Easy ways to bypass antivirus systems

Intruducion

Attila Marosi

OSCE, OSCP, ECSA, CEH

IT security expert at

GovCERT-Hungary (SSNS)

Email: attila.marosi@gmail.com

Web: http://marosi.hu

Twitter: @0xmaro

Why?

- All of us use AntiVirus (AV) systems
- These solutions are very important for us!
- Do we know the real abilities of these systems? (I trust my own experiances.)
- I want to MOTIVATE the vendors to make their job better.
- Who able to avoid these systems? (only just a few one or anyone)

What can you expect from this topic?

DEEPSEC

I will bypass, on the spot:

signatures, emulation/virtualization, sandboxing, firewalls, …

How much time is needed for this result?

- Only 15 hours without a cent investment.

BUT, it is a technical presentation so sadly some demo-effect could be happen ☺

Challenges?

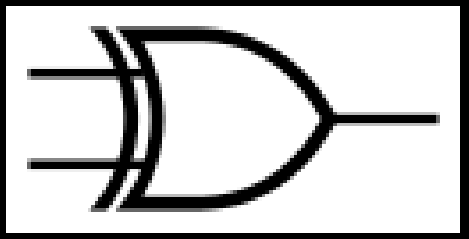
- Well-know shellcode (Metaspolit Framework)
  - shell_reverse_tcp
- Well-know techniques for avoid the detection
  - Just google „bypass antivirus” – tons of good articles.
- **Péter Szőr** – ‘The art of computer virus research and defense’ (2005)
- VirusTotal.com
  - 48 antivirus systems,
  - it is not equale with „desktop” test, but good for check the way
- 4 version will be tested with virtual PC in runtime
- „only” Microsoft Windows OS

DEEPSEC

DEEPSEC

DEMO

Code encryption



- XOR (exclusive or)
  - onyl signatures detection won't work
    - without emutation/virtualization this can't detectable
  - very easy to implement
  - not so easy to decrypting without information
  - the encrypting, decrypting process is same

DEEPSEC

DEMO

Code injection

- Main usage:
  - Dll injection
    - Load a dll to a selected (victim) process
  - Code Injection
    - Inject byte code to the selected (victim) process
    - Position-independent code (PIC) is needed!

Code injection (2)

- The attacker (evil) perspective:
  - easy to implement and use
  - we can act by the name of the victim process!
  - msfpayload shellcode(s) are PIC
- For the AV(s) perspective :
  - the emulation/virtualization is difficult
  - need to monitoring kernel API calls (e.g.: kernel API hooking)

Firewall bypass

- We need to inject our code to a process which has right to comm. on the network (e.g. iexplorer.exe).
- How we can find a good one?
  - API calls
    - GetTcpTable2()
  - basic built in commands
    - netstat -no

Import table

- Every external function which is used by a program is listed in the Import Table (it is a basic functionality of the PE files)
- These Import Tables rows are observed by AVs
- These calls are suspicious:
  - OpenProcess
  - VirtualAllocEx
  - WriteProcessMemory
  - CreateRemoteThread !! <- this is the worst

DEEPSEC

DEMO

Metamorphous „encoding”

- Metamorphous codes
  - junk commands (pl.: NOP)
  - change registers
  - chage commands to similars

    original: **push dword 0x9dbd95a6**

    metamorf.: **push dword 0xc5ee94b1**

    sub dword [esp], 0x2830ff0b
- How?
  1. msfpayload -> ndisasm (disassembler)
  2. **change the ASM source-code** with Python
  3. rebuild the code with nasm

DEEPSEC

DEMO

DEEPSEC

THANK YOU FOR YOUR ATTENTION! ANY QUESTIONS?