Malware Analysis And Reversing Engineering

Andrea Mambretti (andrea2.mambretti@mail.polimi.it)

Politecnico di Milano

September 11, 2012

Education

- In 2008 I received my high school diploma in Information Technology
- In 2011 I received my bechelor degree in Computer Science Engineering @POLIMI
- I'm a Master of Science student in Computer Science Engineering @POLIMI

Interests

- I'm mainly interest in Security (Malware Analysis, Reversing Engineering, Exploiting and other low level stuff) and in Hacking Operative System (Linux and FreeBSD kernel)
- I'm in the security team of Profs Maggi/Zanero for my master thesis in Automatic labeling behaviour system for malware
- I'm a component of the CTF team of polimi

What is Malware?

Definitions

- A Malware is any program that works against the interests of the system's user or owner
- Malware is short for malicious software - computer programs designed to infiltrate and damage computers without the users consent. Malware is the general term covering all the different types of threats to your computer safety

Which are the targets?

- Actually everyone can be a target indipendently from the Operative System they run!!!
- But Windows users (again the majority sadly) are the most exposed

Taxonomy

- Viruses
- Worms
- Trojan Horses
- Backdoors
- Mobile Code
- Adware/Spyware
- Rootkit
- Botnet
- Downloader and/or Launcher
- Future Malware (such as Information-Stealing Worms and BIOS/Firmware Malware)

What is Malware used for?

- Backdoor Access
- Denial of Service (DoS) Attack
- Vandalism
- Resource Theft
- Information Theft

What is Basic Static Analysis?

- B.S.A. consists in a very simple set of operation that can allow a malware analyst to get usefull information about a certain binary file
- B.S.A. tools can give us infomation about:
  - **Antivirus Scanning**: tell us if it is an already known malware or not and if yes which kind (ex www.virustotal.com)
  - **Hashing**: tell if a certain file has been corrupted or not
  - **Strings**: give us all the strings defined as costant in the program that can be usefull to undestand what that file does
  - **Recognizers Packing and Obfuscation**: if the file uses some type of packing (ex: upx) or obfuscation to avoid recognition
  - **Header Analysis**: Looking inside the header give us information about import and export function, when it's compiled, if it's packet, if there's some extra segment and other stuff

When is it necessary?

- If someone is really paranoic the answer is every time before launch an application
- for all the others when something of suspicious is detected during the previous execution
- after this phase a human analyzer knows if it is needed to procede with other analysis such as dynamic and advaced static analysis

What is Dynamic Analysis?

- D.A. consists in looking the behaviour of a malware running it and loggin every change and action done in the system
- of course is not always possible use this technique because a specific malware can damage the system and makes the information unreachable
- so what can we do to avoid it?

Possible Solution: Virtual Machine

- Using Virtual Machine we can set a perfect enviroment for our malware (fake a whole network of hosts,logger etc.)
- We can check out of box what happens inside (system call, network operation etc.).
  if something goes wrong we can, using snapshots, rollback the machine state to a sane position
- Possible sequence of operation:
  - Start with a clean snapshot with no malware running on it
  - Trasfer the malware to the virtual machine
  - Conduct your analysis on the virtual machine
  - Take your notes, screenshots, and data from the virtual machine and transfer it to the physical machine
  - Revert the virtual machine to the clean snapshot

Existing VM

VM

Someone of them are already prepared to Malware Analysis

- VMware
- VirtualBox
- Qemu
- Cuckoo
- Anubis
- Andrubis
- A ton of others

Problems and Solution

Problems

- It's prooved Today's Malware can avoid the dynamic analysis. They recognize all the well-know VB and they change their behavior when are runned in
- The worst scenario happens when specific malware are done to exploit vulnerabilities inside VM to own the whole machine where the VB is running

Solution

- One of the possible solution to understand if the malware has behaved not in his standard way and which are his possibility is to see the code more deeply with the Advanced Static Analysis (aka Reverse Engineering)

What is Reverse Engineering in General?

Definition

- Reverse engineering is taking apart an object to see how it works in order to duplicate or enhance the object. The practice, taken from older industries, is now frequently used on computer hardware and software.
- Reverse engineering is usually conducted to obtain missin knowledge, ideas, adn design philosophy when such information is unavailable

What is Software reverse engineering?

Definition

- Software reverse engineering involves reversing a program's machine code (the string of 0s and 1s that are sent to the logic processor) back into the assembly language (x86, x86-64, ARM so on and so forth)
- Software reverse engineering requires a combination of skills and a thorough undestanding of computers and software development, but like most worthwhile subjects, the only real prerequisite is a strong curiosity and desire to learn. Software reverse engineering integrates several arts: code breaking, puzzle solving, programming, and logical analysis.

When do we use it?

- Reversing Application
- Security-Related Reversing
  - Malicious Software
  - Reversing Cryptographic Algorithms
  - Digital Rights Management
  - Auditing Program Binaries
- Reversing Software Development
  - Achieving Interoperability with Proprietary Software
  - Developing Competing Software
  - Evaluating Software Quality and Robustness

Background of a good Reverser

- Assembly Language
- Compilers
- Virtual Machine and Bytecodes (ex Java)
- Operative Systems

Tools of a good Reverser

- System-Monitoring Tools
- Disassemblers
- Debuggers
- Decompilers

Some Common Reversing Tools

- IDA Pro
- OllyDbg
- WinDbg
- PEBrowse Professional Interactive
- SoftICE

Is it Legal?

- The Legal debate around reverse engineering has been going on for years
- 1998 - Digital Millenium Copyright Act
  - Circumvention of copyright protection systems
  - The development of circumvention technologies
- Luckily, DMCA makes several exceptions
  - Interoperability
  - Encryption research
  - Security testing
  - Educational institutions and public libraries
  - Government investigation
  - Regulation
  - Protection of privacy

Reverse on Malware and Antireversing Techniques

Today's malware and commercial program use techniques against Reversing

- Anti-Disassembly (Linear and Flow Oriented Disassembly)
  - Jump instructions with the same target
  - Jump instruction with a Costant Condition
  - Impossible disassembly
- Anti-Debugging
  - They understand that are executed in a debugger and change their behaviour either crashing itself, the debugger or totally doing other stuff
- Anti-Virtual Machine Techniques
- Packers and Unpacking

Conclusion

- Software Reverse Engineering is a very powerfull instrument but it requires a lot of lowlevel-knowledge
- Appling this technique on malware analysis is not optional if we want undestand how the malware works
- Can be very time consuming and if all the tools used for the analysis are not setted correctly there's no way to reverse the malware

End

- Thanks Folk...Questions?
- So now one practical example