REVERSE ENGINEERING – CLASS 0x08

FUTURE DIRECTIONS IN RE

Cristian Rusu

LAST TIME

- **Running code that is not native**
- **.NET RE**
- **Java RE**

.

TODAY

- **Review**

- **Future directions**

MAKE SURE YOUR SYSTEM IS STILL YOURS

.

MAKE SURE YOUR SYSTEM IS STILL YOURS

- **rootkits**
- **System Call Hooking**

- **https://exploit.ph/linux-kernel-hacking/2014/07/10/system-call-hooking/index.html**
- **https://blog.aquasec.com/linux-syscall-hooking-using-tracee**

- **Alex Matrosov, Eugene Rodionov, Sergey Bratus, “Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats”, 2019**

MAKE SURE YOU HAVE THE CORRECT TOOLS

.

MAKE SURE YOU HAVE THE CORRECT TOOLS

- **static analysis**
- **dynamic analysis**

- **what we did not talk about is network activity: wireshark**
- **(maybe also detailed memory forensics)**

- **make sure you have the correct setup**
  - isolation
  - sandbox/VM

https://www.wireshark.org/

ANTI-RE METHODS

.

ANTI-RE METHODS

- **Anti-debugging**
- **Anti-VM**
- **Obfuscation**
- **Packing**
- **...**

ANTI-RE METHODS: ANTI-DEBUGGING

- **try to guess that the current process is being debugged**
  - if it is, do nothing “interesting”

- **find anti-debugging tools in: running processes, title of the windows, registry installation keys**
- **both Windows and Linux have APIs to detect debuggers:**
  - Windows we have *IsDebuggerPresent*, *CheckRemoteDebuggerPresent*, *ProcessDebugPort*, *OutputDebugString* etc.
  - Linux we have *ptrace*, *procfs*, etc.

- ***TrapFlag* for each instruction**

- **Check Point Research, Anti-debugging tricks https://anti-debug.checkpoint.com/**
- **Anti-debugging techniques, https://users.cs.utah.edu/~aburtsev/malw-sem/slides/02-anti-debugging.pdf**
- **Anti-debugging topics, https://github.com/topics/anti-debugging**

.

ANTI-RE METHODS: ANTI-VM

- **may be hard, because the whole point of a VM is to make the guest OS “feel” like it is running on bare metal**

- **Anti-VM (connected to anti-debugging): parameters of the system, OS version, timing, etc.**

- **Anti-debugging and anti-VM techniques and anti-emulation, https://resources.infosecinstitute.com/topic/anti-debugging-and-anti-vm-techniques-and-anti-emulation/**

.

ANTI-RE METHODS: OBFUSCATION

- **makes code harder to understand**
- **prevents patterns matching (*metamorphing malware*)**
- **comes with performance penalties**
- ***data obfuscation*: reordering, encoding, data to procedures …**

- ***plain-text code obfuscation* is (kinda) trivial: js, php, python, etc.**

- ***machine code obfuscation*:**
  - **NOP instruction insertion**
  - **non-sense instruction insertion**
  - **replacing instructions**
  - **reordering instructions**
  - **adding jump instructions**
  - **function joining**
  - **control flow flattening**
  - **opaque jump conditions**
  - **…**

https://www.blackhat.com/presentations/bh-usa-08/Hosmer/BH_US_08_Hosmer_Polymorphic_Malware.pdf

ANTI-RE METHODS: PACKING

- **packing: blocks static analysis (mostly), *UPX***
- **very few imports from libraries**
- **disassembler can virtually find nothing useful**
- **PE header contains the usual suspects, *UPX0 …***
- **high entropy (something is encrypted or compressed)**

https://upx.github.io/

ANTI-RE METHODS: OBFUSCATION AND PACKING

- **MALWARE ANALYSIS - VBScript Decoding & Deobfuscating, https://www.youtube.com/watch?v=3Q9-X_NRlJc**

- **Lecture 26: Obfuscation, https://www.cs.cmu.edu/~fp/courses/15411-f13/lectures/26-obfuscation.pdf**

- **A Tutorial on Software Obfuscation, https://mediatum.ub.tum.de/doc/1367533/file.pdf**

- **Awesome Executable Packing, https://github.com/packing-box/awesome-executable-packing**

.

ANTI-CHEATING

.

ANTI-CHEATING

- **Kernel drivers (hooking in again a problem)**

- https://www.wired.com/story/kernel-anti-cheat-online-gaming-vulnerabilities/
- https://www.leagueoflegends.com/en-us/news/dev/dev-null-anti-cheat-kernel-driver/

- **Valve Anti-Cheat,** https://en.wikipedia.org/wiki/Valve_Anti-Cheat

.

MALWARE ANALYSIS

- **Workshop: Malware Analysis 1,** https://www.youtube.com/watch?v=d4d8VRsk4-0
- **Workshop: Malware Analysis 2,** https://www.youtube.com/watch?v=Gm9rzqM_RJk
- **Malware Hunting with Memory Forensics,** https://www.youtube.com/watch?v=ilmq5FUctsE

- **Analysis of RedXOR Malware,** https://ritcsec.wordpress.com/2022/05/06/analysis-of-redxor-malware/

RE AND ML

- **binary diffing**
- **source code diffing**

- **CodeQL: "CodeQL lets you query code as though it were data", https://codeql.github.com/**
- **AFL++: "AFL++ a brute-force fuzzer coupled with an exceedingly simple but rock-solid instrumentation-guided genetic algorithm", https://github.com/AFLplusplus/AFLplusplus**

- **LLM will affect code writing/debugging/RE**
  - PAY ATTENTION TO THIS!

.

NICE DEMOS

- **Modern Binary Exploitation, https://github.com/RPISEC/MBE**
- **LifeOverflow CTF playlist, https://www.youtube.com/watch?v=MpeaSNERwQA&list=PLhixgUqwRTjywPzsTYz28l-qezFOSaUYz**
- **Discover Vulnerabilities in Intel CPUs!, https://www.youtube.com/watch?v=x_R1DeZxGc0**
- **HakByte: How to use Postman to Reverse Engineer Private APIs, https://www.youtube.com/watch?v=mbrX1_CVG-0**

NICE CONFERENCES

- **DEF CON, https://www.defcon.org/**
- **CCC, www.media.ccc.de/c/35c3**
- **Hack in the Box, www.conference.hitb.org**
- **OffensiveCon, www.offensivecon.org**
- **RECON, https://recon.cx/**
- **Usenix ENIGMA, www.youtube.com/c/USENIXEnigmaConference/videos**

.

WHAT WE DID TODAY

- **Make sure your system is still yours**
- **Make sure you have the correct tools**
- **Anti-RE methods:**
  - anti-debugging
  - anti-VM
  - obfuscation
  - packing
- **Anti-cheating**
- **Malware analysis**
- **RE and ML**
- **References**
  - nice demos
  - nice conferences

.

NO NEXT TIME

- \0

.