05 15.46	
1.946.002.623	475098	24	c:\pagefile.sys	systemks.exe	12/14/05 15.46	
1.946.150.979	475134	24	c:\pagefile.sys	kill.bat	12/14/05 15.46	
1.946.341.074	475180	24	c:\pagefile.sys	various	12/14/05 15.46	
1.946.853.148	475305	24	c:\pagefile.sys	port.ru	12/14/05 15.46	
1.946.944.736	475328	24	c:\pagefile.sys	AppRedUp.exe	12/14/05 15.46	
1.992.121.340	486357	24	c:\pagefile.sys	poopdeck shat	12/14/05 15.46	
2.083.327.430	508624	24	c:\pagefile.sys	klez	12/14/05 15.46	
2.083.887.379	508761	24	c:\pagefile.sys	\hiddenrun.exe	12/14/05 15.46	
2.133.837.083	520956	24	c:\pagefile.sys	BACKDOOR	12/14/05 15.46	
3.534.363.539	862881	13291	c:\...*.VBN	www.lsky lion		
	855389	13686	c:\...*.VBN	ddraw32.dll dumb@ss		
833.465.253	203482	13843	c:\...*.VBN	www.lsky lion	9/3/03 14.49	12/2/03 16.53
81.414.923	19876	14482	c:\...*.VBN	immortal.hackers.com	12/1/03 16.41	12/2/03 16.48
832.948.423	203356	14712		c:\cgl.bat	12/1/03 16.52	12/2/03 16.49
	202854	15614		BACKDOOR		
	202839	17811	c:\...*.VBN	bomb	12/1/03 17.43	12/2/03 16.49
2.560.569.802	625139	17834	c:\...*.VBN	hlog.dll	12/1/03 17.44	12/2/03 16.49
1.312.369.947	320402	18307	c:\...*.VBN	zombie@gmx.ne	12/2/03 12.16	12/2/03 16.50
177.037.225	43222	19870	c:\...*.VBN	rooted, Ph33r	12/2/03 13.26	12/2/03 16.49
1.544.700.995	377124	21153	c:\...*.VBN	http://dodo521.126.com	12/2/03 14.20	12/2/03 16.53
3.294.408.528	804299	22240	c:\...*.VBN	WFWNET.EXE	12/2/03 14.42	12/2/03 16.53
	136335	22579	c:\...*.VBN	bomb	12/2/03 14.56	12/2/03 14.56
515.372.748	125823	22710	c:\...*.VBN	dark-society.ml.org	12/2/03 15.01	12/2/03 16.53
564.013.449	137698	22787	c:\...*.VBN	hackBoy	12/2/03 15.05	12/2/03 16.49
	137872	22839	c:\...*.VBN	BACKDOOR		
704.399.685	171972	24557	c:\...*.VBN		12/2/03 16.03	12/2/03 16.52
1.742.289.188	425363	24692	c:\...*.VBN	$username $password	12/2/03 16.06	
1.742.325.800	425372	24692	c:\...*.VBN	www.lsky lion		
	21774	24744	c:\...*.VBN	configserversmtp.163.com #ab89d@yahoo.com		
664.216.198	162162	31142	c:\winnt\memory.dmp	sars.tar	1/7/04 9.31	1/7/04 9.31

641.760.784	156679	36142	c:\winnt\memory.dmp	GravediggerV2.0	1/7/04 9.31	1/7/04 9.31
642.806.054	156935	36142	c:\winnt\memory.dmp	Worm.jpg.vbs	1/7/04 9.31	1/7/04 9.31
1.378.176.428	336468	36142	c:\winnt\memory.dmp	Wingate rundl1.exe h43K42.bat	1/7/04 9.31	1/7/04 9.31
1.379.698.620	336840	36142	c:\winnt\memory.dmp	rs]>>bp.reg	1/7/04 9.31	1/7/04 9.31
3.099.435.894	756698	36142	c:\winnt\memory.dmp	winrpc	1/7/04 9.31	1/7/04 9.31
1.708.009.969	416994	37670	c:\.....\drwatson\user.dmp	Wingate rundl1.exe h43K42.bat	3/26/04 12.50	3/26/04 12.50
208.118.906	50810		un.	vmload.vxd		
243.211.252	59377		un.	dllhost.exe		
689.487.177	168331		un.	\\c.tmp\n#A		
831.324.909	202960		un.	ddraw32.dll		
831.473.640	202996		un.	c:\ntdetect.vbs		
832.095.237	203148		un.	smtp.mail.ru		
886.549.554	216442		un.	Wingate rundl1.exe h43K42.bat		
1.544.901.803	377173		c:\...*.VBN	Melhack	12/2/03 13.32	12/2/03 16.50
	377269			c:\adride.exe c:\kirin.bat VX team		
1.545.319.959	377275		un.	root@hacker		
1.545.597.836	377343		un.	\mirlog.vxd		
1.546.097.377	377465		un.	Demo for sniffTrojan		
1.546.731.878	377620		un.	fg32.exe		
1.546.818.218	377641		un.	test4me1@yahoo.com		
1.546.997.492	377684		un.	WS2_32.dll		
	203011		un.	bomb		
	752452		un.	BACKDOOR STATDX modify by H-A-O-S		

Table 6 - String searches samples in the C partition unallocated space

Data unit #	Cluster #	Metadata #	Filename	String	Creation Date	Access date
29950	57212	23749	*.VBN	Configserversmtp…	02/12/2003	
42314	70964	23942	*.VBN	Configserversmtp…	02/12/2003	
178344	307619	22166	*.VBN	Configserversmtp…	02/12/2003	
307074	850983	43352	*.VBN	Configserversmtp…	02/12/2003	
129072	203482	13843	*.VBN	www.heibai.net www.lsky flooding	16/07/2003	03/09/2003
223499	377141	21006	*.VBN	Total scanning bot server infected	02/12/2003	02/12/2003
277177	772623	un		Remote exploit; security hole		
314324	868946	15589	_INSTI32I.EX_	Next to the victim; admin@fl.com	02/09/03	
6023	19876	14482	*.VBN	!passwd; immortal-hackers.com	01/12/2003	

24440	50854	18270	*.VBN	Picture.jpg.vbs; batch trojan	
294884	829443	Un.		Scriptbot; igloo trojan	
191199	324348	20886	*.VBN	trojan	
128738	203148	Un.		www.hao3344.com	
223701	377343	Un.		@sina.com.cn; @netease; @etang.com	
224042	377684	Un.		Antique.az.com	
281934	781454	Un.		@163.com	
95989	136335	22579	*.VBN	bomb	02/12/2003
128429	202839	17811	*.VBN	bomb	01/12/2003
128601	203011	Un		bomb	
96748	137872	22839	*.VBN	backdoor	
128444	202854	15614	*.VBN		
266989	752452	Un.			

Table 7 - Network connections before LovGate execution

```
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    192.168.2.10:139       0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1028      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1028      192.168.2.1:139        ESTABLISHED
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    192.168.2.10:137       *:*
  UDP    192.168.2.10:138       *:*

  UDP    192.168.2.10:500       *:*
```

Table 8 - Network connections after the LovGate execution

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1092           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1351           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1353           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1355           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1357           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1359           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1361           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1363           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1365           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1367           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1369           0.0.0.0:0              LISTENING
  TCP    192.168.2.10:139       0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1028      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1028      192.168.2.1:139        ESTABLISHED
  TCP    192.168.2.10:1351      192.168.2.163:445      SYN_SENT
  TCP    192.168.2.10:1352      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1352      192.168.2.163:139      SYN_SENT
  TCP    192.168.2.10:1353      192.168.2.164:445      SYN_SENT
  TCP    192.168.2.10:1354      0.0.0.0:0              LISTENING

  TCP    192.168.2.10:1354      192.168.2.164:139      SYN_SENT
  TCP    192.168.2.10:1355      192.168.2.165:445      SYN_SENT
  TCP    192.168.2.10:1356      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1356      192.168.2.165:139      SYN_SENT
  TCP    192.168.2.10:1357      192.168.2.166:445      SYN_SENT
  TCP    192.168.2.10:1358      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1358      192.168.2.166:139      SYN_SENT
  TCP    192.168.2.10:1359      192.168.2.167:445      SYN_SENT
  TCP    192.168.2.10:1360      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1360      192.168.2.167:139      SYN_SENT
  TCP    192.168.2.10:1361      192.168.2.168:445      SYN_SENT
  TCP    192.168.2.10:1362      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1362      192.168.2.168:139      SYN_SENT
  TCP    192.168.2.10:1363      192.168.2.169:445      SYN_SENT
  TCP    192.168.2.10:1364      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1364      192.168.2.169:139      SYN_SENT
  TCP    192.168.2.10:1365      192.168.2.170:445      SYN_SENT
  TCP    192.168.2.10:1366      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1366      192.168.2.170:139      SYN_SENT
  TCP    192.168.2.10:1367      192.168.2.171:445      SYN_SENT
  TCP    192.168.2.10:1368      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1368      192.168.2.171:139      SYN_SENT
  TCP    192.168.2.10:1369      192.168.2.172:445      SYN_SENT
  TCP    192.168.2.10:1370      0.0.0.0:0              LISTENING
  TCP    192.168.2.10:1370      192.168.2.172:139      SYN_SENT
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1026           *:*
  UDP    192.168.2.10:137       *:*
  UDP    192.168.2.10:138       *:*
  UDP    192.168.2.10:500       *:*
```

6.1 References

[1] System Forensics, Investigation & Response by SANS Institute -http://www.sans.org/sans2006/description.php?tid=205

[2] http://www.foolmon.net/security

[3] http://users.erols.com/gmgarner/forensics

[4] http://www.sleuthkit.org

[5] http://www.winalysis.com

[6] by Marc Russinovich and Bryce Cogswell – http://www.sysinternals.com/SystemInformationUtilities.html

[7] by Marc Russinovich and Bryce Cogswell – http://www.sysinternals.com/SystemInformationUtilities.html

[8] http://www.datarescue.com/idabase

[9] http://aspack.com