ECE 4309

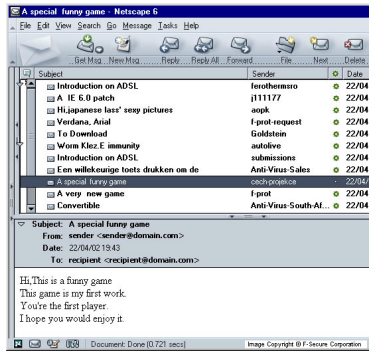
Malicious Software – part 2

Dr. Valerio Formicola

Payload System: File Corruption

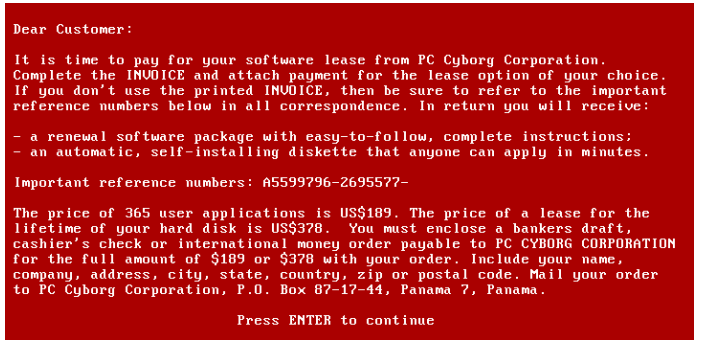
- Payload refers to the portion of code in Viruses, Worms and Trojans that is in charge of executing malicious actions
- Examples:
  - Chernobyl virus (1998) targeted executable files in Win 95, 98.
    Time bomb: when a trigger date is reached, the virus deletes data on the infected system by overwriting the first megabyte of the hard drive with zeroes, resulting in massive corruption of the entire file system
  - Klez mass worm (2001) in Win 95-XP. On the 13th of the month it was emptying local files (0 size) and mailing to contacts in the email address book. It can stop and delete some anti-virus programs running on the system

Chernobyl virus:
File system damaged



Klez spreading emails

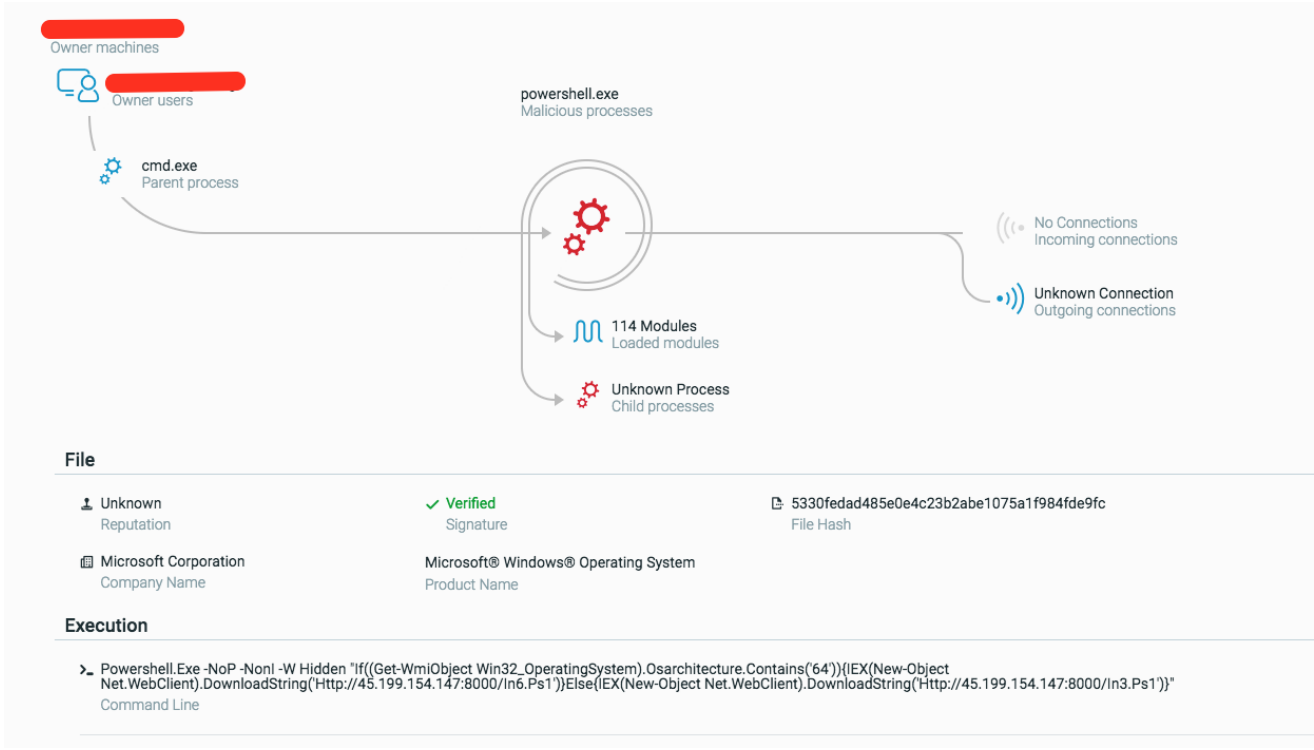
Payload System: File Encryption/Ransomware



- Ransomware
  - Encrypts the user's data and demands payment in order to access the key needed to recover the information
- PC Cyborg Trojan, aka AIDS (1989):
  - Mid-2006 a number of worms and Trojans appeared that used public-key cryptography with increasingly larger key sizes to encrypt data.
  - Vector: Spam emails or drive-by-download
  - The user needed to pay a ransom, or to make a purchase from certain sites, in order to receive the key to decrypt this data
- Wannacry (2017)
  - Targeted mobiles, Linux servers, Windows machines
  - Demanded ransom payment in BitCoins threatening to publish secret data
  - Could avoid to pay if there was a recent and complete backup

Payload System: CryptoMalware

- Cryptocurrency is a digital currency.
  - **Bitcoin** is the most well-known cryptocurrency, though **Monero** is also becoming increasingly popular among cybercriminals.
- **Cryptomining**, or cryptocurrency mining, is the process of creating a unit of cryptocurrency wherein "miners" solve complex mathematical equations in order to validate data blocks and add transaction details to a blockchain. This activity, which is legal, is rewarded by payment via cryptocurrency.
- **Cryptojacking**, sometimes called criminal **cryptomining**, is the unauthorized use of a person's or organization's computing resources to mine cryptocurrency.
- Crypto-malware, on the other hand, operates silently and surreptitiously in the background of the user's system. Unlike a ransomware attack that demands payment directly, the crypto-malware attacker hopes that the malicious code remains undetected as long as possible so that they can continue to mine cryptocurrency using the victim's device.



WannaMine process

Payload System: Physical Corruption or Damage

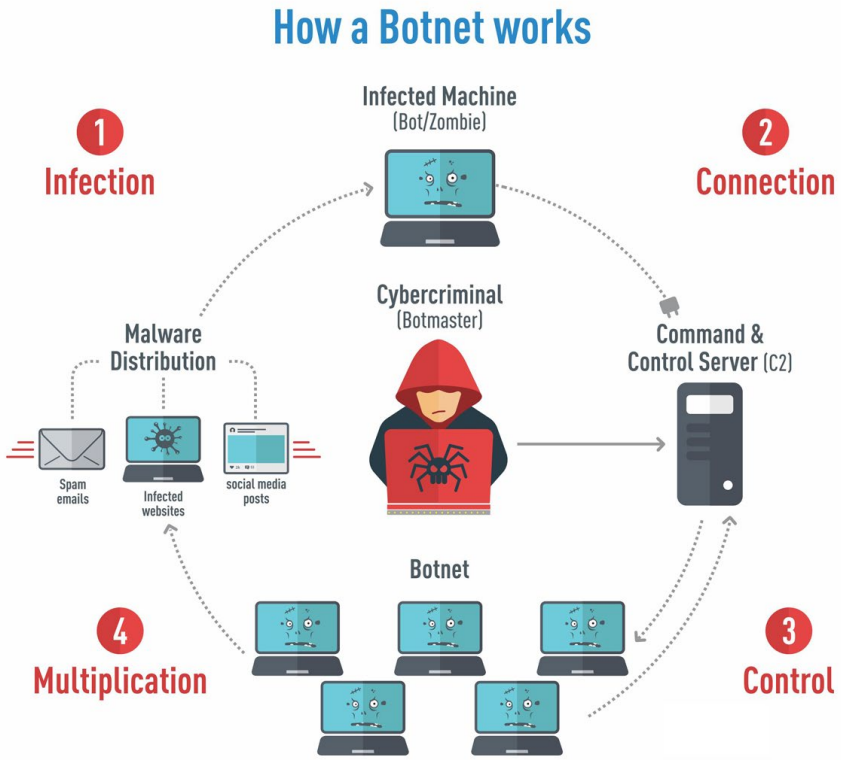
- Real-world damage
  - Causes damage to physical equipment
    - Chernobyl virus rewrites BIOS code, hence requiring to re-flash the BIOS of infected computer
  - Stuxnet worm (2011)
    - Targets specific industrial control system software
  - Physical damages started the "cyberwar" with industrial and political sabotage

Payload System: Logic Bomb

- Refers to the activation mechanism
  - The malware does nothing until some conditions are met
- Example: programmer of a payroll system introduces a logic bomb to crash the program if
  - the programmer is not paid (after 2 payments to others…),
  - generates problems that likely he/she can only fix, etc.
  - adds backdoors to the system (see later slides)

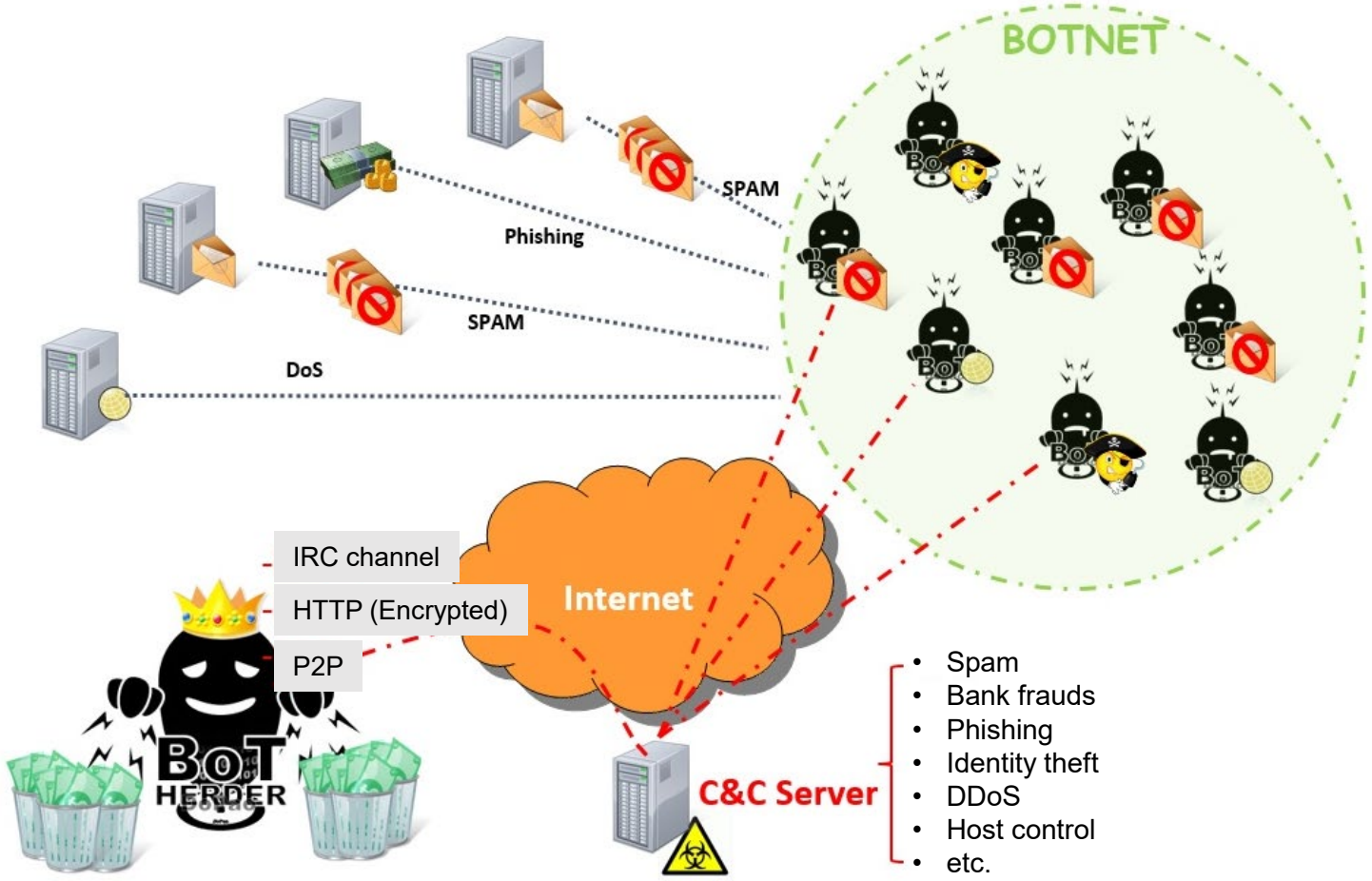
Payload System: Bots and Botnet

- A **bot** is a malware that takes over another Internet attached computer and uses that computer to launch or manage attacks
  - The bot is something called a **Zombie** host
- **Botnet** - collection of bots capable of acting in a coordinated manner
- Uses:
  - Distributed denial-of-service (DDoS) attacks
  - Spamming
  - Sniffing traffic
  - Keylogging
  - Spreading new malware
  - Installing advertisement add-ons and browser helper objects (BHOs)
  - Attacking IRC chat networks
  - Manipulating online polls/games
  - Mining BitCoins
- Two stages in parallel:
  1. BotNet creation: spreading like a worm
  2. BotNet command and control (**C2C**): control of hosts for action (next slide)



Remote Control Facility in a Botnet

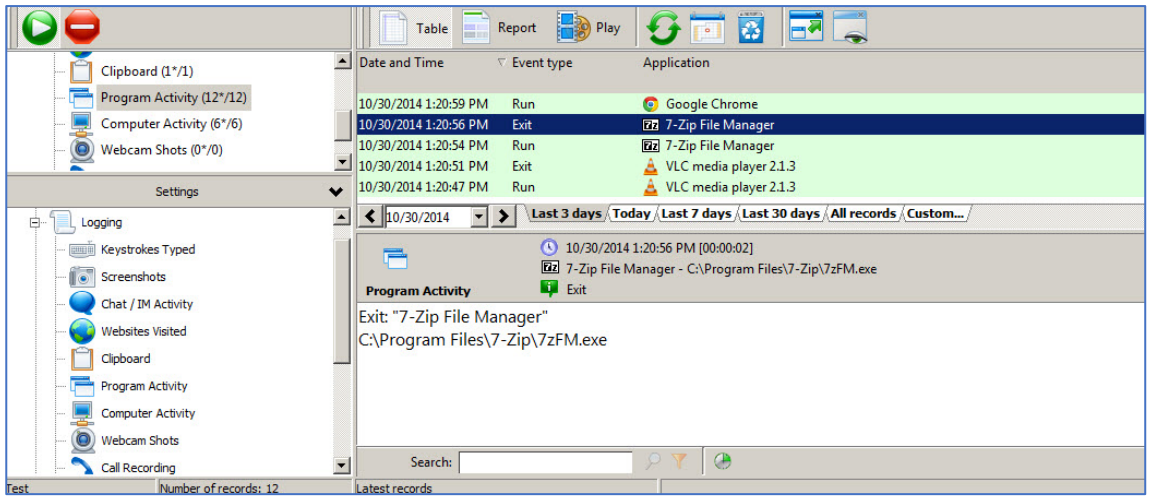
- Bot Vs Worm
  - Worm propagates itself and activates itself to be harmful (simpler payload)
  - Bot payload is controlled by some central facility (payload more complex)
- Typical means of implementing the remote-control facility:
  - IRC Channel: Bots join a specific channel on this server and treat incoming messages as commands
  - More recent botnets use covert communication channels via protocols such as HTTP
  - P2P protocols: Distributed control mechanisms use peer-to-peer protocols to avoid a single point of failure



Payload System: Information Theft with Keyloggers and Spyware

- **Credential Harvesting:** Malware might gather data stored on the infected system for use by the attacker
  - Login, user, passwords or
- **Data Exfiltration:** specific documents with reserved information (espionage) or systems configurations
  - e.g., Stuxnet and Siemens device configurations
- **Keylogger**
  - Captures keystrokes to allow attacker to monitor sensitive information
  - Typically uses some form of filtering mechanism that only returns information close to keywords ("login", "password")
- **Spyware**
  - Subverts the compromised machine to allow monitoring of a wide range of activity on the system
    - Monitoring history and content of browsing activity
    - Redirecting certain Web page requests to fake sites
    - Dynamically modifying data exchanged between the browser and certain Web sites of interest
    - Targeted Content

- BonziBuddy
  - Monitors user searches
  - Provides Targeted Ads
- Bearshare
  - SaveNow
    - Bundled with Bearshare
    - Collects User Information
    - Provides Targeted Ads
- Alexa Toolbar
  - Collects User Data
  - Provides Targeted Content



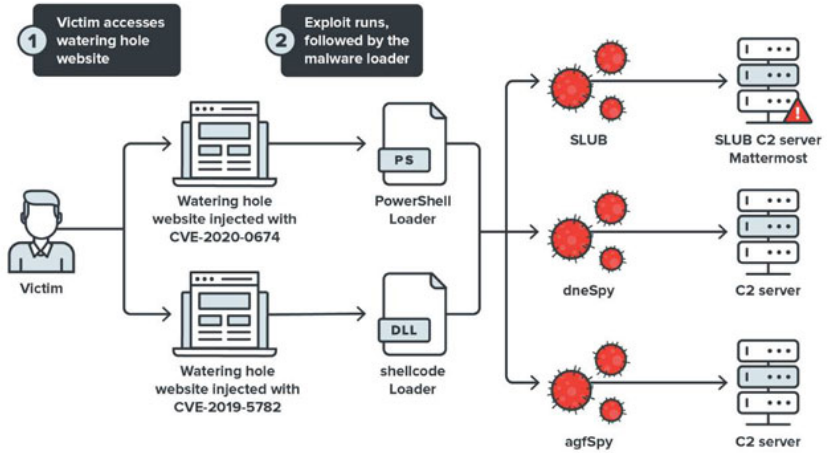
Payload System: Information Theft with Phishing

- Botnets and malwares in general can be part of a phishing and spear-phishing campaign.
  - Exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source
  - Include a URL in a spam e-mail that links to a fake Web site that mimics the login page of a banking, gaming, or similar site
  - Suggests that urgent action is required by the user to authenticate their account
  - Attacker exploits the account using the captured credentials

Date	Sending IP	Sending host name/info	Sender	Subject
2014-09-08 13:59 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	infonum@nordic-flair.com	Order is processed
2014-09-08 15:59 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	help@startcomputerrepair.com	The order #00354911 is ready
2014-09-08 16:35 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	security@acservicerepair.com	The order #00766530 is ready
2014-09-08 17:22 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	operator@thebookchair.com	Order NR00570345
2014-09-08 17:29 UTC	64.68.213.1	prisma-lan-64.68.213.1.bordercomm.com	verification@pinalcountyfair.com	Your order # NR00419810 has been completed
2014-09-08 20:21 UTC	201.130.71.170	host064170.metrored.net.mx	custservice@wholesaleindianhair.com	Your ticket #NR00111413
2014-09-09 04:07 UTC	202.126.172.110	unknown.telstraglobal.net	custservice@lakeunionhair.com	Please download your ticket
2014-09-09 13:16 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	customerssupport@highperformancemassair.com	Your order # ID16-00758758 has been completed
2014-09-09 13:52 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	reference@479landfair.com	Order NR00099546
2014-09-09 14:04 UTC	63.247.185.226	3ff7b9e2.cst.lightpath.net	infonum@nordic-flair.com	Order is processed
2014-09-09 15:38 UTC	63.124.7.24	US, Houston - MCI Communications, Verizon Business	support@chiefsappliancerepair.com	Order #00733903 is processed
2014-09-09 18:26 UTC	209.156.34.194	mail.strataproducts.com	support@cavestclair.com	Your order # ID16-00637196 has been completed

Payload System: Backdoors

- Also known as a **trapdoor**
- Secret entry point into a program allowing the attacker to gain access and bypass the security access procedures
- Installation methods:
  - A *maintenance hook* is a backdoor used by Programmers to debug and test programs
  - Trojans, phishing, waterhole attacks
- Once installed, very often used in C2C to be controlled from outside corporate networks by hackers
  - E.g., for controlling a ransomware or lateral movements



Example of installation using waterhole attacks

https://thehackernews.com/2020/10/browser-exploit-backdoor.html

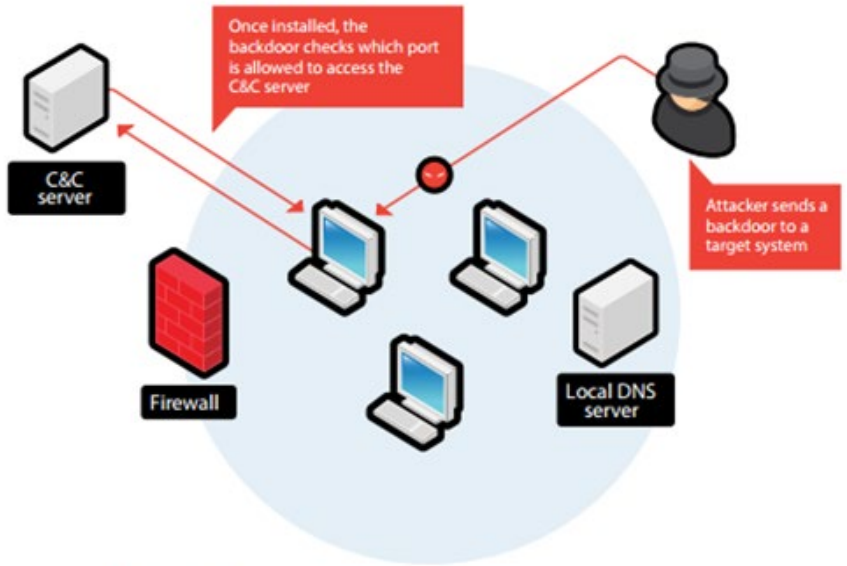
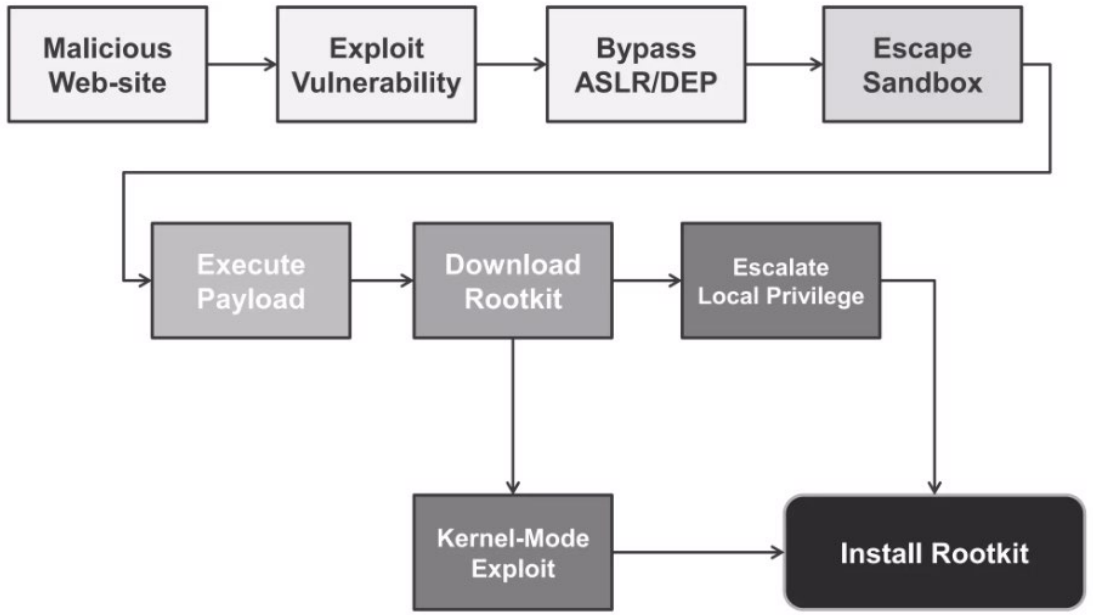


Figure 2: Typical targeted attack on a corporate network

Payload System: Rootkit (1)

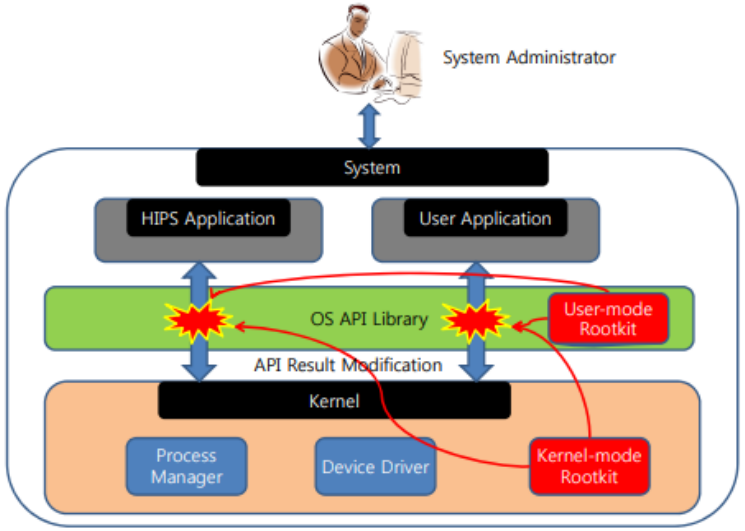
- Designed to enable access to a computer or an area of its software that is not otherwise allowed (for example, to an unauthorized user) and often masks its existence or the existence of other software
- Hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer
- Gives **administrator** (or **root** in Linux) privileges to attacker
  - Can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand
  - Sometimes used for benign or authorized activities:
    - to emulate hardware devices like DVD and CD roms
    - to bypass protections in CD/DVD/BD copy
    - to provide anti-theft protection installed in the BIOS
    - Etc.

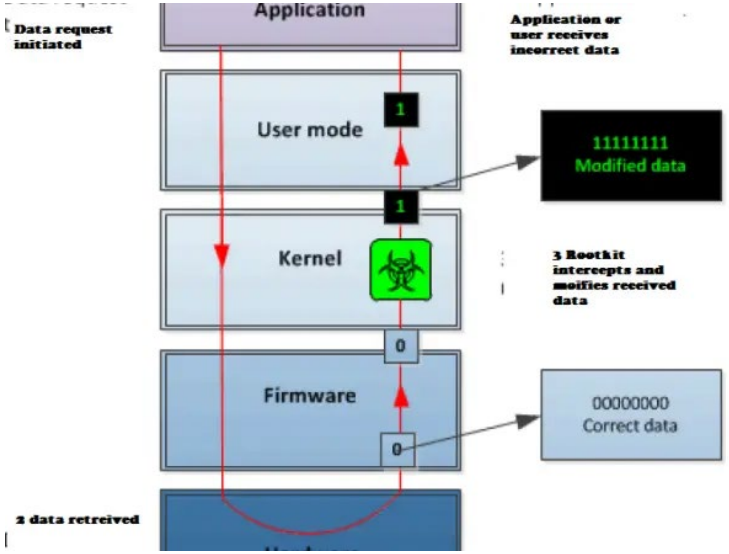


Payload System: Rootkit (2)

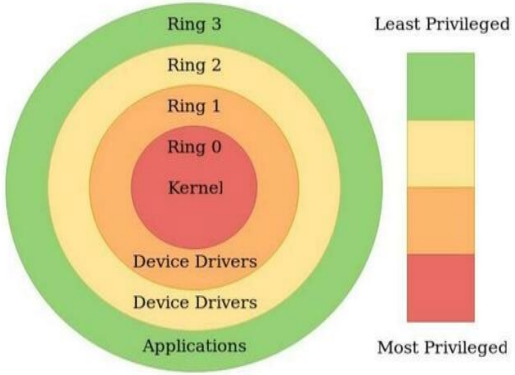
- Operation modes:
  - Persistent: stored in the registry or file-system, self-activated at boot
  - Memory based: stored in memory, hard to detect but needs to be re-downloaded at reboot
  - User mode: hides calls to APIs, for example to hide malicious files listed to the user
  - Kernel mode: hides information returning from kernel APIs, e.g., list of active processes in the system
  - Virtual machine based: it runs a minimal virtual machine hypervisor to hide a malicious virtual machine
  - External mode: installed in the BIOS or master boots (bootkit)

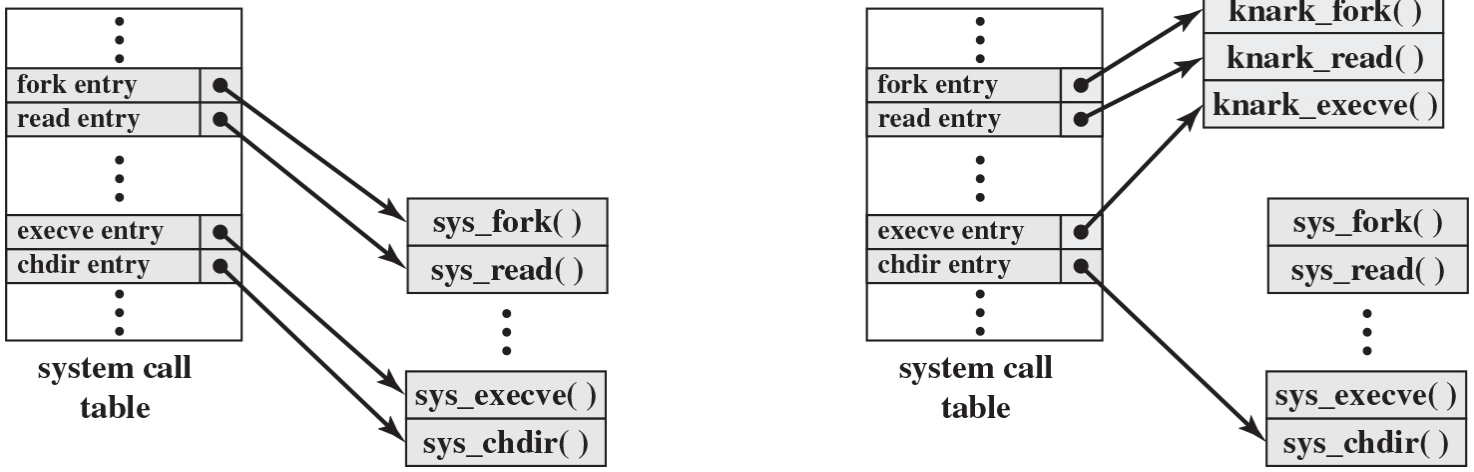
(HIPS) host-based intrusion prevention system: can detect user mode rootkits not kernel mode





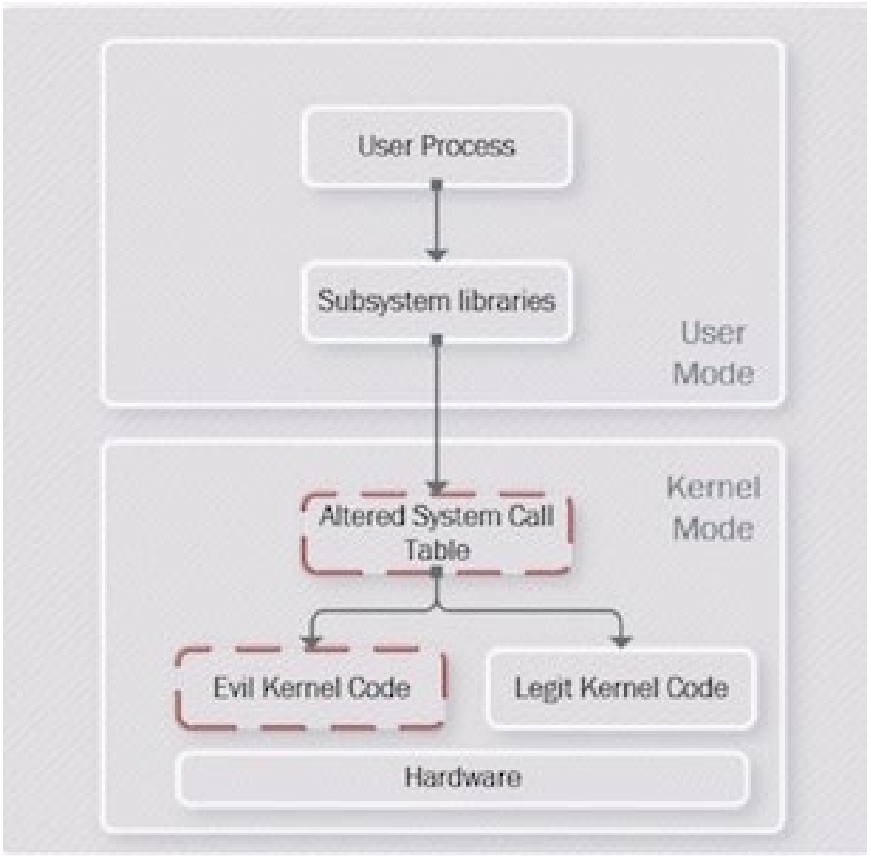
Payload System (3): Kernel Mode rootkit in Linux
Figure 6.3 System Call Table Modification by Rootkit





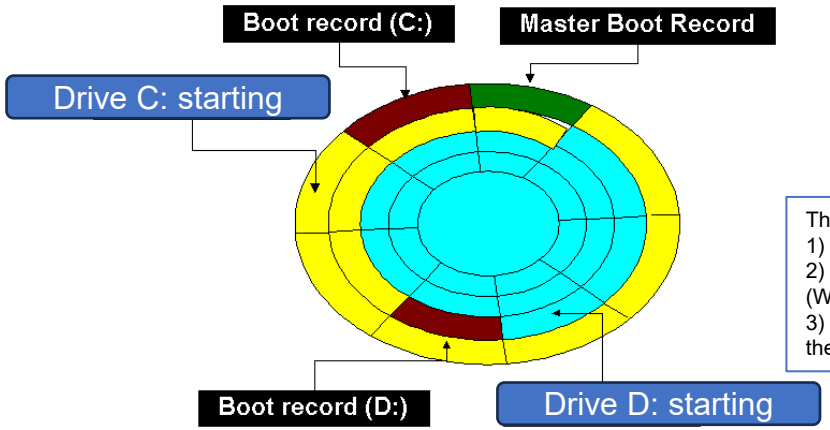
(a) Normal kernel memory layout

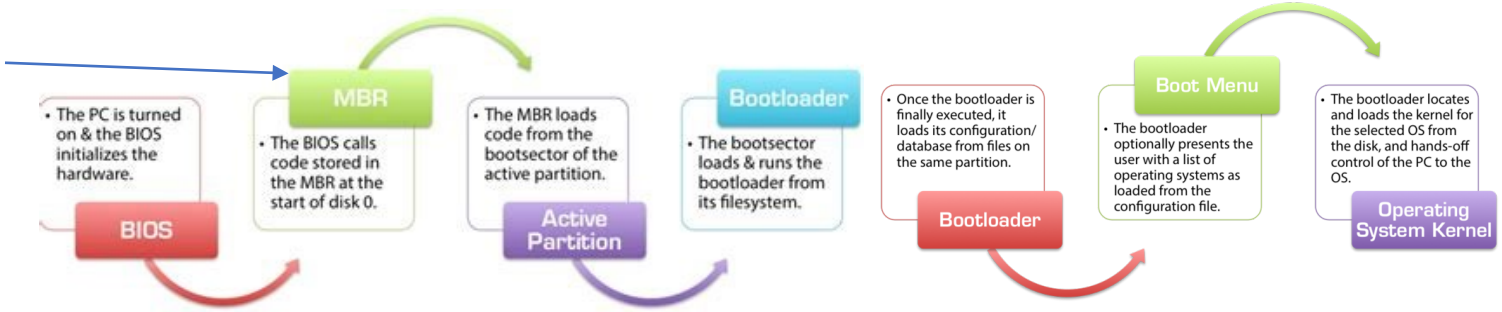
(b) After knark install



Note: System boot process

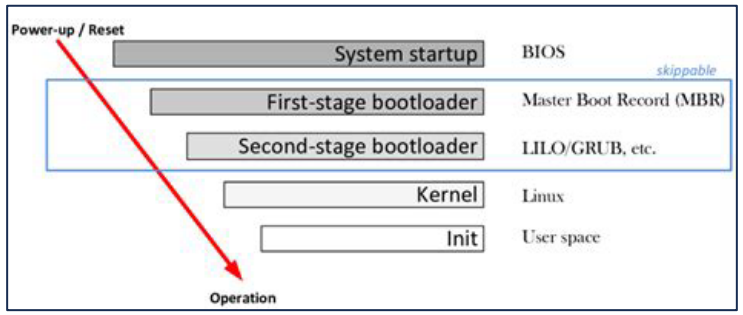
Master boot record is the first area checked in the disk from the BIOS





There are 3 systems executed in a chain to start the Operating System (boot sequence):
1) the first is a firmware in your motherboard (BIOS is old or UEFI is new). The objective of this program is to check hardware resources and start the first boot sector.
2) then, a minimal program is loaded (with a minimal set of tools and libraries), called bootloader: very common are GRUB and Bootmgr.exe (Windows with BIOS) or BootMgrfw.efi (Windows with UEFI)
3) then, after the bootloader at previous step, the actual Operating System is loaded (e.g., Windows 11 or Linux Ubuntu). Each OS first loads the kernel/main functions and then the user space.

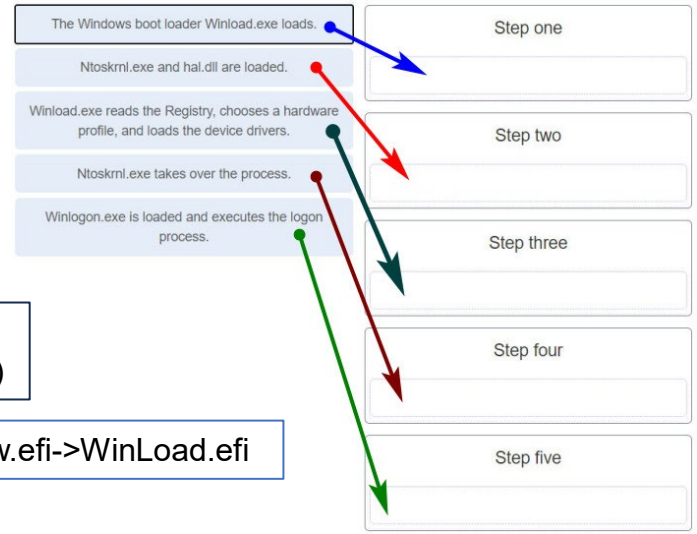
In Linux (from BIOS)



LILO/GRUB:
Bootloaders from Linux community

Bootmgr.exe:
Bootloader from Microsoft for Win systems (after Win Vista)

In Windows, after Bootmgr.exe
(BIOS->MBR->Bootmgr.exe)

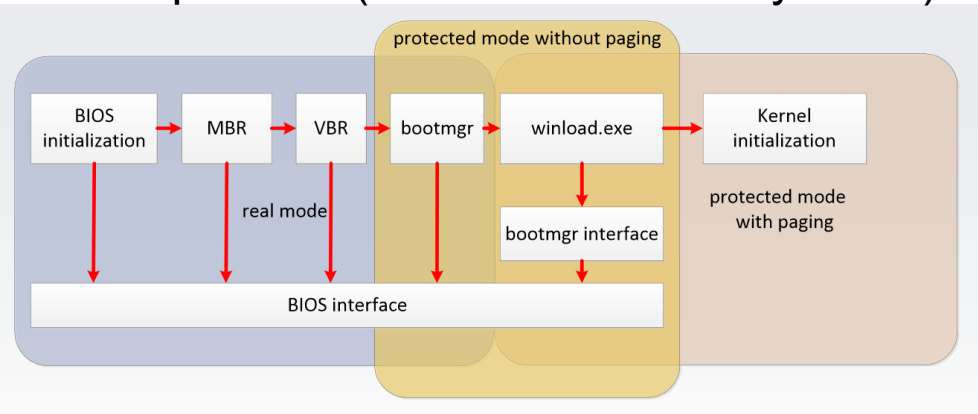


Note: You can also have both systems (e.g., BIOS->MBR->GRUB->(Win or Linux))

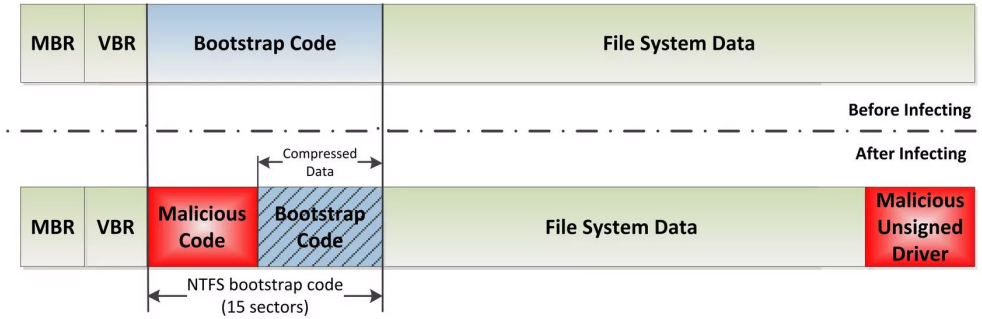
NOTE 2: in UEFI is UEFI->GPT-> EFI/Microsoft/Boot/bootmgfw.efi->WinLoad.efi

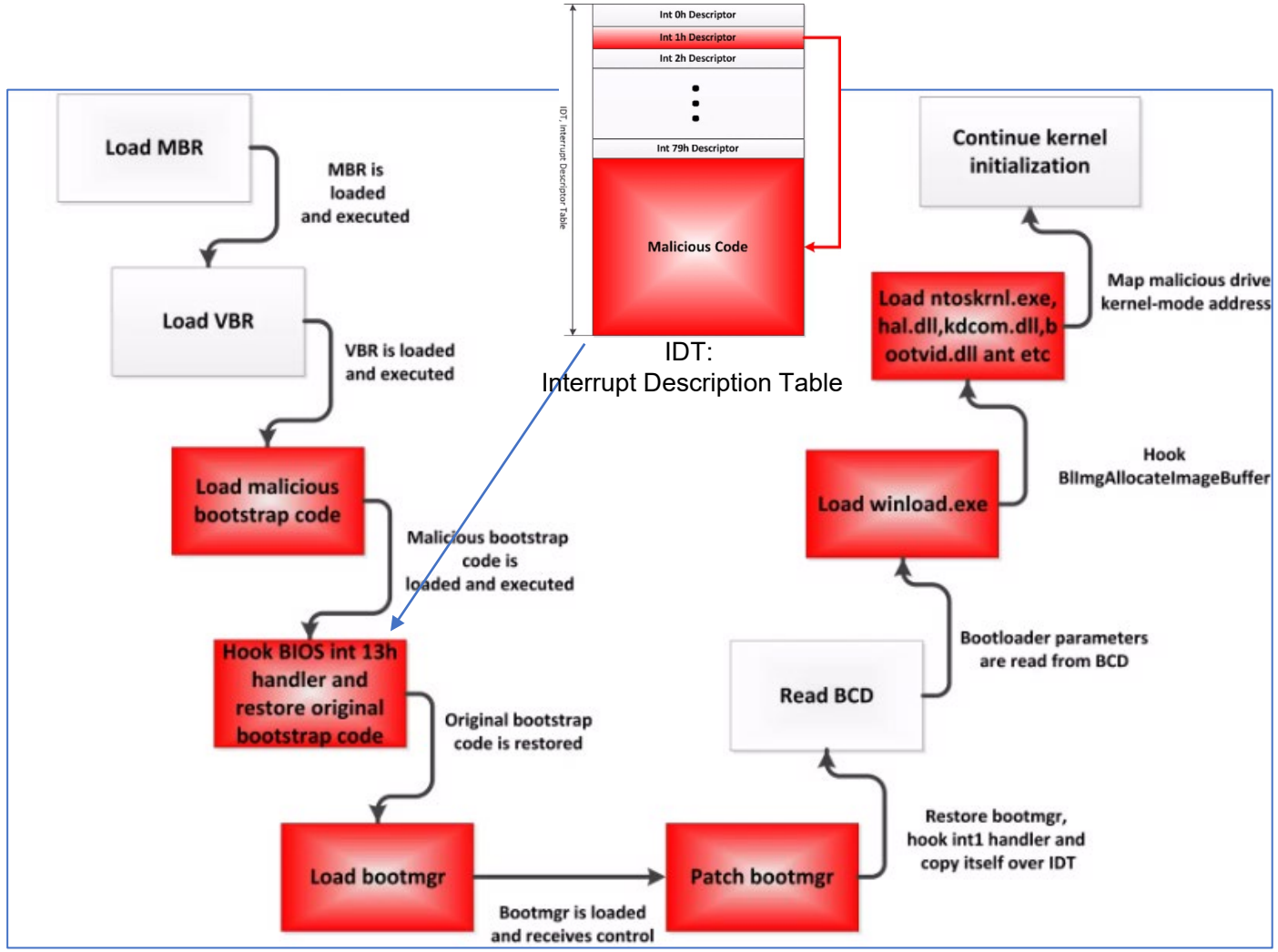
Payload System: Rootkit in boot sector aka Bootkit (4)

Boot process (BIOS + Windows systems)



MBR: master boot record
VBR: volume boot record
bootmgr: windows boot manager (bootloader in Windows)
BCD: boot configuration data

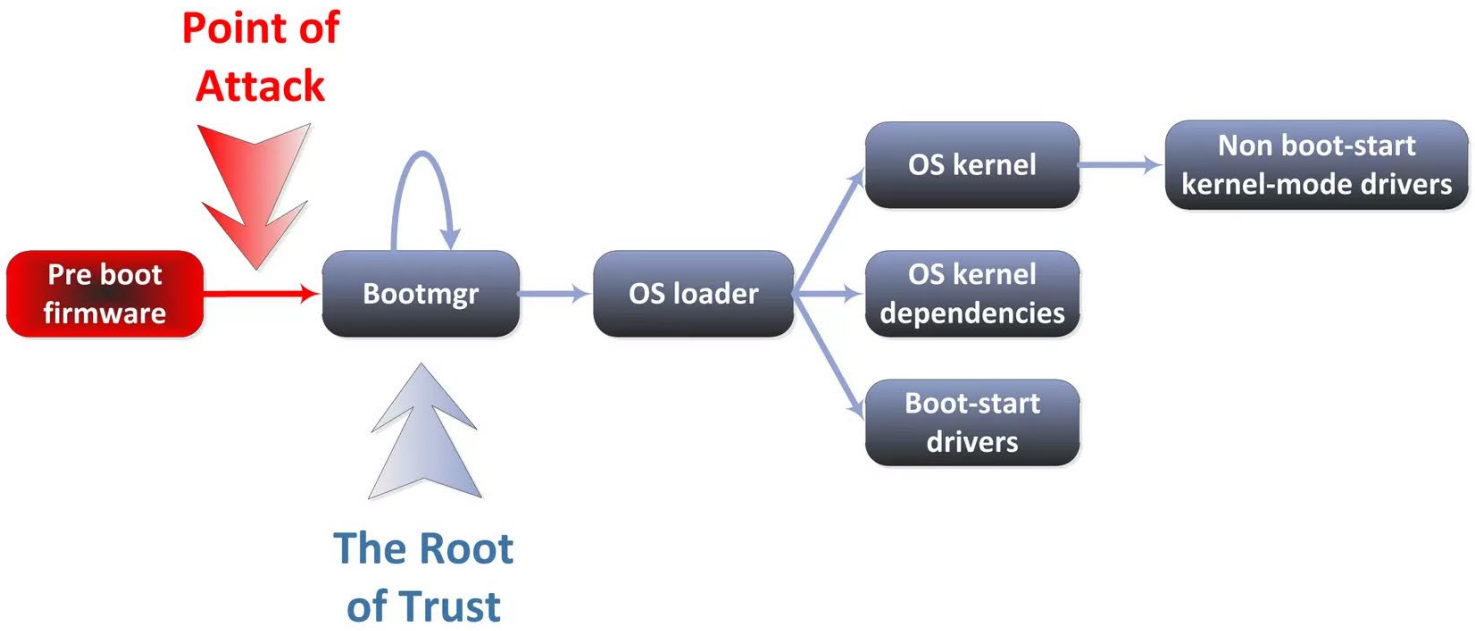




Example from Win64/Rovnix rootkit

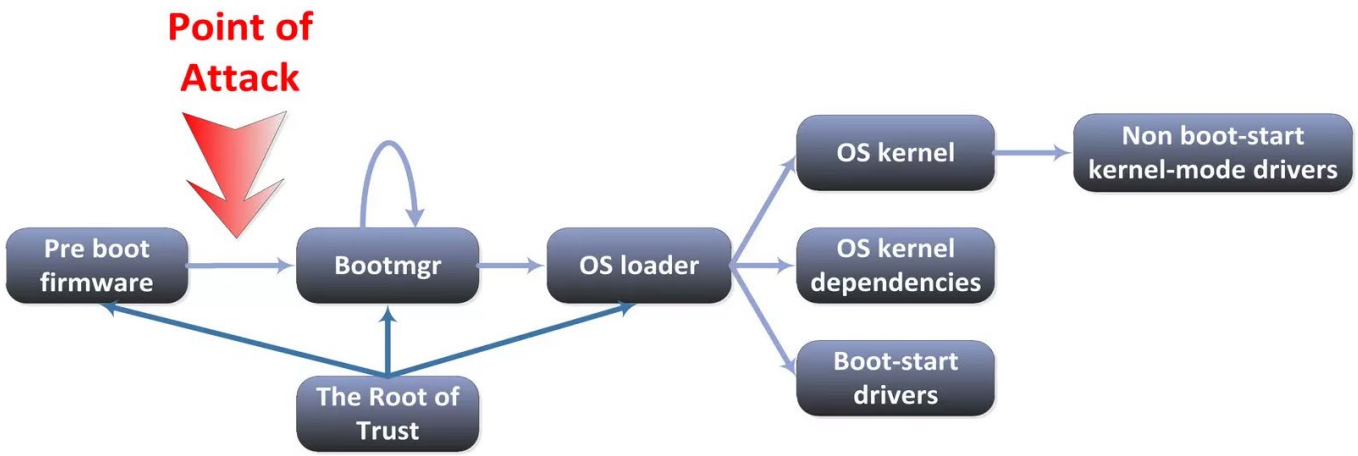
Issue with Bootkits in traditional BIOS (5)

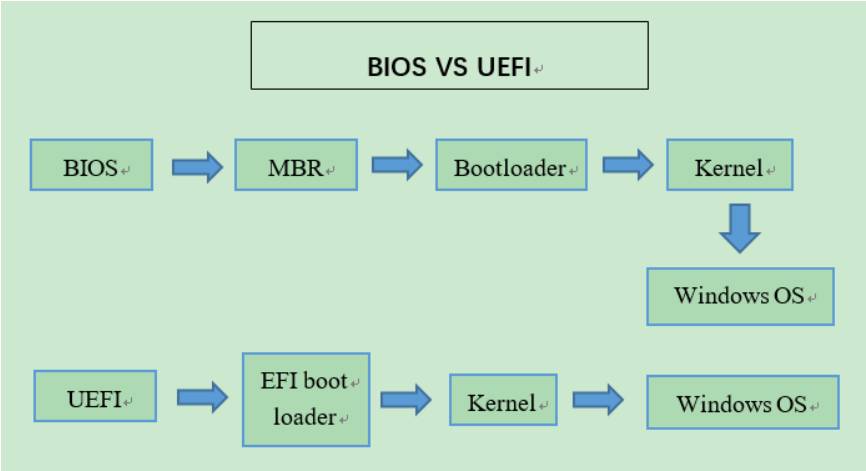
- Untrusted platform problem
  - ✓ BIOS controls boot process, but who controls it?
  - ✓ The trust of trust is below point of attack



Protection from Bootkit: Moving the root of trust (6)

- To resist bootkit attacks we need the root of trust be above point of attack:
  - ✓ TPM
  - ✓ UEFI Secure Boot





UEFI rootkits still possible: article here

Payload System: Fileless malwares and living off the land (LOLbins) malwares

- Fileless malwares attacks are similar to traditional viruses in a number of critical ways. They spread via methods like spam email and malicious websites, and they exploit flaws in browser plug-ins and web browsers themselves. Once they successfully find a way into a system, they inject themselves into memory and conduct further malicious activity, including adding the ability to reinfect the system by the same process at reboot through a registry entry or other technique. At no point do they require local file storage, because they remain memory resident throughout their entire active life—in fact, the only stored artifact of many fileless attacks would be the artifacts of their persistence techniques, like the registry entry shown in Figure.

- Fileless attacks require a vulnerability to succeed, so ensuring that browsers, plug-ins, and other software that might be exploited by attackers are up to date and protected can prevent most attacks.

Other notes about malware differences

- https://sec.cloudapps.cisco.com/security/center/resources/virus_differences
- https://pages.cs.wisc.edu/~jha/jha-papers/security/usenix_2003.pdf