LA NUEVA ERA DE LOS BOOTKITS: PERSISTENCIA Y EVASIÓN DESDE EL NÚCLEO

Descifrando el desarrollo de Bootkits UEFI para Windows 10 y 11

[in/vazquez-vazquez-alejandro]

VICON 2024, Galicia

AGENDA

- Conceptos
  - Bootkits, Rootkits
- Protecciones
  - DSE, SecureBoot
- Proceso de Arranque
  - bootmgfw.efi, winload.efi, ntoskrnl.exe
- Bootkit
  - UEFI Application, Kernel-Mode Driver
- Entorno
  - WDK, EDK2
- Demo
  - ???
- Desarrollo
  - MalwareUnderRadarPKG
- Malware
  - Glupteba, Bootkit UEFI
- Bootkits in the Wild
  - FinFisher, SPEcter, BlackLotus
- Máster y Recursos
  - Reversing y Análisis de Malware

WHOAMI

- **FRIKI** (**F**anático de **R**evolucionar **I**nternamente **K**ernels e **I**nicios de sistema)
- Pastor de ovejas desde los 8 años
- Me gusta el pulpo, de ahí los Bootkits
- Hobbie 1: Desarrollo de Malware
- Hobbie 2: Ciberseguridad Ofensiva
- Docente en Máster de Reversing y Análisis de Malware

[in/vazquez-vazquez-alejandro]

SENSITIVE CONTENT

Age Verification

This presentation contains age-restricted materials including malware and explicit hooking techniques. By entering, you affirm that you are at least 18 years of age and you consent to viewing “hacker” stuff.

Let me In
This is real stuff

No
I prefer OSINT

CONCEPTOS

- Bootkit: Malicious program designed to load as early as possible in the boot process, in order to control all stages of the operating system start up, modifying system code and drivers before security components are loaded. ~ Kaspersky

- Rootkit: Sophisticated piece of malware that can add new code to the operating system or delete and edit operating system code. Rootkits may remain in place for years because they are hard to detect, due in part to their ability to block some antivirus software and malware scanner software. ~ Crowdstrike

CONCEPTOS

- Bootkit: Malicious program designed to load as early as possible in the boot process, in order to control all stages of the operating system start up, modifying system code and drivers before security components are loaded. ~ Kaspersky

UEFI Application → **C/C++ - boot.efi**

- Rootkit: Sophisticated piece of malware that can add new code to the operating system or delete and edit operating system code. Rootkits may remain in place for years because they are hard to detect, due in part to their ability to block some antivirus software and malware scanner software. ~ Crowdstrike

Kernel-Mode Driver → **C/C++ - driver.sys**

PROTECCIONES

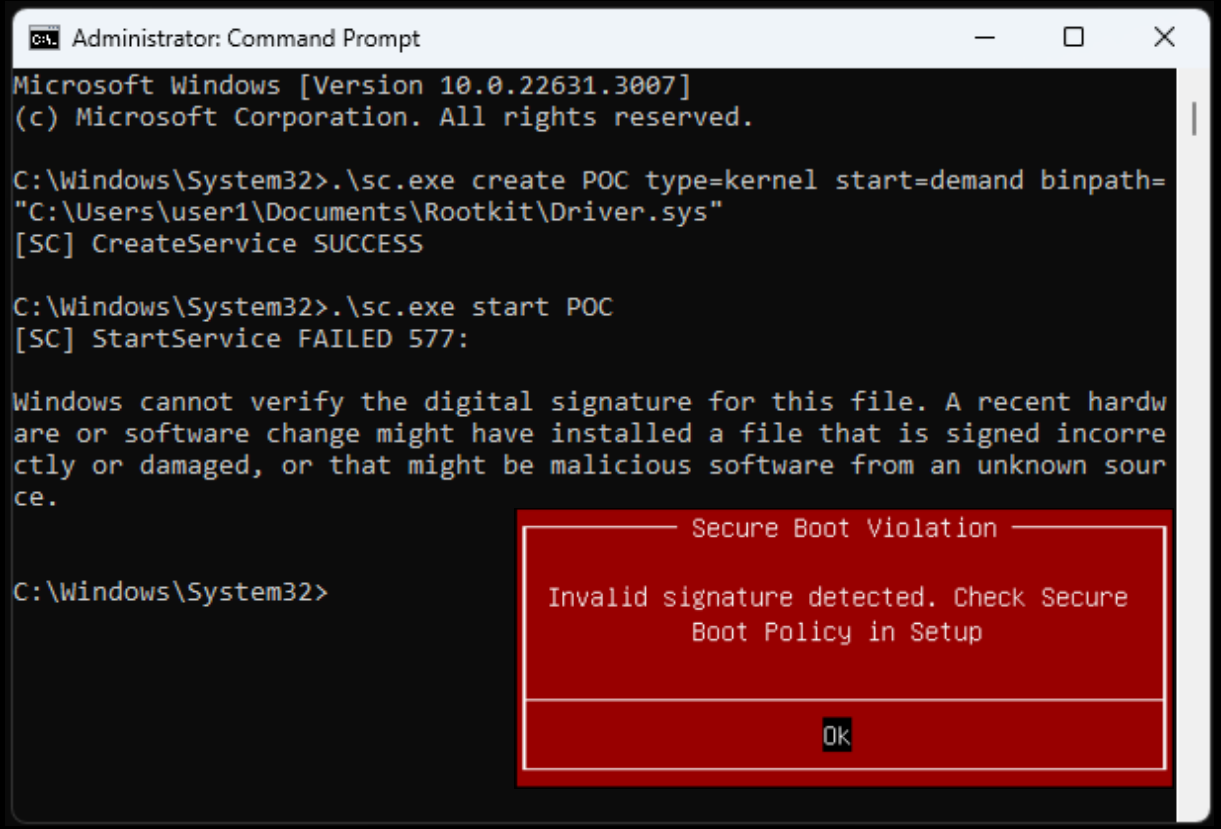
- Driver Signature Enforcement (DSE)

  Windows won't run drivers not certified by Microsoft

- SecureBoot

  Only software trusted by the Original Manufacturer.

  Firmware checks the signature of UEFI firmware drivers, EFI applications and SO



Power ON

Read Instructions

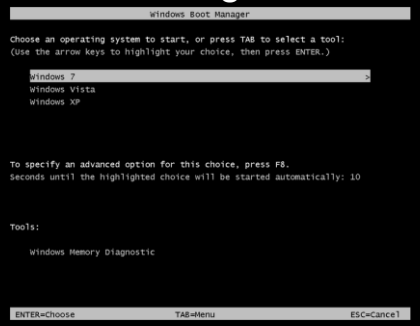
POST

UEFI Firmware

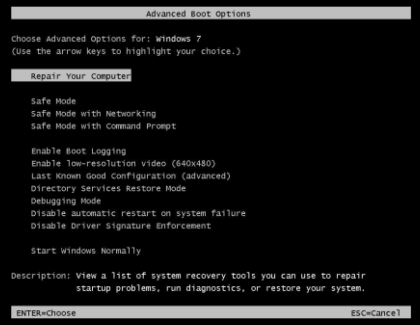
Boot Information

Boot order
Boot0001 = /EFI/Microsoft/boot/bootmgfw.efi
Boot0002 = /EFI/Ubuntu/shimx64.efi
Boot000x = /EFI/Vendor/bootx64.efi

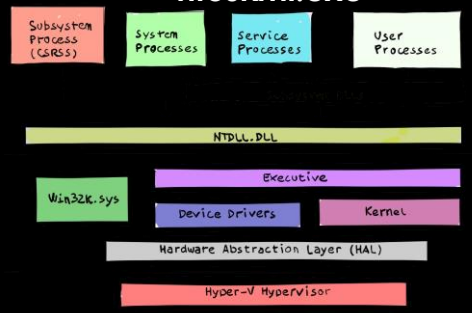
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi

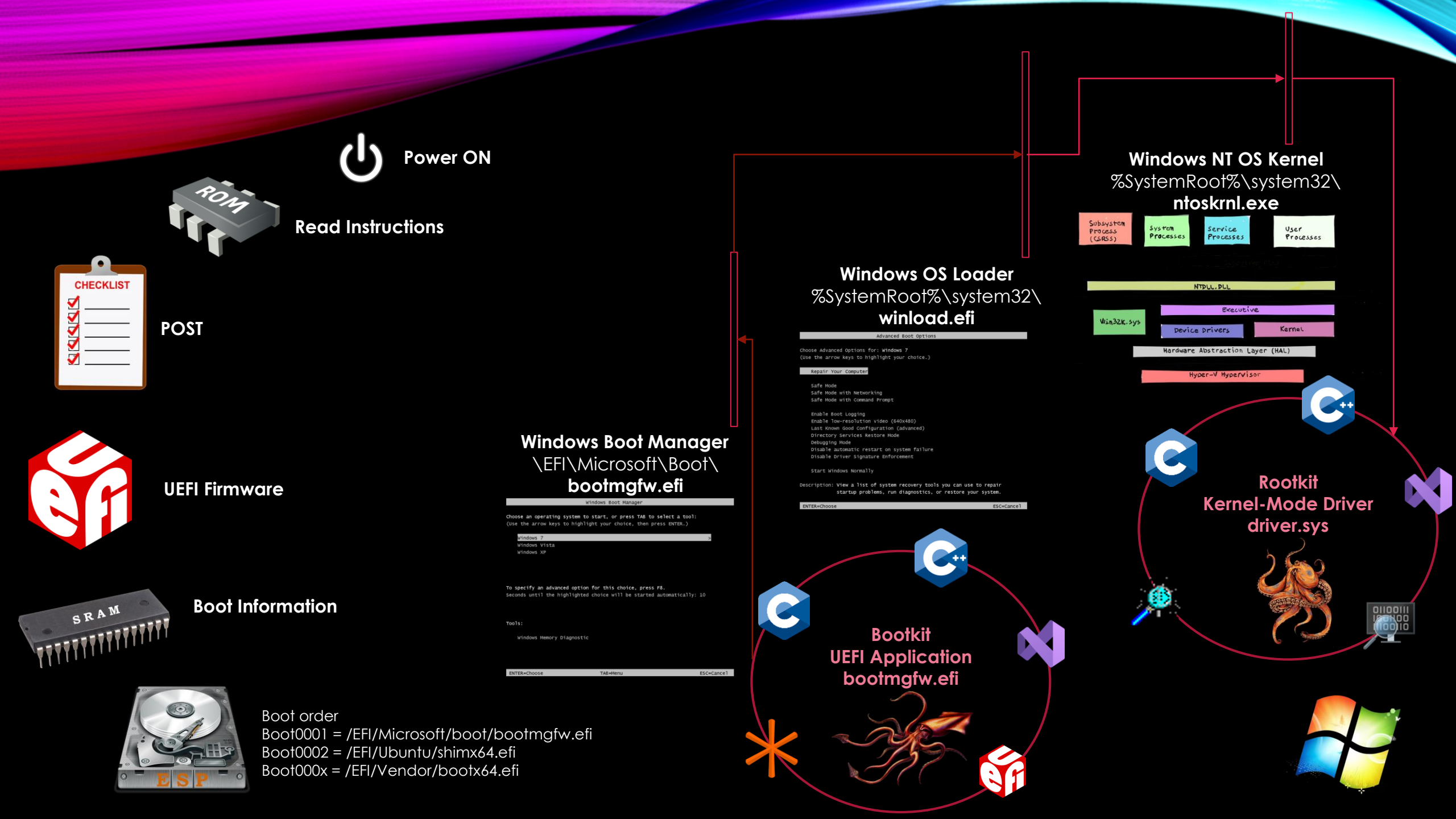


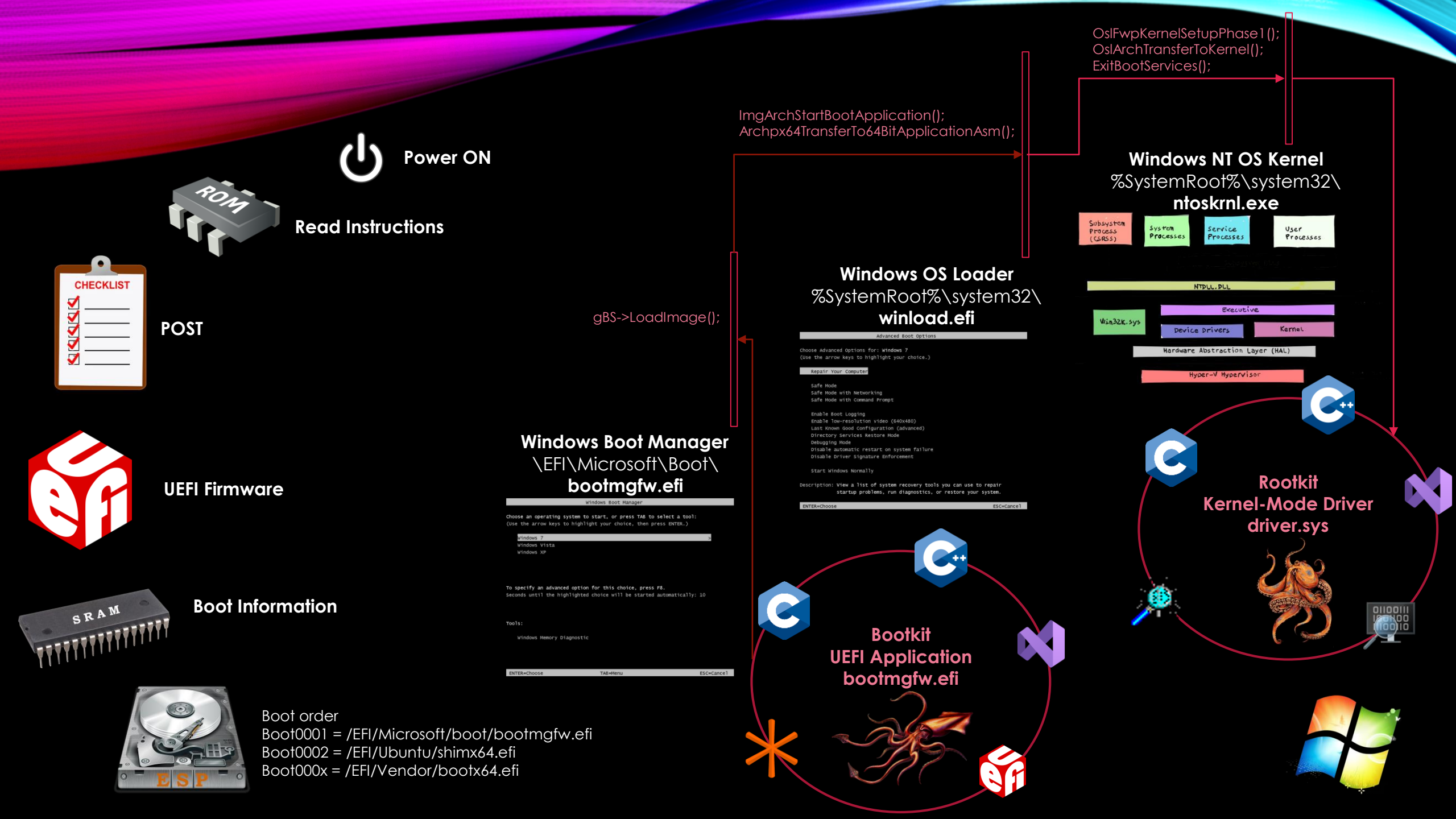
Windows OS Loader
%SystemRoot%\system32\
winload.efi

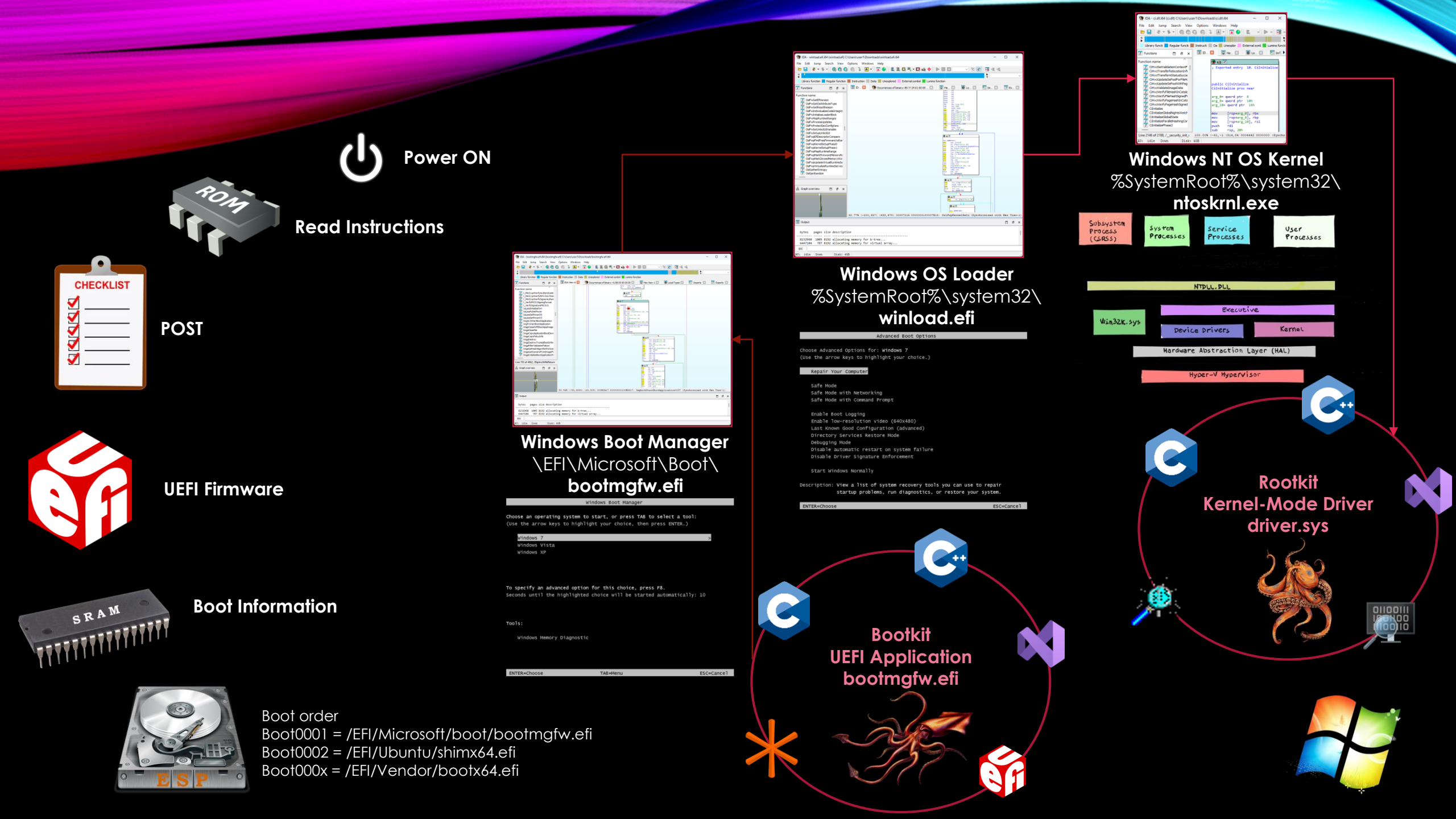


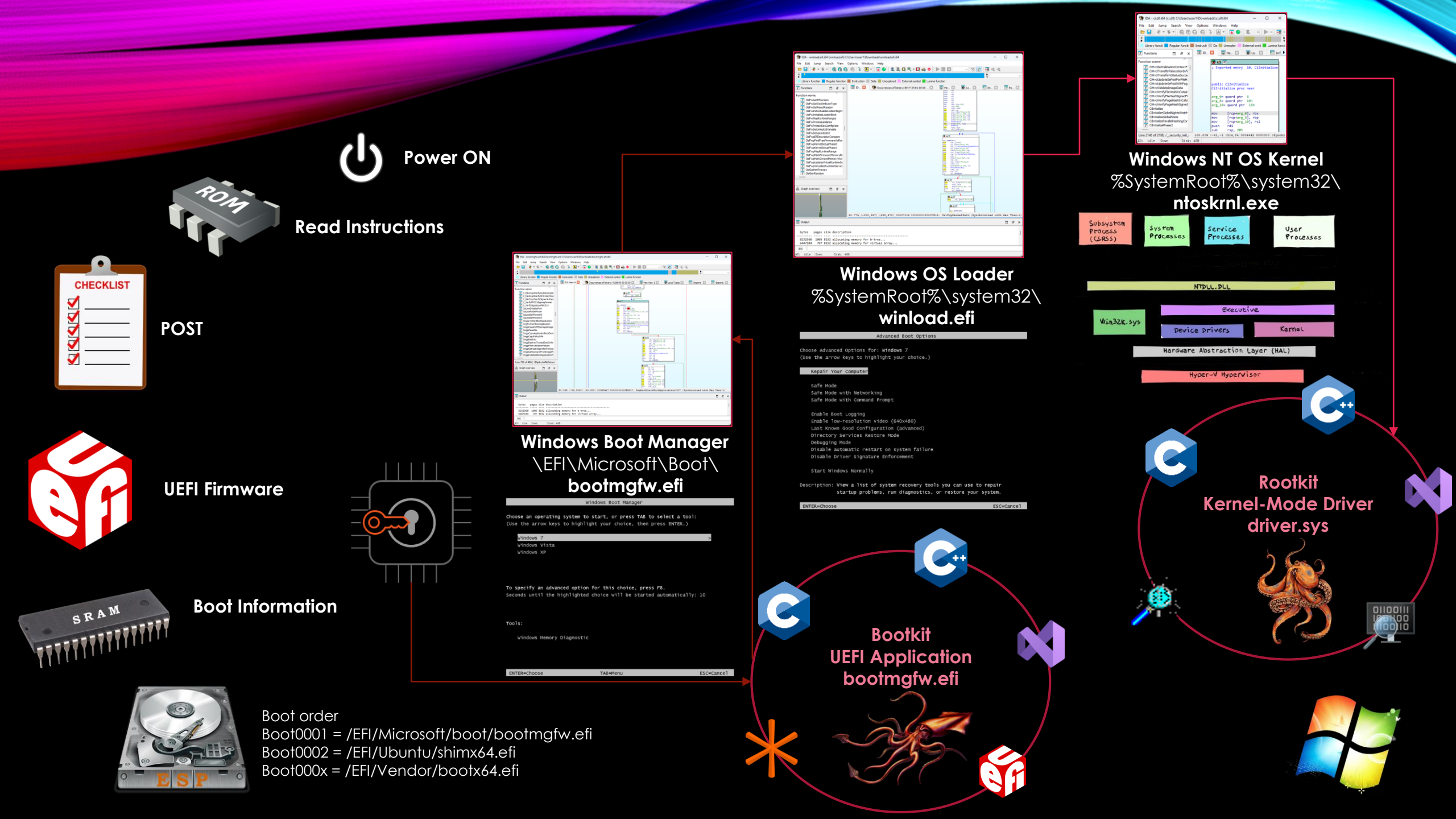
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe




Power ON
ROM
Read Instructions
CHECKLIST
POST
UEFI Firmware
SRAM
Boot Information
ESP
Boot order
Boot0001 = /EFI/Microsoft/boot/bootmgfw.efi
Boot0002 = /EFI/Ubuntu/shimx64.efi
Boot000x = /EFI/Vendor/bootx64.efi
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi
Windows Boot Manager
Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)
Windows 7
Windows Vista
Windows XP
To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10
Tools:
Windows Memory Diagnostic
ENTER=Choose
TAB=Menu
ESC=Cancel
Windows OS Loader
%SystemRoot%\system32\
winload.efi
Advanced Boot Options
Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)
Repair Your Computer
Safe Mode
Safe Mode with Networking
Safe Mode with Command Prompt
Enable Boot Logging
Enable low-resolution video (640x480)
Last Known Good Configuration (advanced)
Directory Services Restore Mode
Debugging Mode
Disable automatic restart on system failure
Disable Driver Signature Enforcement
Start Windows Normally
Description: View a list of system recovery tools you can use to repair
startup problems, run diagnostics, or restore your system.
ENTER=Choose
ESC=Cancel
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe
Subsystem Process (CSRSS)
System Processes
Service Processes
User Processes
NTDLL.DLL
Win32k.sys
Executive
Device Drivers
Kernel
Hardware Abstraction Layer (HAL)
Hyper-V Hypervisor
Bootkit
UEFI Application
bootmgfw.efi
Rootkit
Kernel-Mode Driver
driver.sys


Power ON
ROM
Read Instructions
CHECKLIST
POST
UEFI Firmware
SRAM
Boot Information
ESP
Boot order
Boot0001 = /EFI/Microsoft/boot/bootmgfw.efi
Boot0002 = /EFI/Ubuntu/shimx64.efi
Boot000x = /EFI/Vendor/bootx64.efi
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi
Windows Boot Manager
Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)
Windows 7
Windows Vista
Windows XP
To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10
Tools:
Windows Memory Diagnostic
ENTER=Choose
TAB=Menu
ESC=Cancel
gBS->LoadImage();
ImgArchStartBootApplication();
Archpx64TransferTo64BitApplicationAsm();
Windows OS Loader
%SystemRoot%\system32\
winload.efi
Advanced Boot Options
Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)
Repair Your Computer
Safe Mode
Safe Mode with Networking
Safe Mode with Command Prompt
Enable Boot Logging
Enable low-resolution video (640x480)
Last Known Good Configuration (advanced)
Directory Services Restore Mode
Debugging Mode
Disable automatic restart on system failure
Disable Driver Signature Enforcement
Start Windows Normally
Description: View a list of system recovery tools you can use to repair
startup problems, run diagnostics, or restore your system.
ENTER=Choose
ESC=Cancel
Bootkit
UEFI Application
bootmgfw.efi
OslFwpKernelSetupPhase1();
OslArchTransferToKernel();
ExitBootServices();
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe
Subsystem Process (CSRSS)
System Processes
Service Processes
User Processes
NTDLL.DLL
Win32k.sys
Executive
Device Drivers
Kernel
Hardware Abstraction Layer (HAL)
Hyper-V Hypervisor
Rootkit
Kernel-Mode Driver
driver.sys


Power ON
ROM
Read Instructions
CHECKLIST
POST
UEFI Firmware
SRAM
Boot Information
ESP
Boot order
Boot0001 = /EFI/Microsoft/boot/bootmgfw.efi
Boot0002 = /EFI/Ubuntu/shimx64.efi
Boot000x = /EFI/Vendor/bootx64.efi
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi
Windows Boot Manager
Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)
Windows 7
Windows Vista
Windows XP
To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10
Tools:
Windows Memory Diagnostic
ENTER=Choose
TAB=Menu
ESC=Cancel
Windows OS Loader
%SystemRoot%\system32\
winload.efi
Advanced Boot Options
Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)
Repair Your Computer
Safe Mode
Safe Mode with Networking
Safe Mode with Command Prompt
Enable Boot Logging
Enable Low-resolution Video (640x480)
Last Known Good Configuration (Advanced)
Directory Services Restore Mode
Debugging Mode
Disable automatic restart on system failure
Disable Driver Signature Enforcement
Start Windows Normally
Description: View a list of system recovery tools you can use to repair
startup problems, run diagnostics, or restore your system.
ENTER=Choose
ESC=Cancel
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe
Subsystem Process (CSRSS)
System Processes
Service Processes
User Processes
NTDLL.DLL
Win32K.sys
Executive
Device Drivers
Kernel
Hardware Abstraction Layer (HAL)
Hyper-V Hypervisor
Bootkit
UEFI Application
bootmgfw.efi
Rootkit
Kernel-Mode Driver
driver.sys

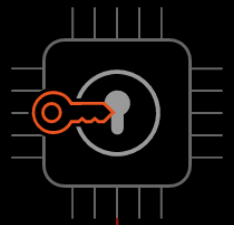

Power ON
ROM
Read Instructions
CHECKLIST
POST
UEFI Firmware
SRAM
Boot Information
ESP
Boot order
Boot0001 = /EFI/Microsoft/boot/bootmgfw.efi
Boot0002 = /EFI/Ubuntu/shimx64.efi
Boot000x = /EFI/Vendor/bootx64.efi
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi
Windows Boot Manager
Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)
Windows 7
Windows Vista
Windows XP
To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10
Tools:
Windows Memory Diagnostic
ENTER=Choose
TAB=Menu
ESC=Cancel
Windows OS Loader
%SystemRoot%\system32\
winload.efi
Advanced Boot Options
Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)
Repair Your Computer
Safe Mode
Safe Mode with Networking
Safe Mode with Command Prompt
Enable Boot Logging
Enable low-resolution video (640x480)
Last Known Good Configuration (advanced)
Directory Services Restore Mode
Debugging Mode
Disable automatic restart on system failure
Disable Driver Signature Enforcement
Start Windows Normally
Description: View a list of system recovery tools you can use to repair
startup problems, run diagnostics, or restore your system.
ENTER=Choose
ESC=Cancel
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe
Subsystem Process (CSRSS)
System Processes
Service Processes
User Processes
NTDLL.DLL
Win32K.sys
Executive
Device Drivers
Kernel
Hardware Abstraction Layer (HAL)
Hyper-V Hypervisor
Rootkit
Kernel-Mode Driver
driver.sys
Bootkit
UEFI Application
bootmgfw.efi

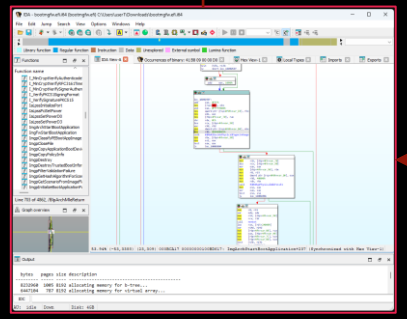
ENTORNO

- WDK
  Develop, test and deploy drivers for Windows.
  Software toolset from Microsoft.

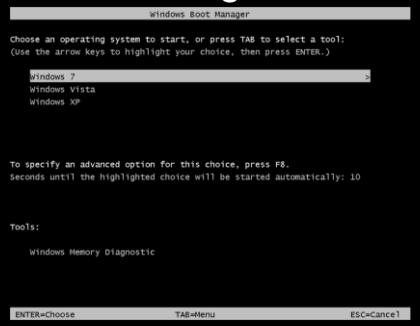
- EDK2
  Official development environment for UEFI applications, UEFI Drivers, DXE Drivers.
  Developed by the open-source Tianocore project (Intel, HP and Microsoft)
  Full on implementation of the UEFI specification.

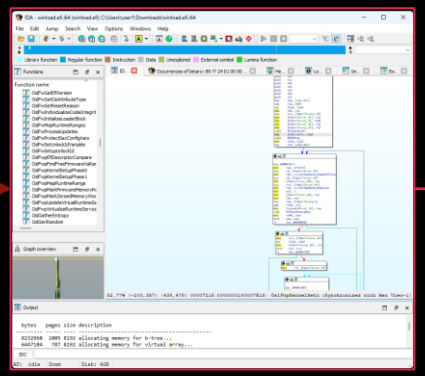
- +
  Visual Studio, C/C++, Reverse Engineering, WinDbg, IDA



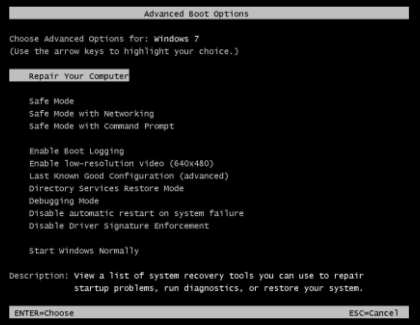


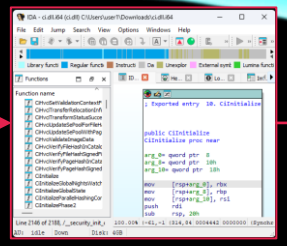
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi



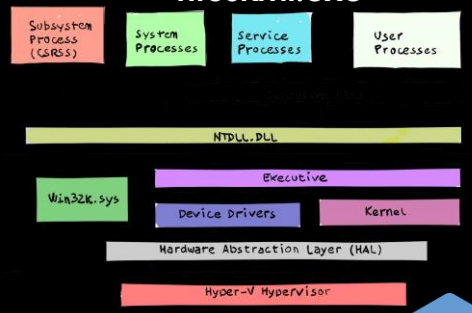


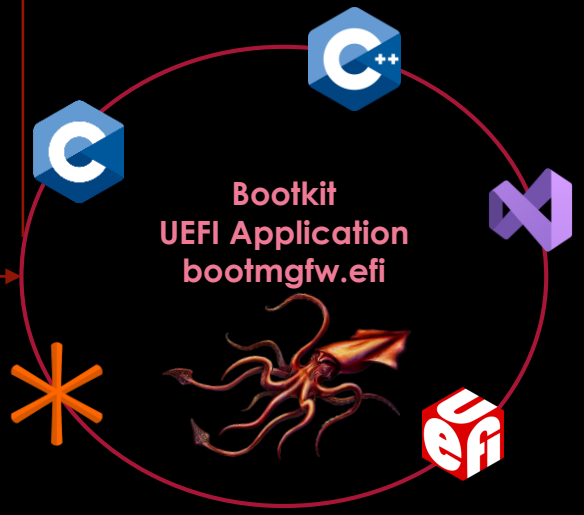
Windows OS Loader
%SystemRoot%\system32\
winload.efi

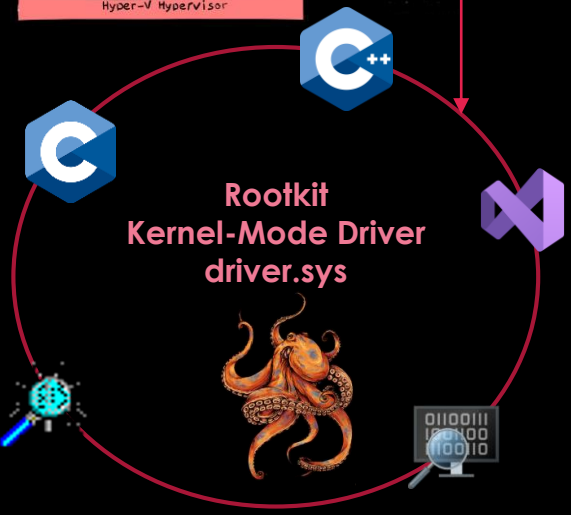




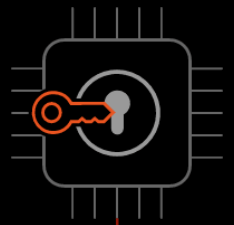
Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe

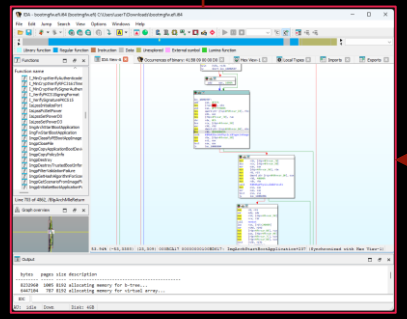




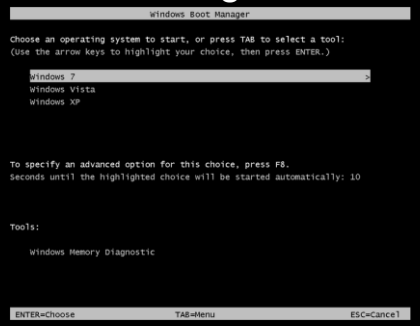


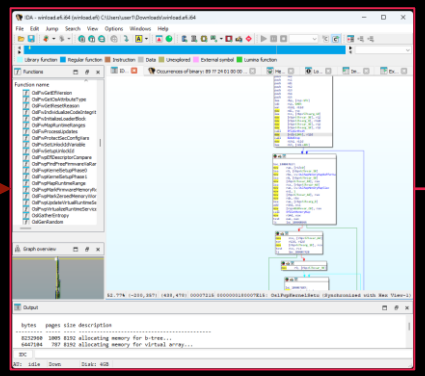

DEMO
LOADIND...



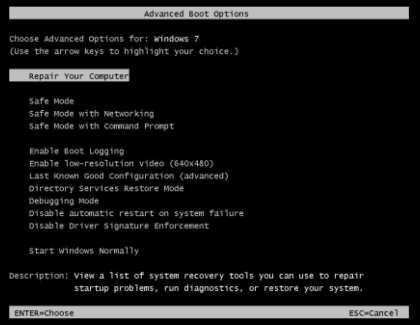


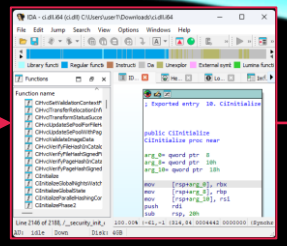
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi



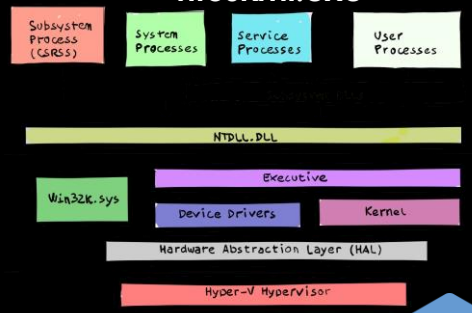


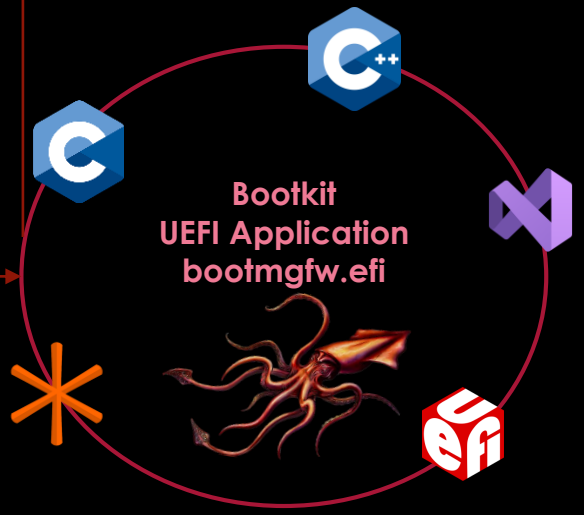
Windows OS Loader
%SystemRoot%\system32\
winload.efi

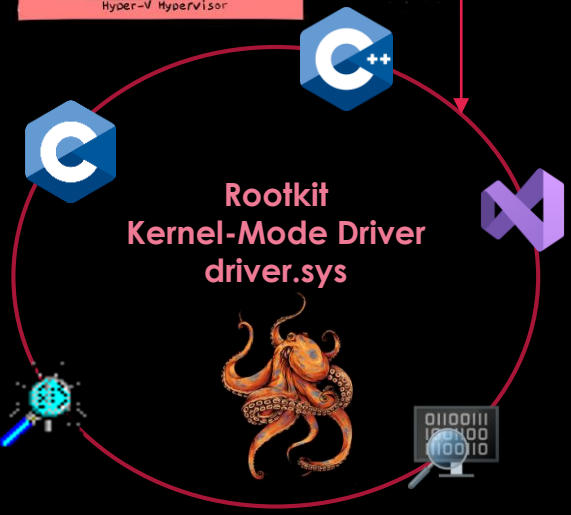




Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe



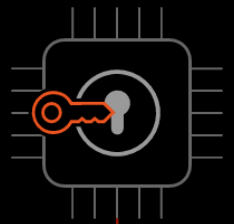


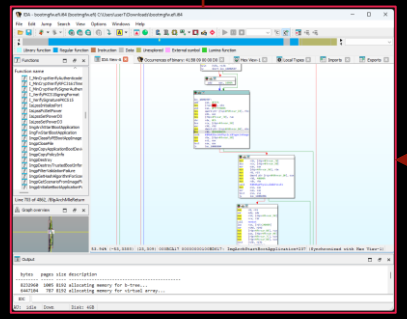


UEFI Specification

2.10

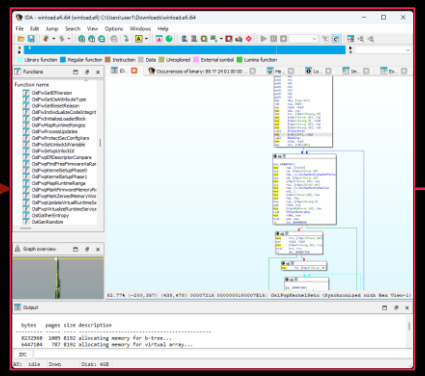
Search docs





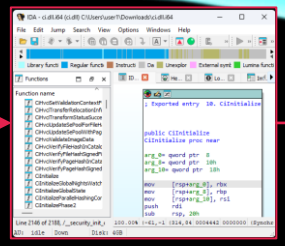
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi

```
Windows Boot Manager

Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)

    Windows 7
    Windows Vista
    Windows XP


To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10


Tools:

    Windows Memory Diagnostic


ENTER=Choose                    TAB=Menu                    ESC=Cancel
```

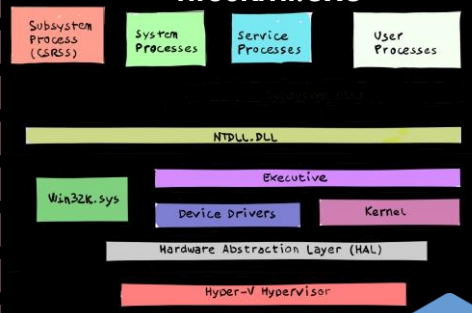


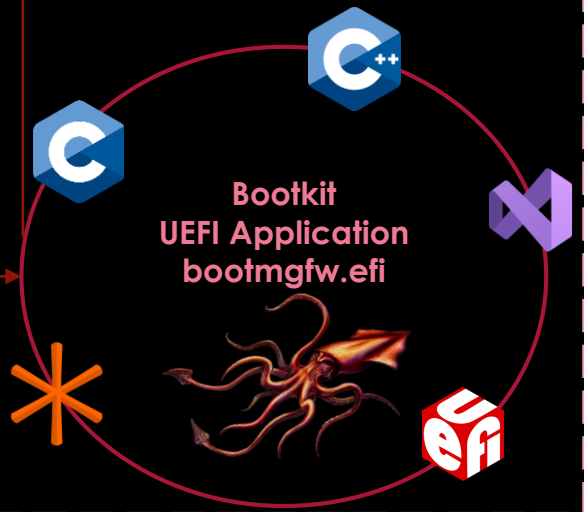
Windows OS Loader
%SystemRoot%\system32\
winload.efi

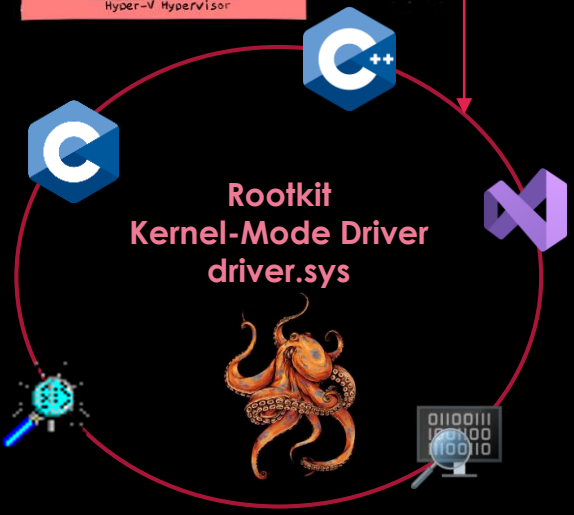
```
Advanced Boot Options

Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)

  Repair Your Computer

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally

Description: View a list of system recovery tools you can use to repair
             startup problems, run diagnostics, or restore your system.

ENTER=Choose                                                ESC=Cancel
```



Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe







DESARROLLO

- MalwareUnderRadarPkg
  - MalwareUnderRadarPkg.dec
  - MalwareUnderRadarPkg.dsc
  - README.md
  - Application
    - MalwareUnderRadarApplicationUEFI.c
    - MalwareUnderRadarApplicationUEFI.inf
    - Functions
      - 1FunctionsNetwork.c
      - 1FunctionsNetwork.h
      - 2FunctionsLayerApplication.c
      - 2FunctionsLayerApplication.h
      - Utils
        - 1FunctionsUtilsFiles.c
        - 1FunctionsUtilsFiles.h

```
Status = gBS->OpenProtocol(TcpHandle, &gEfiTcp4ProtocolGuid, (VOID **)&TcpListener,
                           gImageHandle, NULL, EFI_OPEN_PROTOCOL_GET_PROTOCOL);

if (Status == EFI_NO_MAPPING)
{
    do
    {
        Status = TcpListener->GetModeData(TcpListener, NULL, NULL,
                                          &Ip4ModeData, NULL, NULL);
    } while (!Ip4ModeData.IsConfigured);

    Status = TcpListener->Configure(TcpListener, &TcpConfigData);
}
```

```
Status = gBS->CreateEvent(EVT_NOTIFY_SIGNAL, TPL_CALLBACK, RequestCallback, NULL,
                          &RequestToken.Event);

RequestToken.Status = EFI_SUCCESS;
RequestToken.Message = &RequestMessage;
gRequestCallbackComplete = FALSE;

Status = HttpProtocol->Request(HttpProtocol, &RequestToken);
```

GLUPTEBA

Glupteba Overview

Glupteba is built to be modular, which allows it to download and execute additional components or payloads. This modular design makes Glupteba adaptable to different attack scenarios and environments, and it also allows its operators to adapt to different security solutions.

Over the years, malware authors have introduced new modules, allowing the threat to perform a variety of tasks including the following:

- Delivering additional payloads
- Stealing credentials from various software
- Stealing sensitive information, including credit card data
- Enrolling the infected system in a cryptomining botnet
- Crypto hijacking and delivering miners
- Performing digital advertising fraud
- Stealing Google account information
- Bypassing UAC and having both rootkit and bootkit components
- Exploiting routers to gain credentials and remote administrative access

Diving Into Glupteba's UEFI Bootkit

By Lior Rochberger and Dan Yashnik
February 12, 2024 at 6:00 AM
Category: Malware
Tags: Advanced Threat Prevention, Advanced URL Filtering, Advanced WildFire, Cloud-Delivered Security Services, coin miner, Cortex XDR, credential stealer, DNS security, next-generation firewall, Prisma Cloud, RedLine infostealer, Smoke Loader

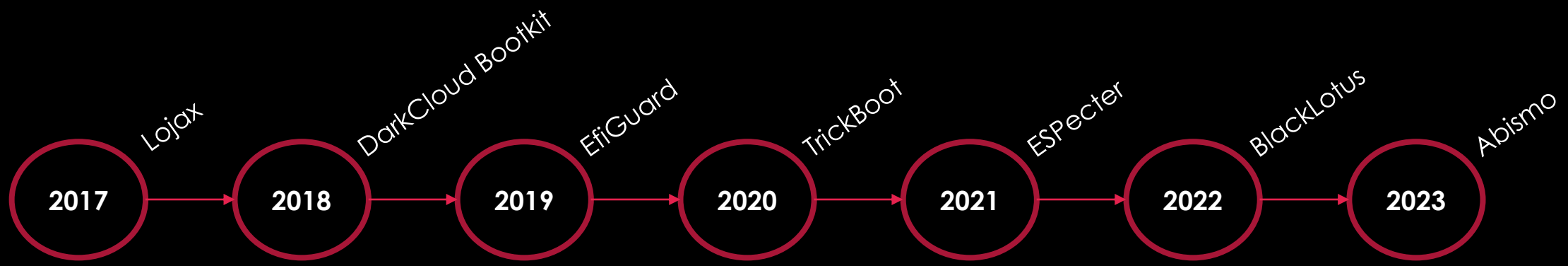
Executive Summary

Glupteba is advanced, modular and multipurpose malware that, for over a decade, has mostly been seen in financially driven cybercrime operations. This article describes the infection chain of a new campaign that took place around November 2023.

Despite being active for over a decade, certain capabilities that Glupteba's authors have added have remained undiscovered or unreported – until now. We will focus on one intriguing and previously undocumented feature: a Unified Extensible Firmware Interface (UEFI) bootkit. This bootkit can intervene and control the OS boot process, enabling Glupteba to hide itself and create a stealthy persistence that can be extremely difficult to detect and remove.

GLUPTEBA

An ESP malware implant executes code before Windows boots, undermining security features. An SPI flash memory implant offers more control, executing earlier in the boot process but requires higher privileges, increasing complexity.
~ Malware Developers



GLUPTEBA

Uncovering Glupteba's Bootkit Installer

We start our analysis with a bootkit installer binary disguised as a legitimate Windows binary (`csrss.exe`). When analyzing this installer, a clear lack of strings and functions indicates the file is packed in some way. This means we have some work to do before we can analyze the actual logic of the installer.

EfiGuard

EfiGuard is an open-source and portable UEFI bootkit that patches the Windows kernel by executing a DXE Runtime driver (`EfiGuardDxe.efi`) to disable PatchGuard and driver signature enforcement (DSE).

As documented in the GitHub project, `EfiGuardDxe.efi` can be executed either by installing it in a UEFI driver entry or booting a custom loader (`Loader.efi`) that loads the driver and then continues to load Windows. Glupteba uses the latter method.

1. The main_mountEFI function mounts the ESP into the B: drive
2. B:\EFI\Microsoft\Boot\bootmgfw.efi is renamed to B:\EFI\Microsoft\Boot\fw.efi
3. B:\EFI\Boot\bootx64.efi is renamed to B:\EFI\Boot\old.efi
4. The asset embedded\bootmgfw.efi is written to B:\EFI\Microsoft\Boot\bootmgfw.efi and to B:\EFI\Boot\bootx64.efi
5. The asset embedded \EfiGuardDxe.efi is written to B:\EFI\Boot\EfiGuardDxe.efi

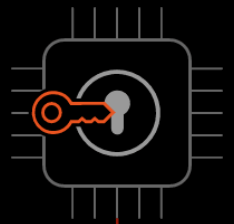
BOOTKIT UEFI

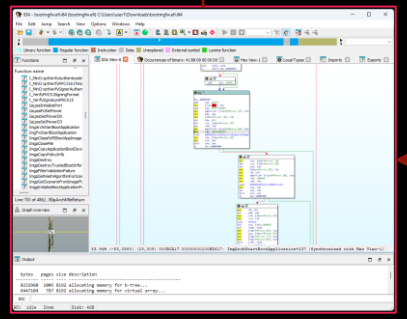
UEFI Application	DXE Runtime Driver	Kernel-Mode Driver	Client
UEFI Specification	DXE	Hide processes	IOCTLs
Protocols	Hooking Functions	Block network connections	Input/Output Request Packets
Download Malware	Runtime Services	Keylogger	Communication
User-Mode Components	**Persistence Tools**	**C&C Interface**	**Additional Malware**
HTTP Downloader	Services	Data Encoding	Trojan
Cryptomining	Registry	Encrypted Channel	Spyware
Stealer	Com Hijacking	Protocol Tunneling	Ransomware

UEFI Specification

2.10

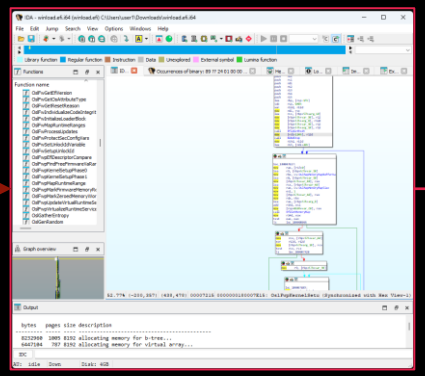
Search docs





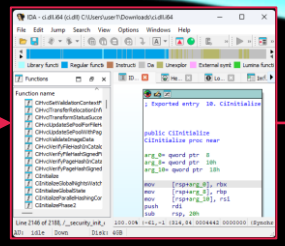
Windows Boot Manager
\EFI\Microsoft\Boot\
bootmgfw.efi

```
Windows Boot Manager

Choose an operating system to start, or press TAB to select a tool:
(Use the arrow keys to highlight your choice, then press ENTER.)

    Windows 7
    Windows Vista
    Windows XP


To specify an advanced option for this choice, press F8.
Seconds until the highlighted choice will be started automatically: 10


Tools:

    Windows Memory Diagnostic


ENTER=Choose                    TAB=Menu                    ESC=Cancel
```

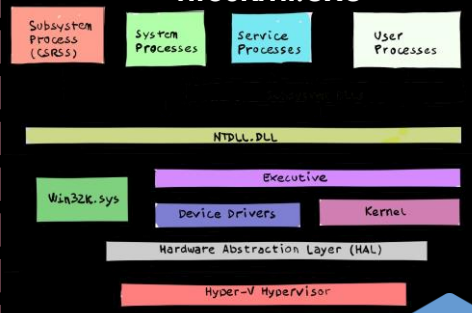


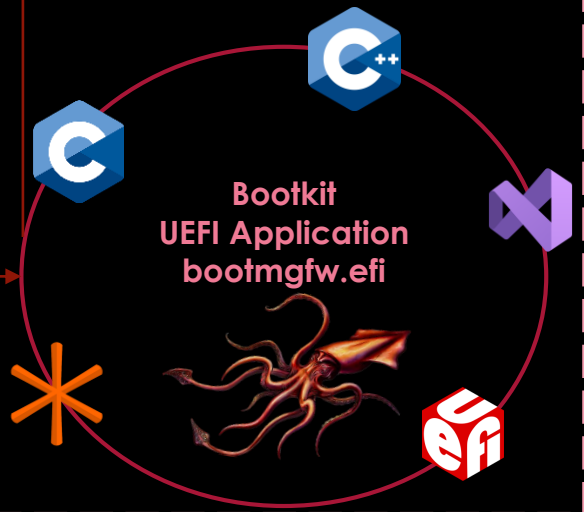
Windows OS Loader
%SystemRoot%\system32\
winload.efi

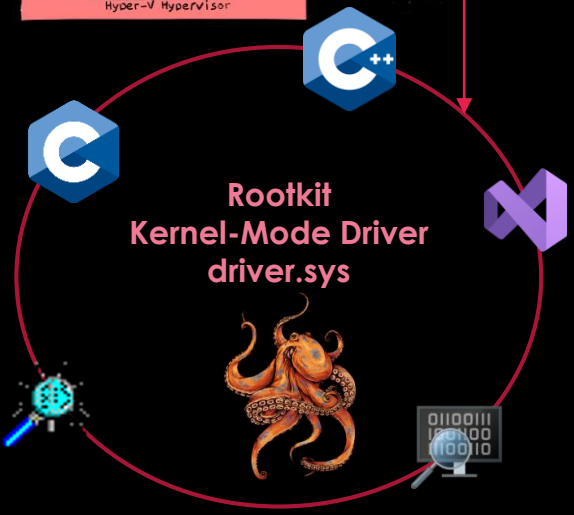
```
Advanced Boot Options

Choose Advanced Options for: Windows 7
(Use the arrow keys to highlight your choice.)

  Repair Your Computer

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally

Description: View a list of system recovery tools you can use to repair
             startup problems, run diagnostics, or restore your system.

ENTER=Choose                                                ESC=Cancel
```



Windows NT OS Kernel
%SystemRoot%\system32\
ntoskrnl.exe







FINFISHER

- Spyware
- UEFI Bootkit
- Windows, Linux and MacOS
- Obfuscation
- C&C

UEFI infection

During our research, we found a UEFI bootkit that was loading FinSpy. All machines infected with the UEFI bootkit had the Windows Boot Manager (**bootmgfw.efi**) replaced with a malicious one. When the UEFI transfers execution to the malicious loader, it first locates the original Windows Boot Manager. It is stored inside the **efi\microsoft\boot\en-us** directory, with the name consisting of hexadecimal characters. This directory contains two more files: the Winlogon Injector and the Trojan Loader. Both of them are encrypted with RC4. The decryption key is the EFI system partition GUID, which differs from one machine to another.

P:\EFI\Microsoft\Boot\en-US

n	Name	Size	
..		Up	
050ad6a5		468480	Encrypted Backdoor Loader
4182b569		1492 K	Original Windows Boot Manager
82056bd2		6236	Encrypted Winlogon Injector
bootmgfw.efi.mui		77112	Clean files
bootmgr.efi.mui		77112	
memtest.efi.mui		44856	

Sample contents of the \efi\microsoft\boot\en-us\ directory

Once the original bootloader is located, it is loaded into memory, patched and launched. The patched launcher:

- Patches the function of the OS loader that transfers execution to the kernel
- The patched function hooks the kernel's **PsCreateSystemThread** function, which, when called for the first time, creates an additional thread that decrypts the next loader stage and launches it.

ESPECTER

ESET RESEARCH

UEFI threats moving to the ESP: Introducing ESPecter bootkit

ESET research discovers a previously undocumented UEFI bootkit with roots going back all the way to at least 2012

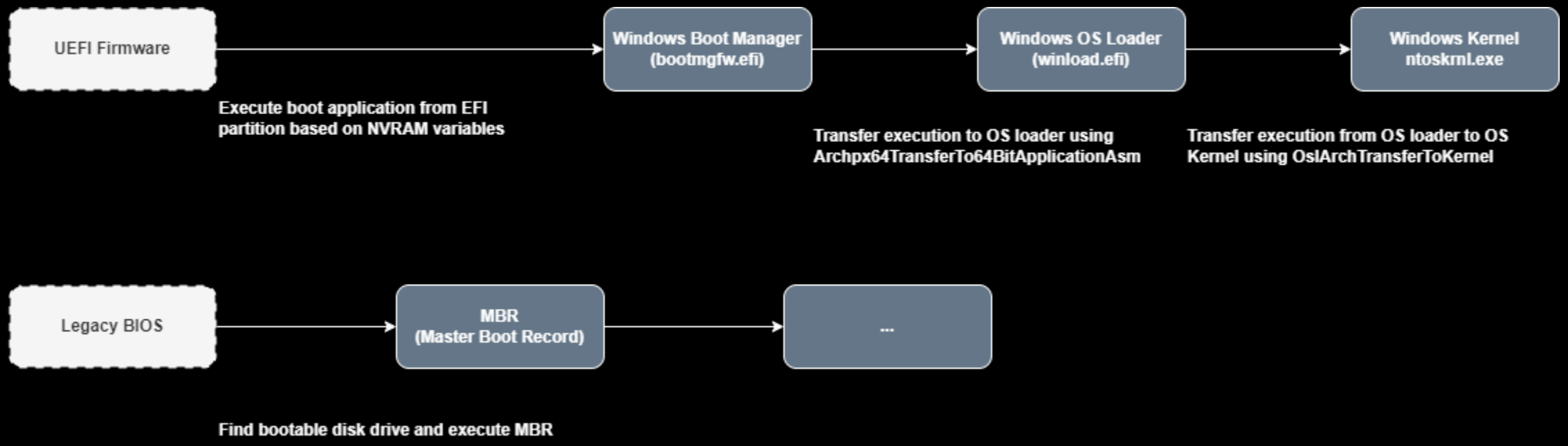
Martin Smolár Anton Cherepanov

ESET researchers have analyzed a previously undocumented, real-world UEFI bootkit that persists on the EFI System Partition (ESP). The bootkit, which we've named ESPecter, can bypass Windows Driver Signature Enforcement to load its own unsigned driver, which facilitates its espionage activities. Alongside Kaspersky's recent discovery of the unrelated FinSpy bootkit, it is now safe to say that real-world UEFI threats are no longer limited to SPI flash implants, as used by Lojax.

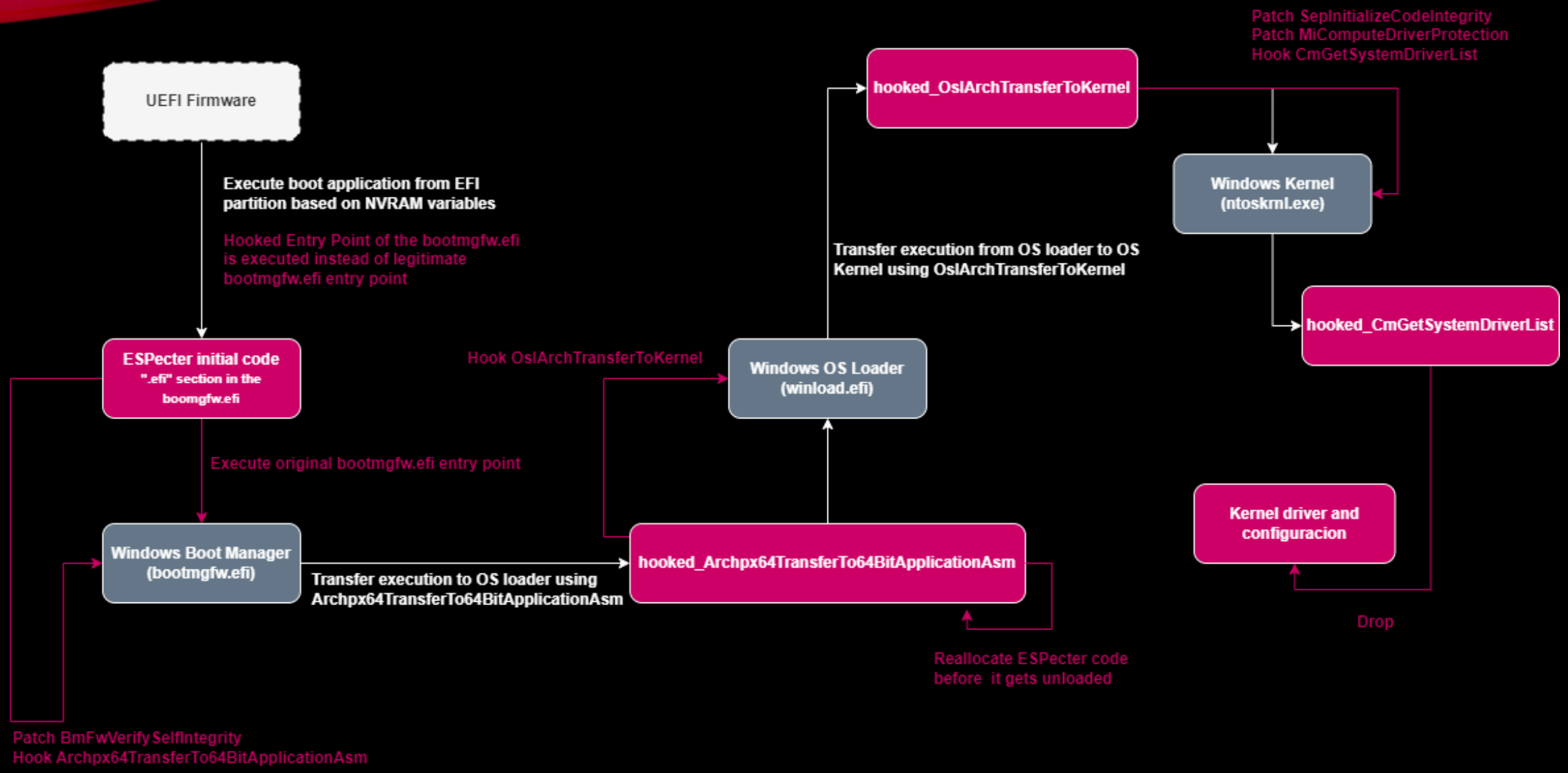
The days of UEFI (Unified Extensible Firmware Interface) living in the shadows of the legacy BIOS are gone for good. As a leading technology embedded into chips of modern computers and devices, it plays a crucial role in securing the pre-OS environment and loading the operating system. And it's no surprise that such a widespread technology has also become a tempting target for threat actors in their search for ultimate persistence.

- Spyware
- UEFI Bootkit
- Windows
- DSE
- Mode-Kernel Driver

ESPECTER


UEFI Firmware
Windows Boot Manager (bootmgfw.efi)
Windows OS Loader (winload.efi)
Windows Kernel ntoskrnl.exe
Execute boot application from EFI partition based on NVRAM variables
Transfer execution to OS loader using Archpx64TransferTo64BitApplicationAsm
Transfer execution from OS loader to OS Kernel using OslArchTransferToKernel
Legacy BIOS
MBR (Master Boot Record)
...
Find bootable disk drive and execute MBR

ESPECTER


UEFI Firmware
Execute boot application from EFI partition based on NVRAM variables
Hooked Entry Point of the bootmgfw.efi is executed instead of legitimate bootmgfw.efi entry point
ESPecter initial code
".efi" section in the boomgfw.efi
Execute original bootmgfw.efi entry point
Windows Boot Manager (bootmgfw.efi)
Transfer execution to OS loader using Archpx64TransferTo64BitApplicationAsm
hooked_Archpx64TransferTo64BitApplicationAsm
Hook OslArchTransferToKernel
Windows OS Loader (winload.efi)
Transfer execution from OS loader to OS Kernel using OslArchTransferToKernel
hooked_OslArchTransferToKernel
Windows Kernel (ntoskrnl.exe)
hooked_CmGetSystemDriverList
Patch SeplnitializeCodeIntegrity
Patch MiComputeDriverProtection
Hook CmGetSystemDriverList
Kernel driver and configuracion
Drop
Reallocate ESPecter code before it gets unloaded
Patch BmFwVerifySelfIntegrity
Hook Archpx64TransferTo64BitApplicationAsm

ESPECTER

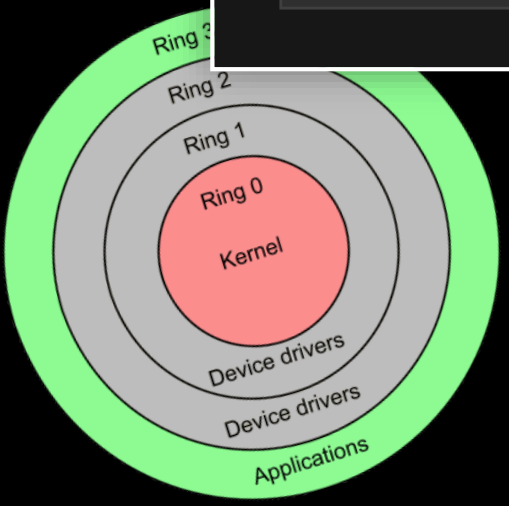
KEYBOARD_INPUT_DATA structure (ntddkbd.h)

KEYBOARD_INPUT_DATA contains one packet of keyboard input data.

Syntax

C++ Copy

```cpp
typedef struct _KEYBOARD_INPUT_DATA {
  USHORT UnitId;
  USHORT MakeCode;
  USHORT Flags;
  USHORT Reserved;
  ULONG  ExtraInformation;
} KEYBOARD_INPUT_DATA, *PKEYBOARD_INPUT_DATA;
```




```cpp
/**
    @brief      Function to read keystrokes from keyboard device.

    @param      pDeviceObject       Pointer to a DEVICE_OBJECT structure representing the device.
    @param      pIrp                Pointer to the I/O request packet (IRP) for reading keystrokes.

    @return     A NTSTATUS value indicating success or an error code if operation fails.
**/


NTSTATUS
DriverReadKeystrokes(
    _In_    PDEVICE_OBJECT      pDeviceObject,
    _In_    PIRP                pIrp
)
{
    IoCopyCurrentIrpStackLocationToNext(pIrp);

    IoSetCompletionRoutine(pIrp, ReadOperationFinished, NULL, TRUE, TRUE, TRUE);

    pendingKey++;

    return IoCallDriver(((PDEVICE_EXTENSION)pDeviceObject->DeviceExtension)->LowerKbdDevice, pIrp);
}
```

BLACKLOTUS

ESET RESEARCH

BlackLotus UEFI bootkit: Myth confirmed

The first in-the-wild UEFI bootkit bypassing UEFI Secure Boot on fully updated UEFI systems is now a reality

Martin Smolár

The number of UEFI vulnerabilities discovered in recent years and the failures in patching them or revoking vulnerable binaries within a reasonable time window hasn't gone unnoticed by threat actors. As a result, the first publicly known UEFI bootkit bypassing the essential platform security feature – UEFI Secure Boot – is now a reality. In this blogpost we present the first public analysis of this UEFI bootkit, which is capable of running on even fully-up-to-date Windows 11 systems with UEFI Secure Boot enabled. Functionality of the bootkit and its individual features leads us to believe that we are dealing with a bootkit known as **BlackLotus**, the UEFI bootkit being sold on hacking forums for $5,000 since at least October 2022.

UEFI bootkits are very powerful threats, having full control over the OS boot process and thus capable of disabling various OS security mechanisms and deploying their own kernel-mode or user-mode payloads in early OS startup stages. This allows them to operate very stealthily and with high privileges. So far, only a few have been discovered in the wild and publicly described (e.g., multiple malicious EFI samples we discovered in 2020, or fully featured UEFI bootkits such as our discovery last year – the ESPecter bootkit – or the FinSpy bootkit discovered by researchers from Kaspersky).

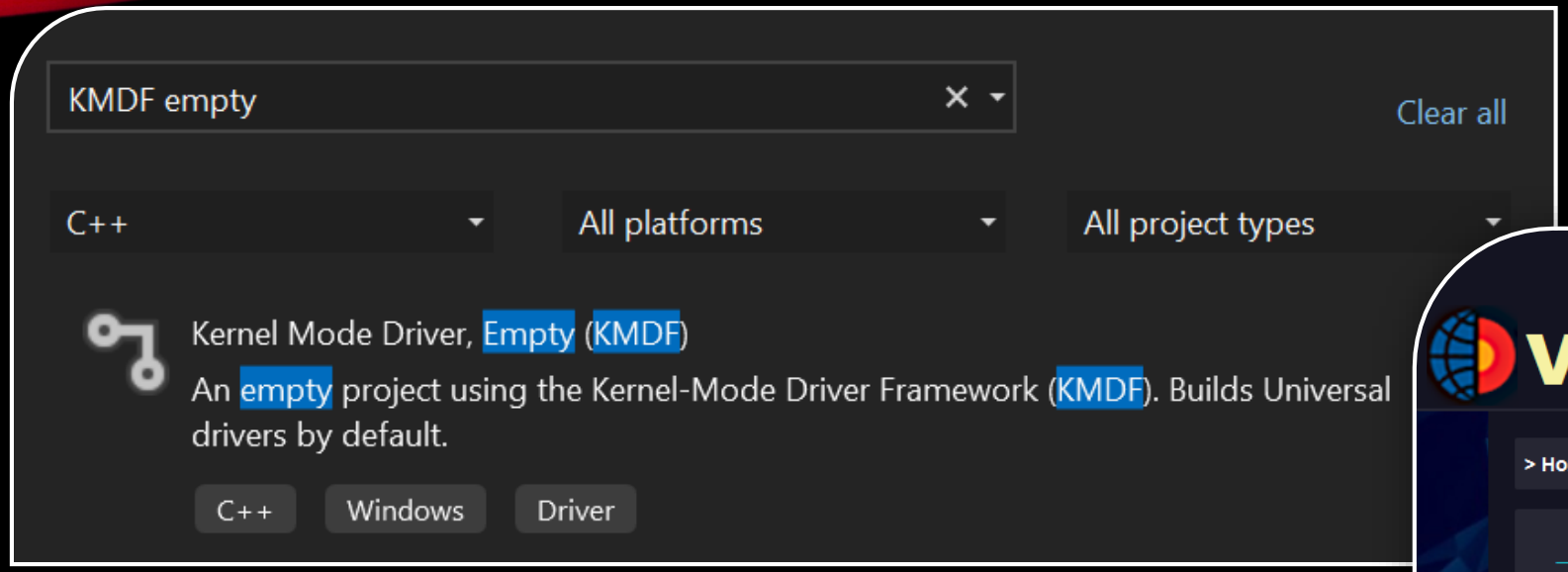
README Unlicense license

baton drop (CVE-2022-21894): Secure Boot Security Feature Bypass Vulnerability

Windows Boot Applications allow the `truncatememory` setting to remove blocks of memory containing "persistent" ranges of serialised data from the memory map, leading to Secure Boot bypass.

- The `truncatememory` BCD element will remove all memory above a specified physical address from the memory map.
- This is performed for each boot application during initialisation, before the serialised Secure Boot policy is read from memory.
- Therefore, such an element can be used to remove the serialised Secure Boot policy from the memory map.
- This will allow dangerous settings to be used in a boot application (`bootdebug`, `testsigning`, `nointegritychecks`), thus breaking Secure Boot.

BLACKLOTUS



The kernel driver is responsible for four main tasks:

1. Injecting the HTTP downloader into winlogon.exe and reinjecting it in case the thread terminated.
2. Protecting bootkit files deployed on the ESP from being removed.
3. Disarming the user-mode Windows Defender process MsMpEngine.exe.
4. Communicating with the HTTP downloader and if necessary, performing any commands.

REVERSING Y ANÁLISIS DE MALWARE

Campus Internacional de Ciberseguridad

Master's degree instructor (Reverse Engineering, Malware Analysis and Bug Hunting)

Currently, I am teacher in the prestigious 'Máster en Reversing, Análisis de Malware y Bug Hunting' at the Campus Internacional de Ciberseguridad.

These are some of the topics I cover:

- Windows Architecture (User Mode, Kernel Mode)
- Windows Protections (DSE, KPP)
- Malware Hunting (Sysinternals Tools)
- Windows Kernel Opaque Structures (EPROCESS, ETHREAD)
- Windows Kernel Debugging (WinDbg)
- WinDbg Scripting (Javascript, PyKd)
- Rootkit Hooking Techniques (IDT, SSDT)
- Rootkit Development (Kernel Mode Drivers, IRPs)
- Bootkit Development (UEFI Applications)
- Bootkit Analysis (ESPecter, BlackLotus)
- Kernel Exploitation (Vulnerable Drivers, Write-what-where)

Máster en Reversing, Análisis de Malware y Bug Hunting
Una de las técnicas por excelencia para analizar el comportamiento de las aplicaciones maliciosas cuando no se tiene el código fuente de la aplicación es el reversing. Los...

PREGUNTAS

VICON

- https://github.com/TheMalwareGuardian/ViconGal

Bootkit

- https://github.com/TheMalwareGuardian/Abismo

Rootkits

- https://github.com/TheMalwareGuardian/Bentico

Recursos

- https://github.com/TheMalwareGuardian/Awesome-Bootkits-Rootkits-Development