ROOTKITS

Operates near the kernel, can slide in anything and launch attacks etc.

Types of rootkits:

1. Hardware / Firmware
   Infection of Hard drive, router, BIOS, motherboard - hard to detect
   Less common but pose a severe threat
2. Memory
   Hide in RAM, most of the time disappear after reboot and they don't inject permanent code. Don't pose that big of a threat
3. Bootloader
   Replaces the legitimate one with the malicious one. This leads to activation of the rootkit before the OS is fully loaded.
4. Kernel mode
   Access the kernel level and can change the functionality of OS. Pose the most severe threat of them all.
5. Virtual
   Loads underneath the OS, hosts the target OS as a VM and intercepts hardware calls. Hard to detect

Short History

- 1990: First ever rootkit created by Lane Davis and Steven Dake for the SunOS
- 1999: NTRootkit is the first ever Windows rootkit
- 2005: Sony BMG rootkit as an anti-piracy tool
- 2009: Machiavelli rootkit is the first ever macOS rootkit
- 2010: Stuxnet
- 2019: LoJax infects UEFI, letting LoJax survive an OS reinstall

Methods of Analysis

Memory Dump Analysis:

- Contains static snapshots of RAM. It's possible to create a memory dump for a single process, system kernel or the entire system. Techniques used with this method can be also deployed on a live system.

Signature based Analysis:

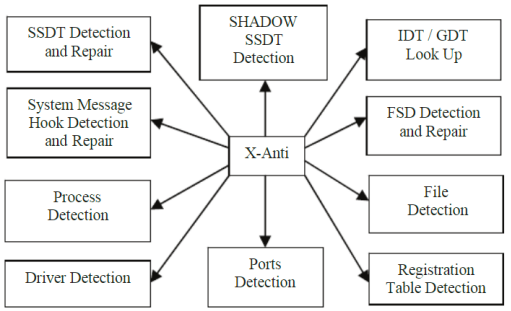
- This method depends on fixed byte sequences from known rootkits. Most AV tools use sigs extracted from rootkit bodies.

Interceptions:

- Rootkits can replace or modify pointer tables(IDT,SSDT,IAT) to specify its own handlers for certain inputs.

Data Comparison:

- Comparison of high-level and low-level system calls may be used in detection of their presence. Another variation is comparing process memory loaded into RAM with the content of the file stored on the hard disk.



Ideal Rootkit detection method (Credits: Leian Liu)

Memory Dumps

Four types of memory dumps:

- Crash dumps; (Created by crashes, entire content of sys memory is saved)
- Raw dumps; (Complete snapshots of OS)
- hiberfil.sys; (used by the OS to enable hibernation, the size of this file will equal to the size of the computers RAM)
- Vmem. (Created by VMware, contains entire volatile memory of a VM)

IDT

- User-mode applications requesting execution of a system-level function will need to raise their privilege level. This is implemented via program interrupts
- In Windows NT, we use the int 2e command to initialize a system request. Int is program interrupt and 2e is referring to a address in the IDT index
- System calls are made using dispatcher service(KiSystemService)
- if `!idt 2e` in WinDbg does not show KiSystemService then the routine has been intercepted

Searching for MZ sigs

- s -d 0×0 L?0xffffffff 0×00905a4d = listing all entries containing the MZ signature
- !lmi = lists all information about the module
- if the module was directly infused into the memory the system will know nothing about the process and the command will produce an error output

Searching for hidden processes

- Windows NT distributes CPU cycles between threads.
- There are three ques:
  1. KiDispatcherReadyListHead = ready to execute
  2. KiWaitInListHead = waiting for a certain event
  3. KiWaitOutListHead = waiting for a certain event
- Enumerate the thread lists, refer to a process owning a particular thread and verify if the process is listed in the process list