# Modern Bootkit Trends:

## Bypassing Kernel-Mode Signing Policy

**Aleksandr Matrosov**

**Eugene Rodionov**

vb 2011
BARCELONA
5-7 October 2011

ESET

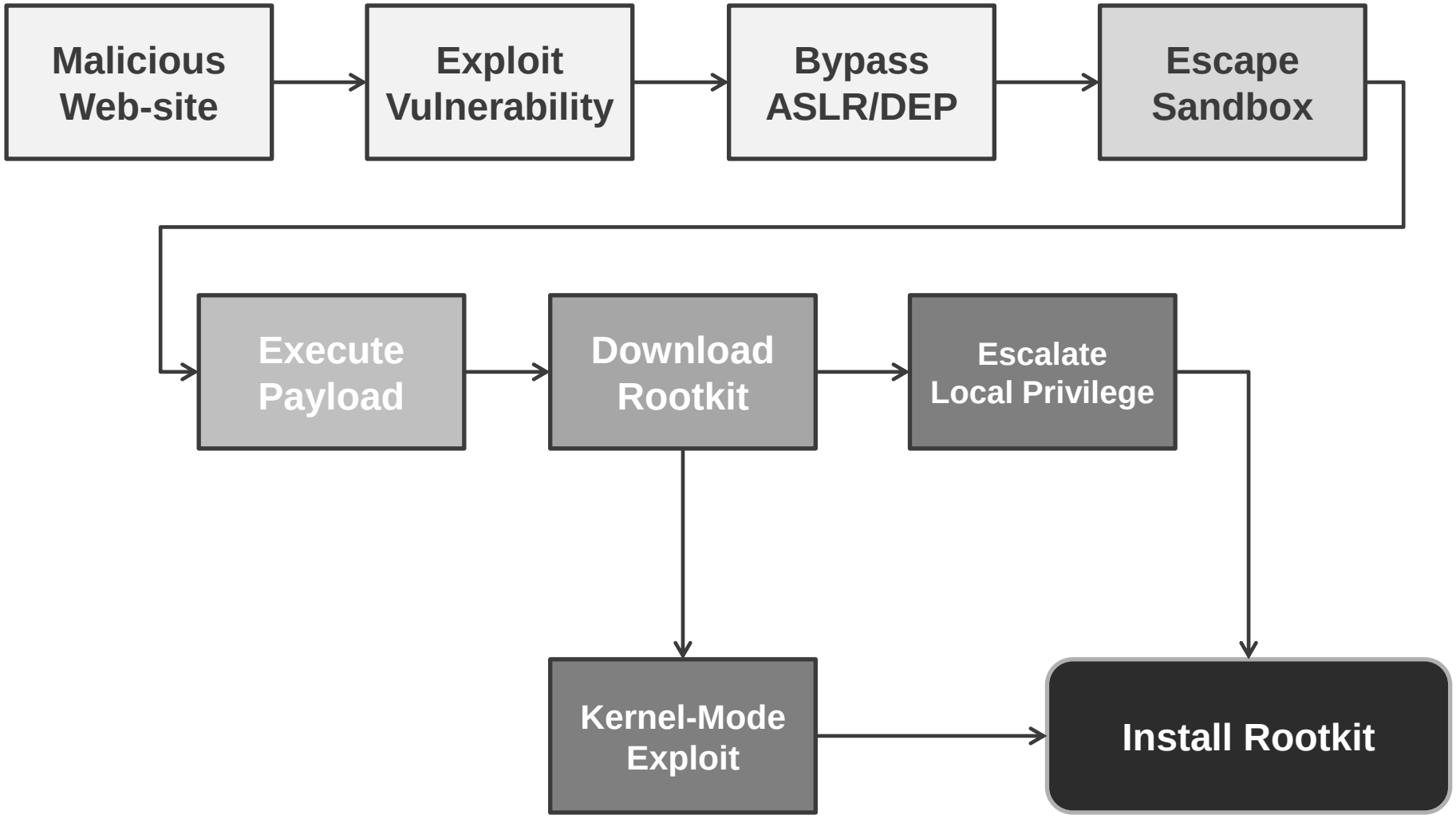
# Agenda

- ✓ Evolution of payloads and rootkits
- ✓ Bypassing code integrity checks
- ✓ Attacking Windows Bootloader
- ✓ Modern Bootkit details:
  - Win64/Olmarik
  - Win64/Rovnix

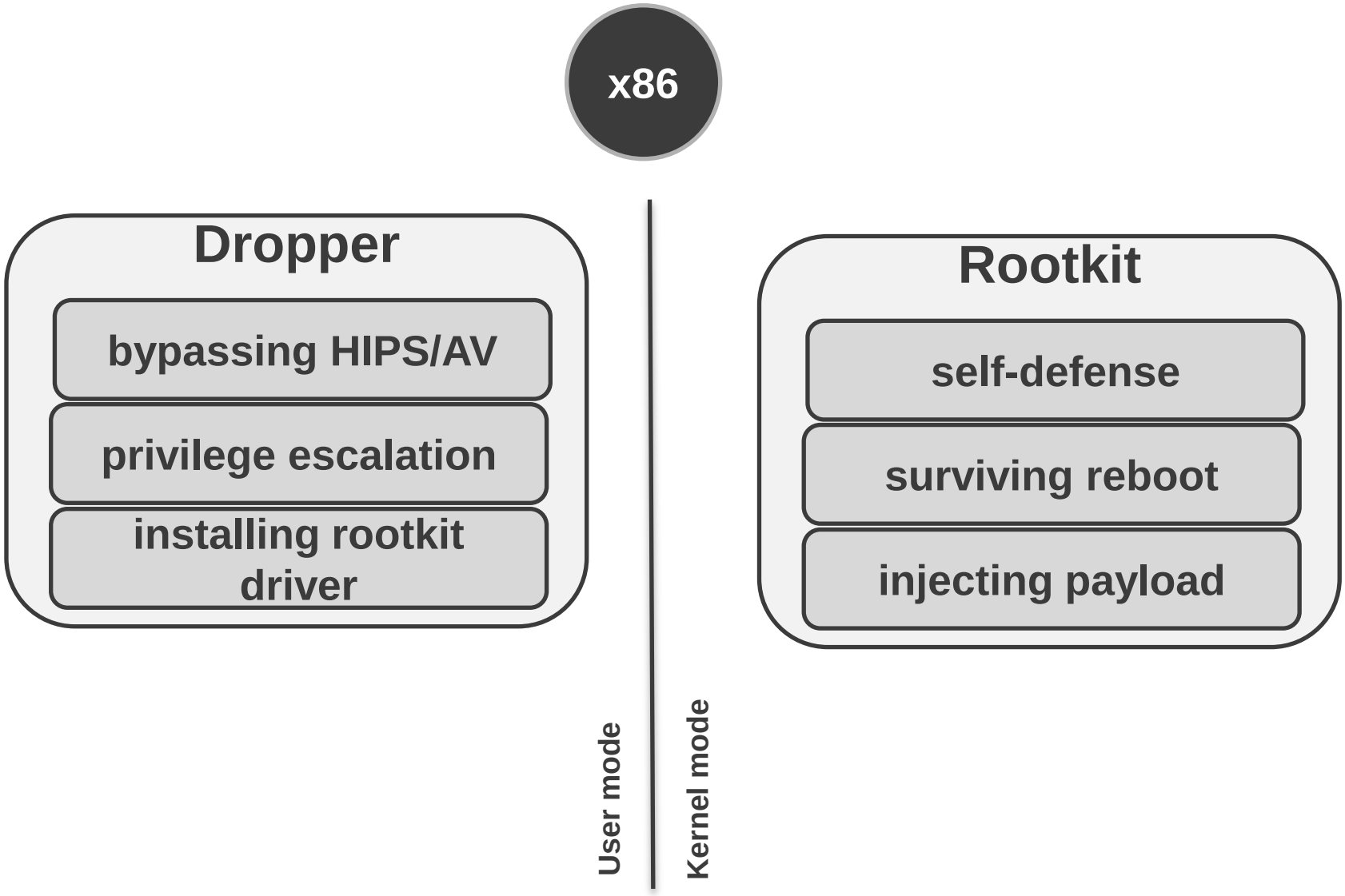
- ✓ What Facilitates Bootkit Attack Vector


# Evolution of Rootkits

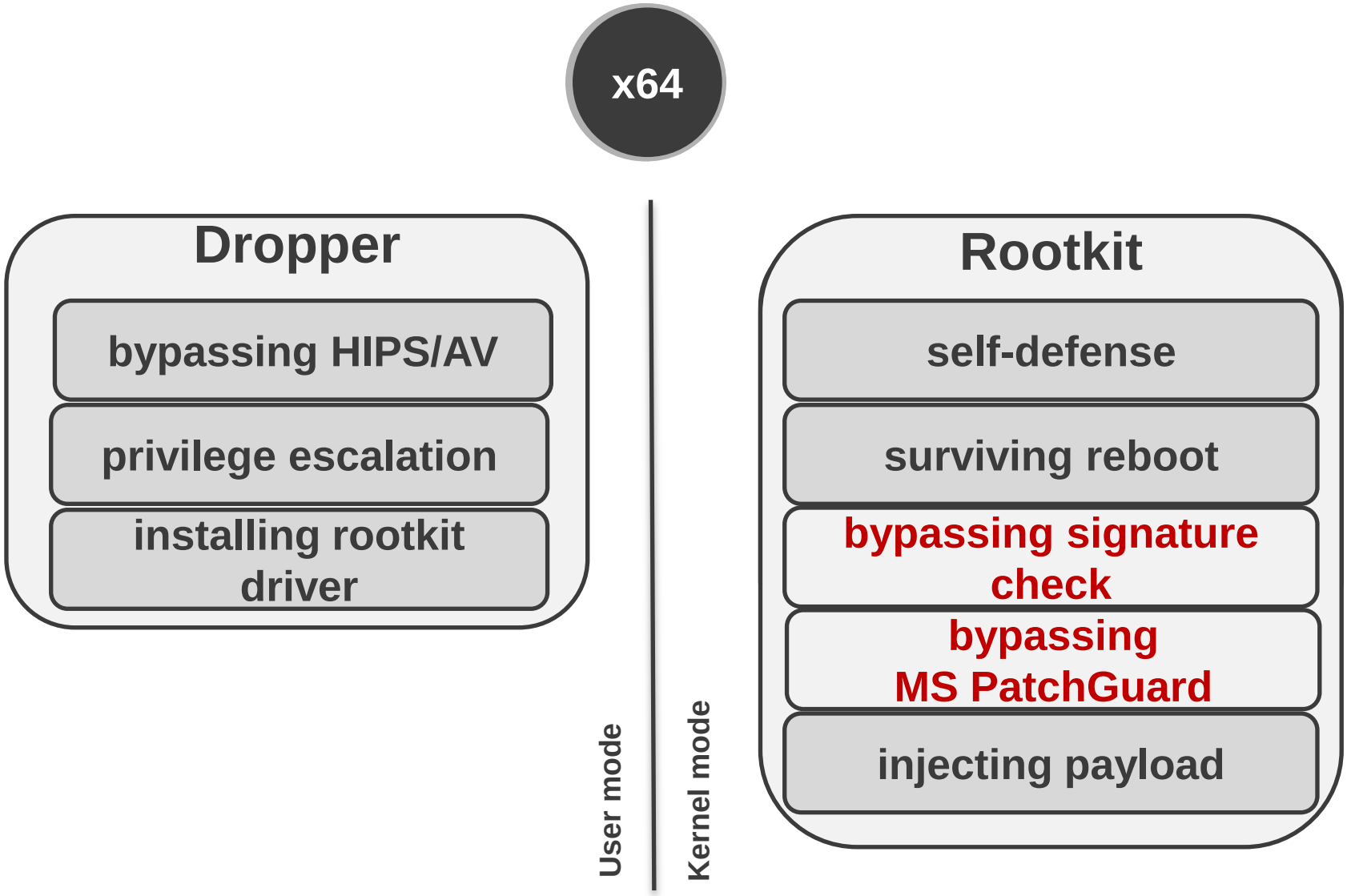
# Evolution of Rootkit Installation



# Evolution of Rootkit Features



# Evolution of Rootkit Features



# Obstacles for 64-bit Rootkits

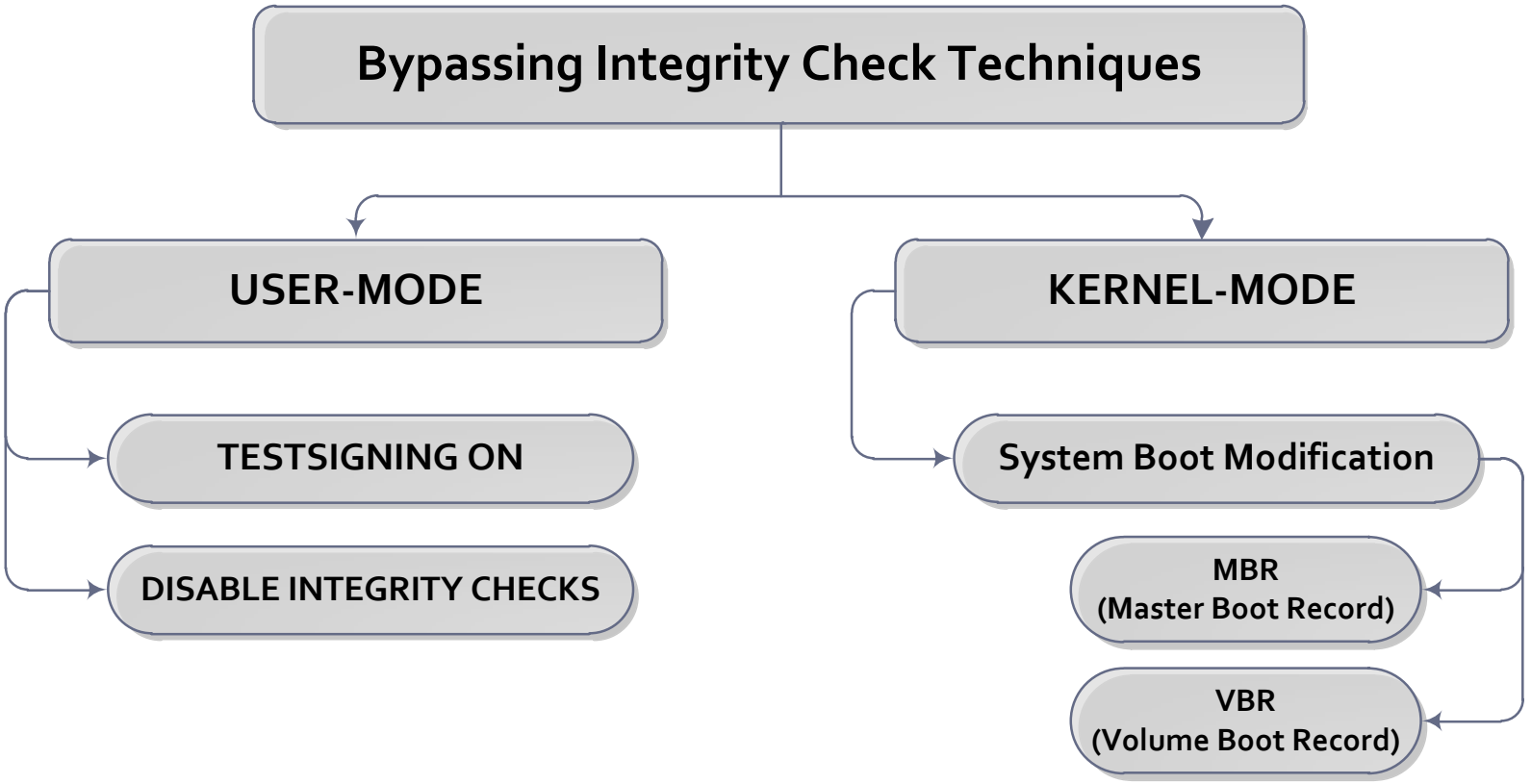
- **Kernel-Mode Code Signing Policy:**
  - ✓ **It is “difficult” to load unsigned kernel-mode driver**
- **Kernel-Mode Patch Protection (Patch Guard):**
  - ✓ **SSDT (System Service Dispatch Table)**
  - ✓ **IDT (Interrupt Descriptor Table)**
  - ✓ **GDT (Global Descriptor Table)**
  - ✓ **MSRs (Model Specific Registers)**


# Bypassing Code Integrity Checks

# Subverting KMCSP

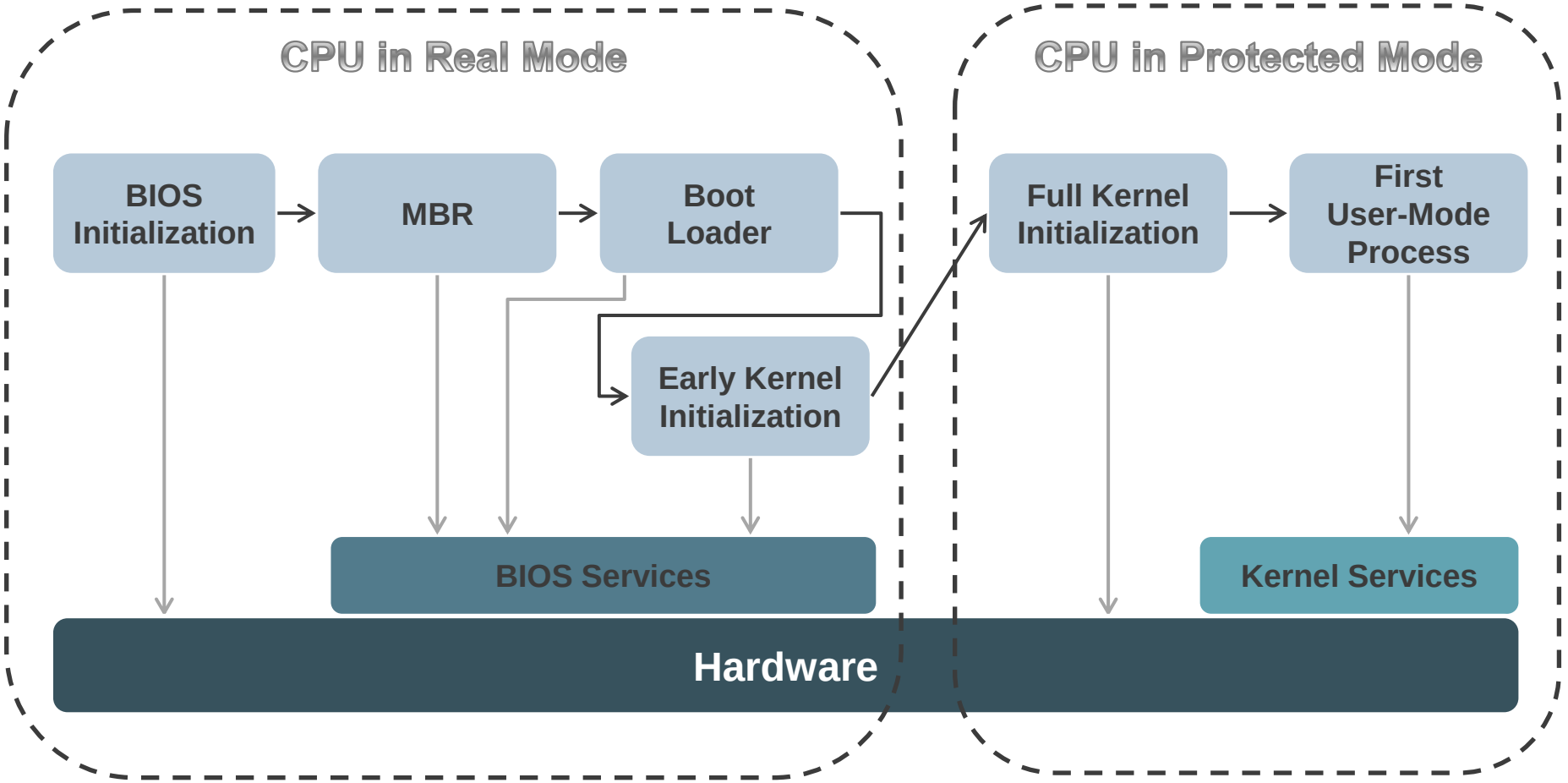
- Abusing vulnerable, signed, legitimate kernel-mode driver
- Switching off kernel-mode code signing checks by altering BCD data:
  - ✓abusing WinPE Mode
  - ✓disabling signing check
  - ✓enabling test signing
- Patching Bootmgr and OS loader

# Bypassing Integrity Checks

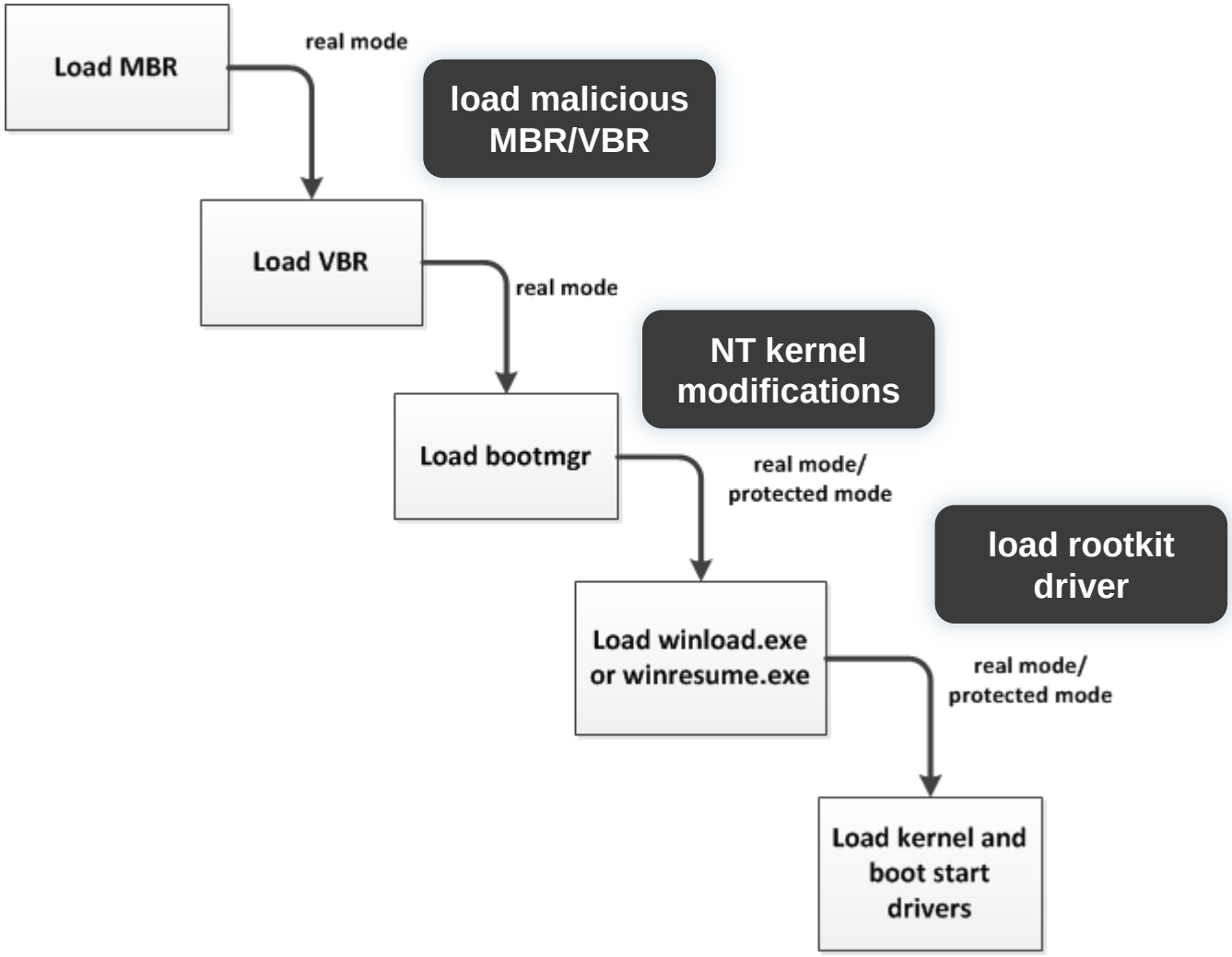



# Attacking Windows Bootloader

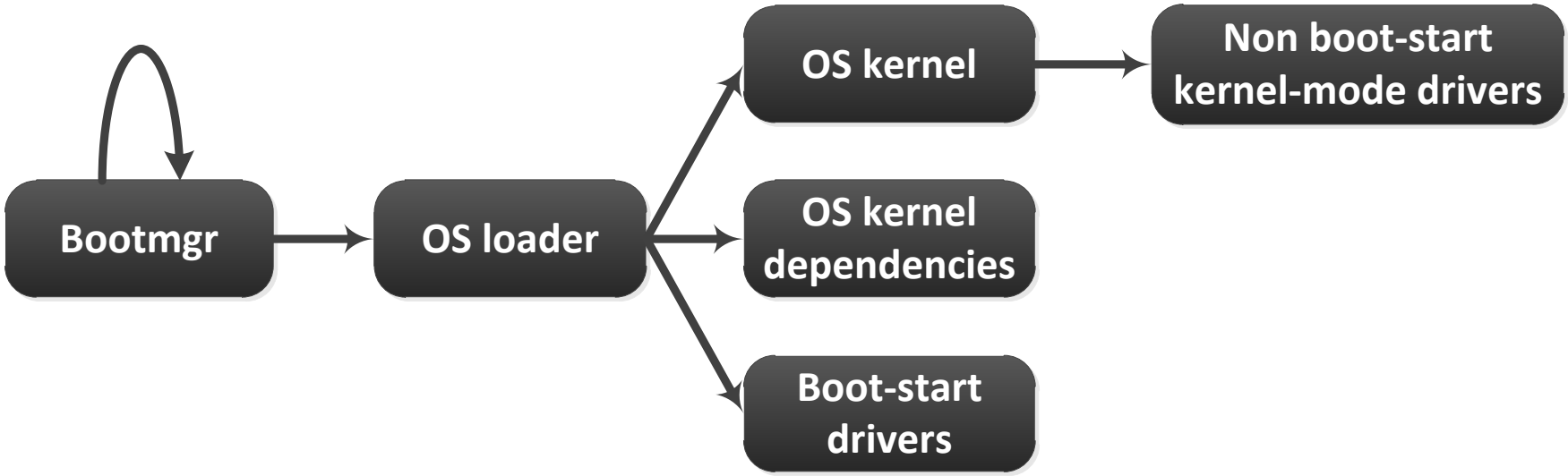
# Boot Process



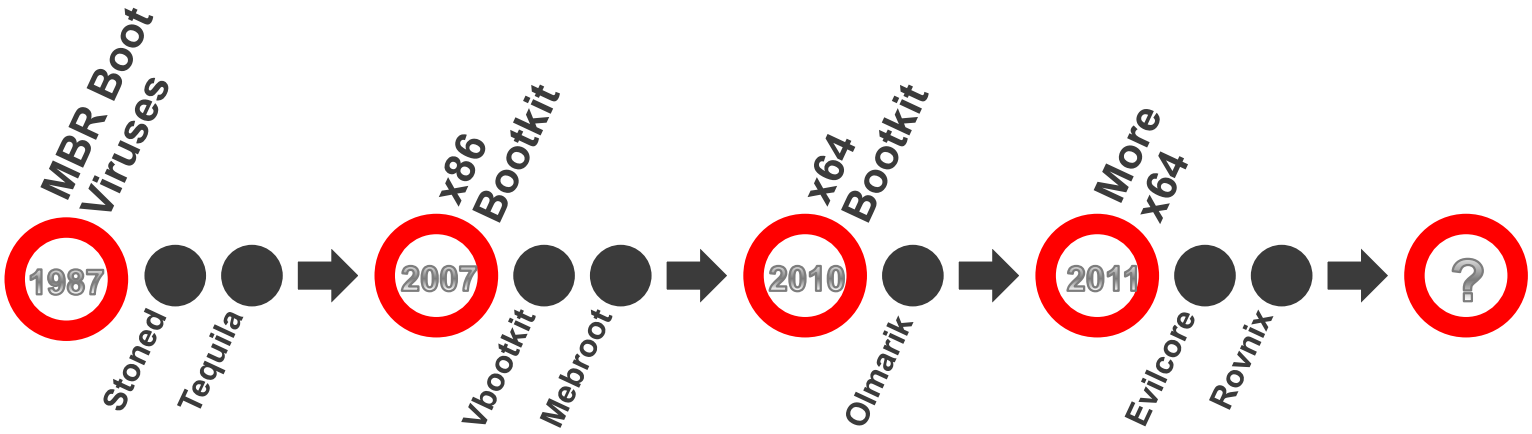
# Boot Process with Bootkit Infection



# Code Integrity Check



# Evolution of Bootkits

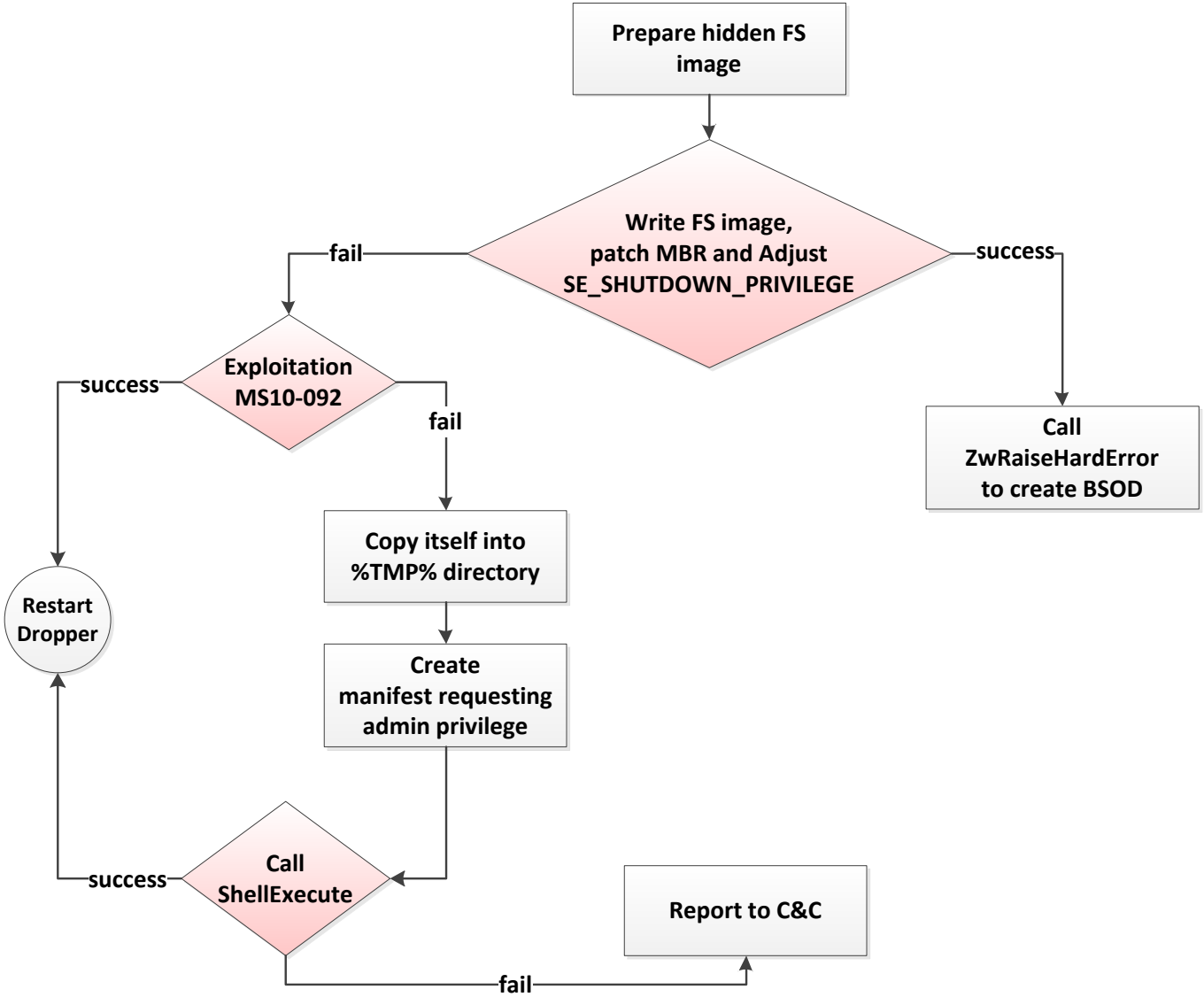



- **Bootkit PoC evolution:**
  - ✓ **eEye Bootroot (2005)**
  - ✓ **Vbootkit (2007)**
  - ✓ **Vbootkit v2 (2009)**
  - ✓ **Stoned Bootkit (2009)**
  - ✓ **Evilcore x64 (2011)**

- **Bootkit Threats evolution:**
  - ✓ **Win32/Mebroot (2007)**
  - ✓ **Win32/Mebratix (2008)**
  - ✓ **Win32/Mebroot v2 (2009)**
  - ✓ **Win64/Olmarik (2010/11)**
  - ✓ **Win64/Rovnix (2011)**

# Win64/Olmarik

# TDL4 Installation on x64



# BCD Elements determining KMCSP
## (before KB2506014)

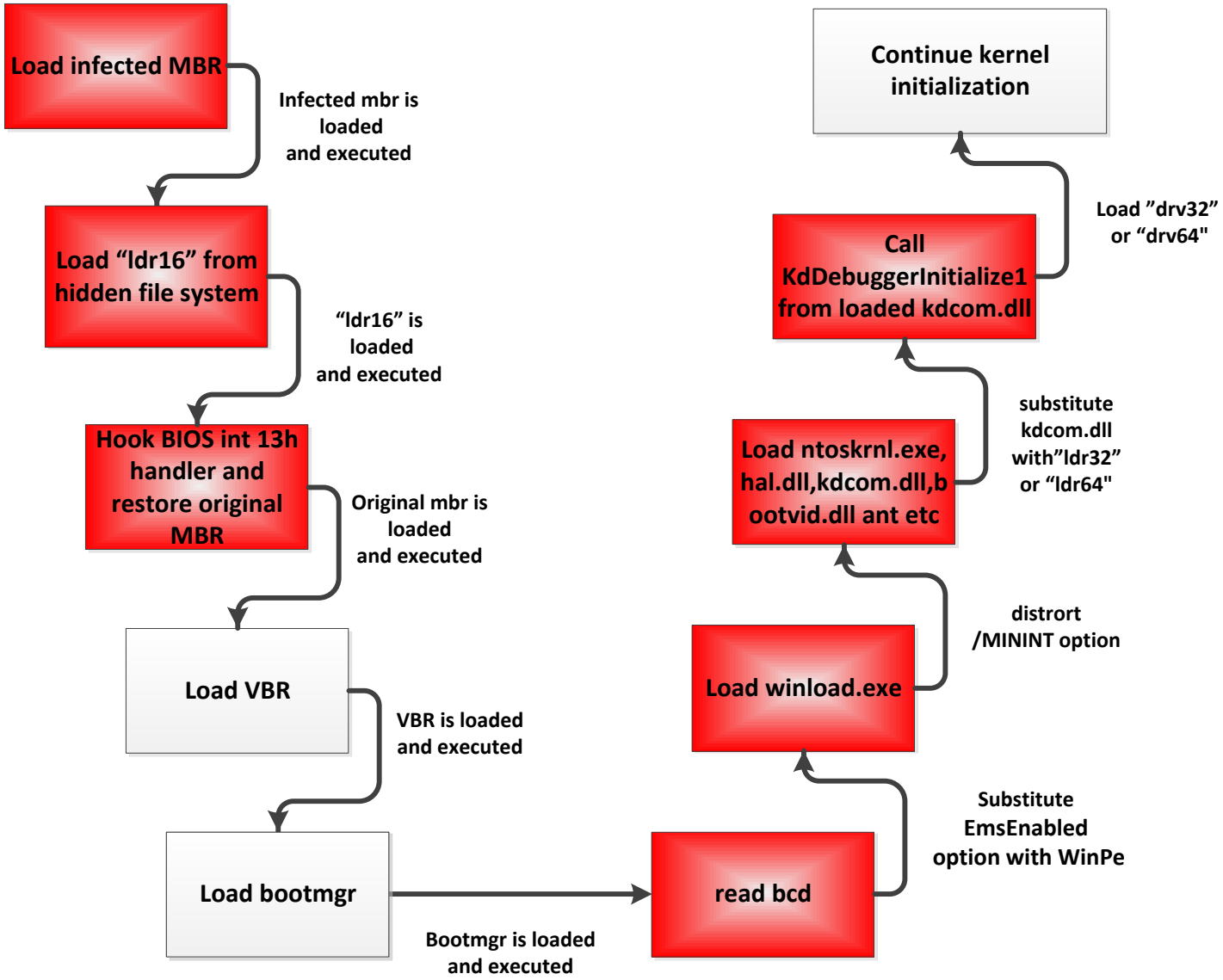
| BCD option | Description |
| --- | --- |
| BcdLibraryBoolean_DisableIntegrityCheck (0x16000020) | disables kernel-mode code integrity checks |
| BcdOSLoaderBoolean_WinPEMode (0x26000022) | instructs kernel to be loaded in preinstallation mode, disabling kernel-mode code integrity checks as a byproduct |
| BcdLibraryBoolean_AllowPrereleaseSignatures (0x16000049) | enables test signing |

# Abusing Win PE mode: TDL4 modules

| Module name | Description |
|---|---|
| **mbr (infected)** | infected MBR loads *ldr16* module and restores original MBR in memory |
| **ldr16** | hooks 13h interrupt to disable KMCSP and substitute *kdcom.dll* with *ldr32* or *ldr64* |
| **ldr32** | reads TDL4's kernel-mode driver from hidden file system and maps it into kernel-mode address space |
| **ldr64** | implementation of *ldr32* module functionality for 64-bit OS |

**int 13h – service provided by BIOS to communicate with IDE HDD controller**

# Abusing Win PE mode: Workflow

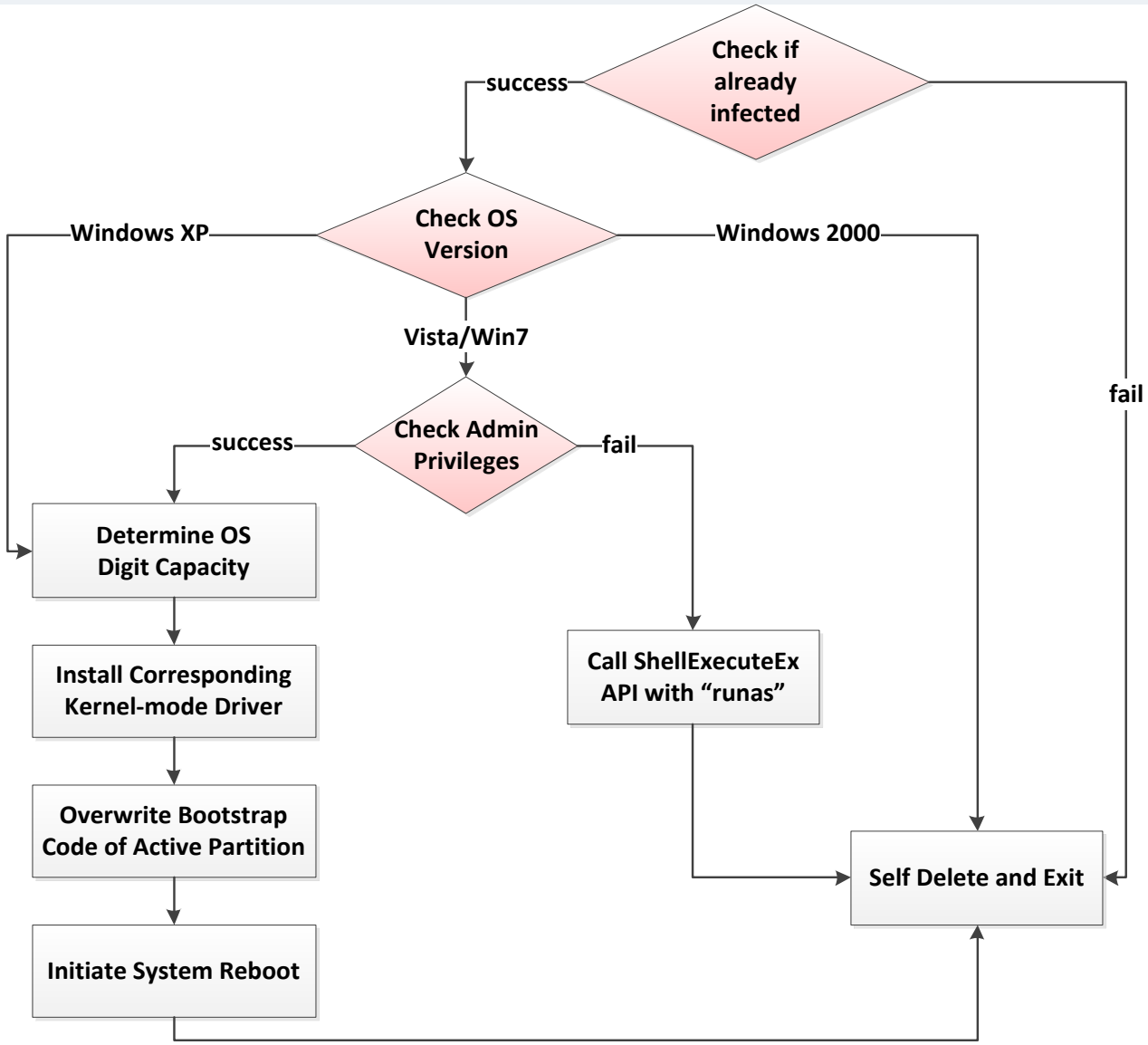


# MS Patch (KB2506014)

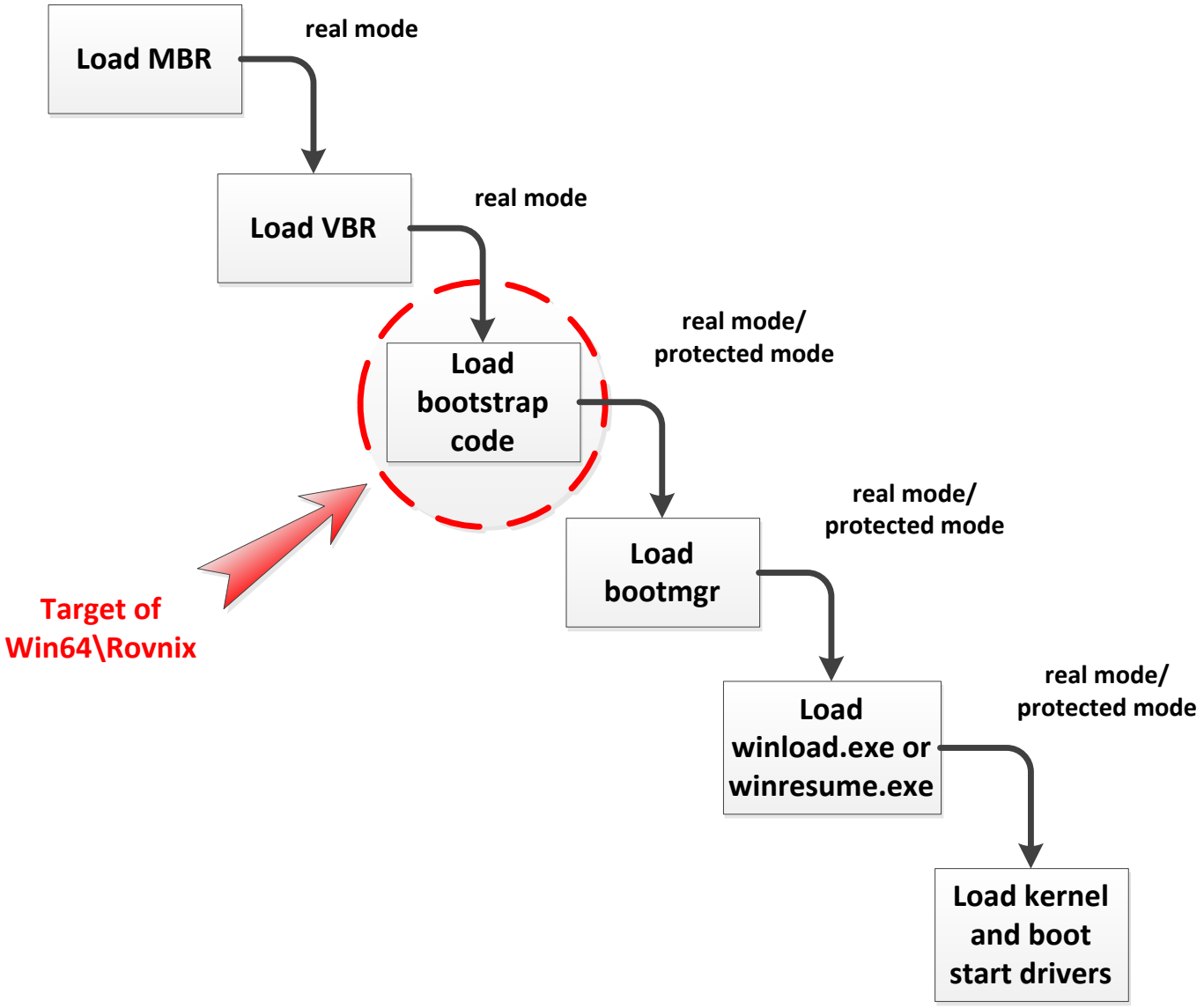
- ***BcdOsLoaderBoolean_WinPEMode*** **option no longer influences kernel-mode code signing policy**
- **Size of the export directory of *kdcom.dll* has been changed**

# Win64/Rovnix

# Win64/Rovnix: Installation

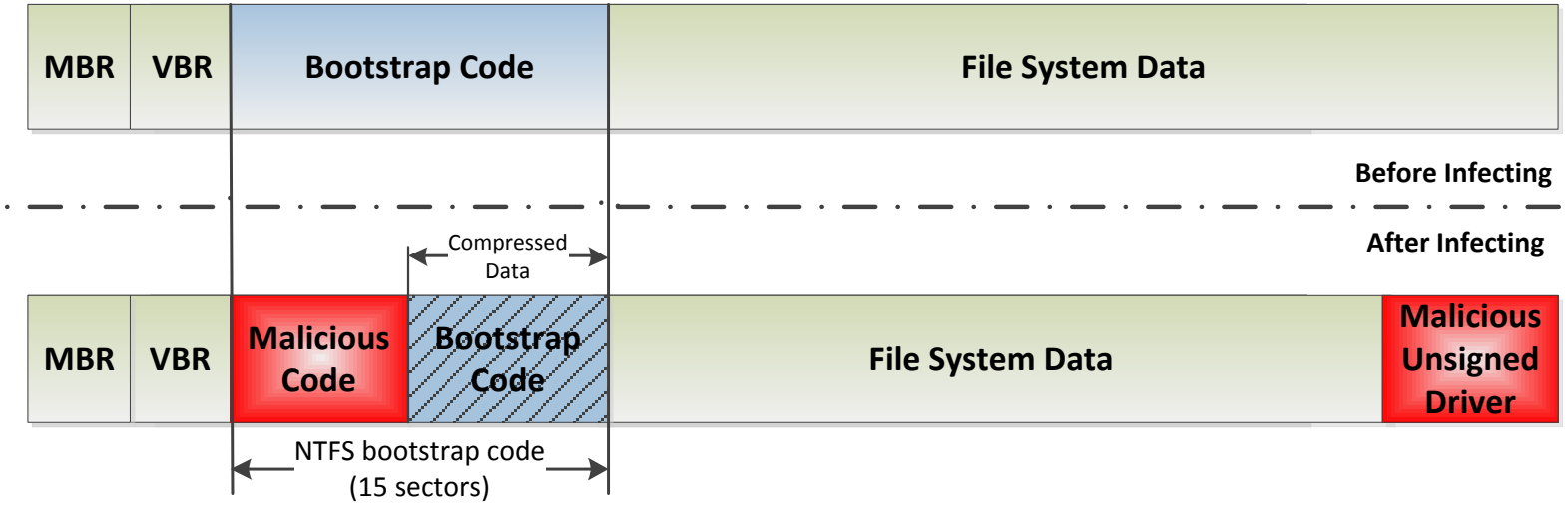


# Win64/Rovnix: Bootkit Overview

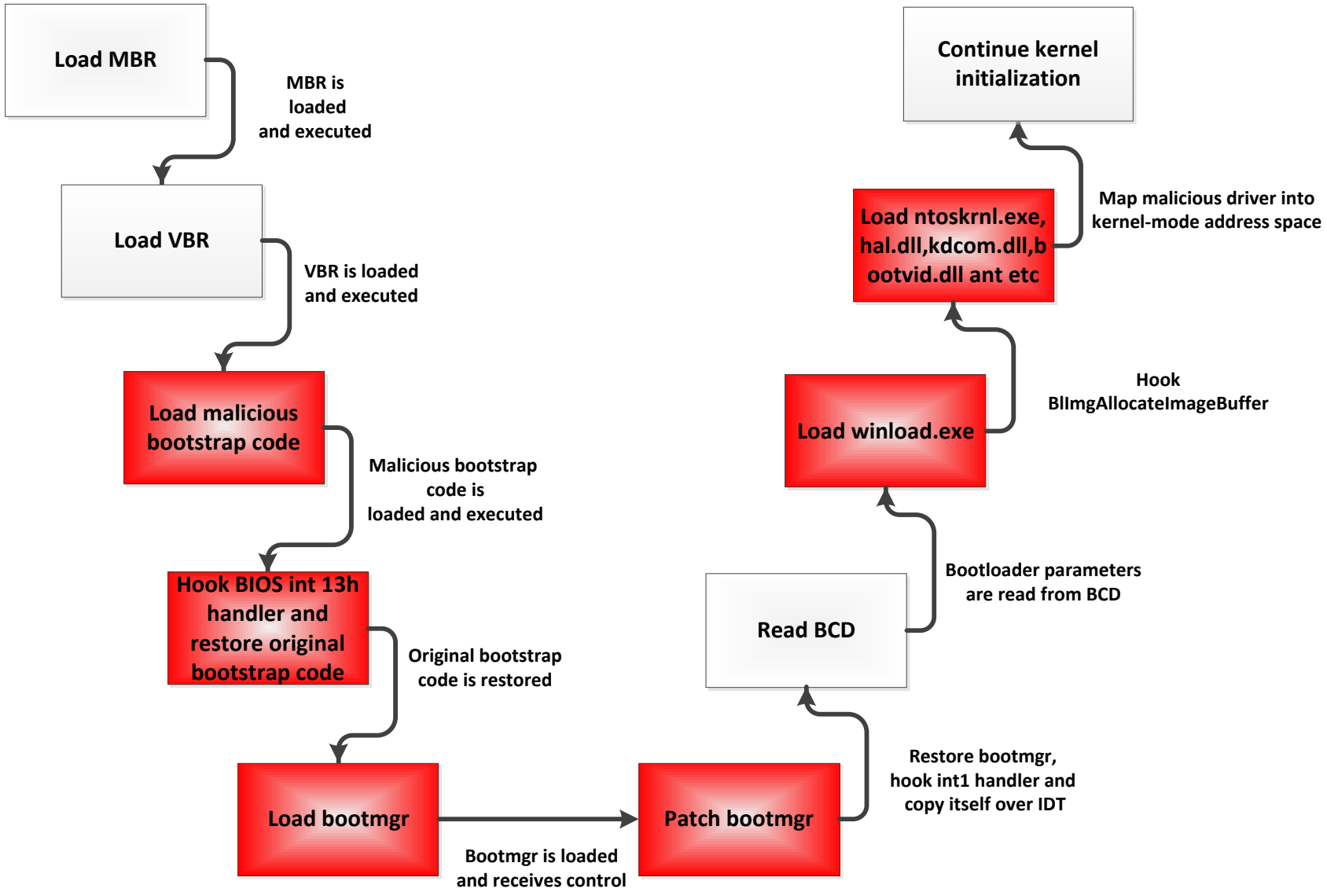


# Win64/Rovnix: Infected Partition Layout

- Win64/Rovnix overwrites bootstrap code of the active partition
- The malicious driver is written either:
  - ✓ before active partition, in case there is enough space
  - ✓ to the end of the hard drive, otherwise

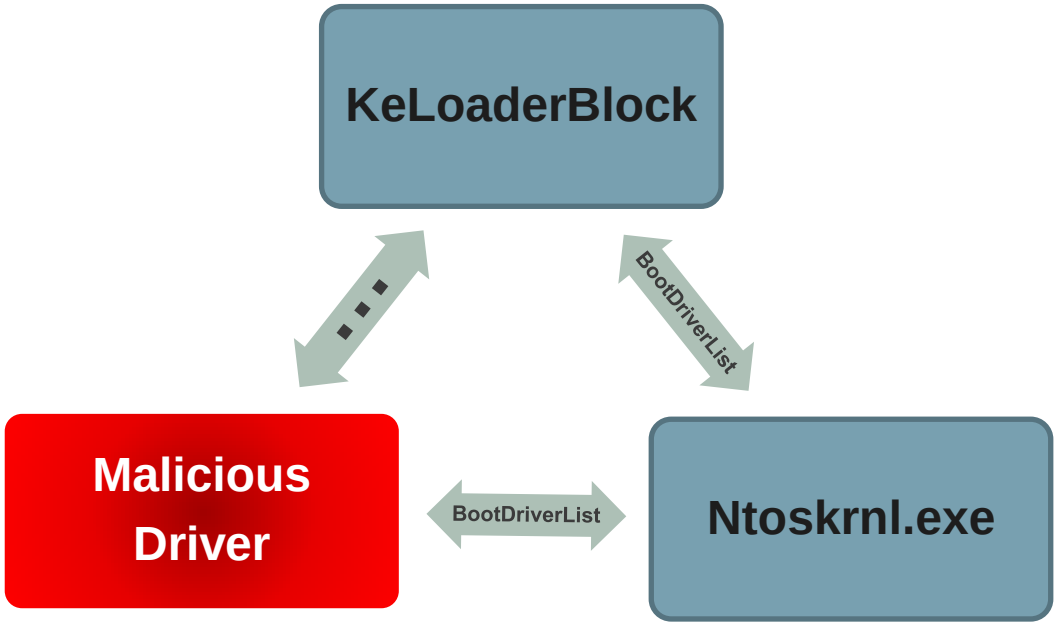


# Win64/Rovnix: Bootkit Details



# Win64/Rovnix: Loading Unsigned Driver

- Insert malicious driver in *BootDriverList* of *KeLoaderBlock* structure
- When kernel receives control it calls entry point of each module in the *BootDriverList*



# Win64/Rovnix: Abusing Debugging Facilities

## Win64/Rovnix:

- hooks Int 1h
  - ✓ tracing
  - ✓ handles hardware breakpoints (DR0-DR7)
- overwrites the last half of IDT (*Interrupt Descriptor Table*)
  - ✓ is not used by OS

As a result the malware is able to:

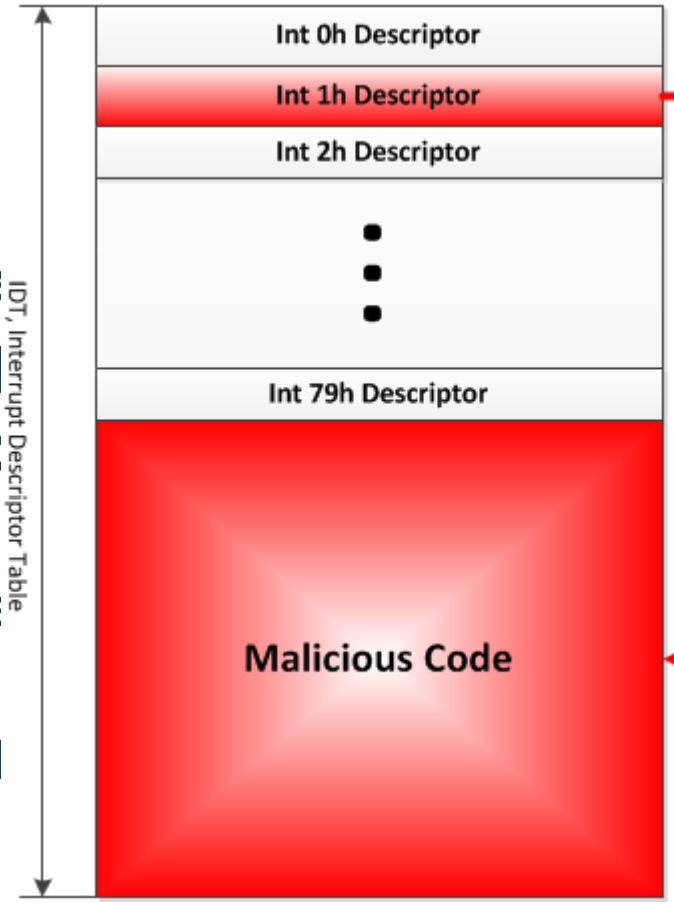
- ✓ set up hooks without patching bootloader components
- ✓ retain control after switching into protected mode

# Win64/Rovnix: Abusing Debugging Facilities

## Win64/Rovnix:

- hooks Int 1h
  - ✓ tracing
  - ✓ handles hardwa
- overwrites the I *Descriptor Table*)
  - ✓ is not used by (

As a result the ma
- ✓ set up hooks :omponents
- ✓ retain control mode



# Olmarik vs Rovnix

| Characteristics | Win64/Olmarik | Win64/Rovnix |
|---|---|---|
| **Privilege escalation** | MS10-092 | ☒ |
| **Reboot technique** | *ZwRaiseHardError* API | *ExitWindowsEx* API |
| **MBR/VBR infection** | MBR | VBR (bootstrap code) |
| **Loading driver** | *ZwCreateDriver* API | Inserting into boot driver list of *KeLoaderBlock* structure |
| **Payload injection** | *KeInitializeApc/ KeInstertQueueApc* APIs | *KeInitializeApc/ KeInstertQueueApc* APIs |
| **Self-defense** | Kernel-mode hooks, MBR monitoring | ☒ |
| **Number of modules** | 10 | 2 |
| **Stability of code** | ■■■■□ | ■■■□□ |
| **Threat complexity** | ■■■□□ | ■■■■□ |


# Bootkit Attack Vector

# Modern Bootkits' Approaches

- **Hooking BIOS 13h Interrupt Handler**
  - ✓ **Win64/Olmarik**
- **Tracing Bootloader Components**
  - ✓ **Win64/Rovnix**
  - ✓ **"Deep Boot" (PoC)**
- **Stealing a Processor's Core**
  - ✓ **"EvilCore" (PoC)**

# Tracing Bootloader Components

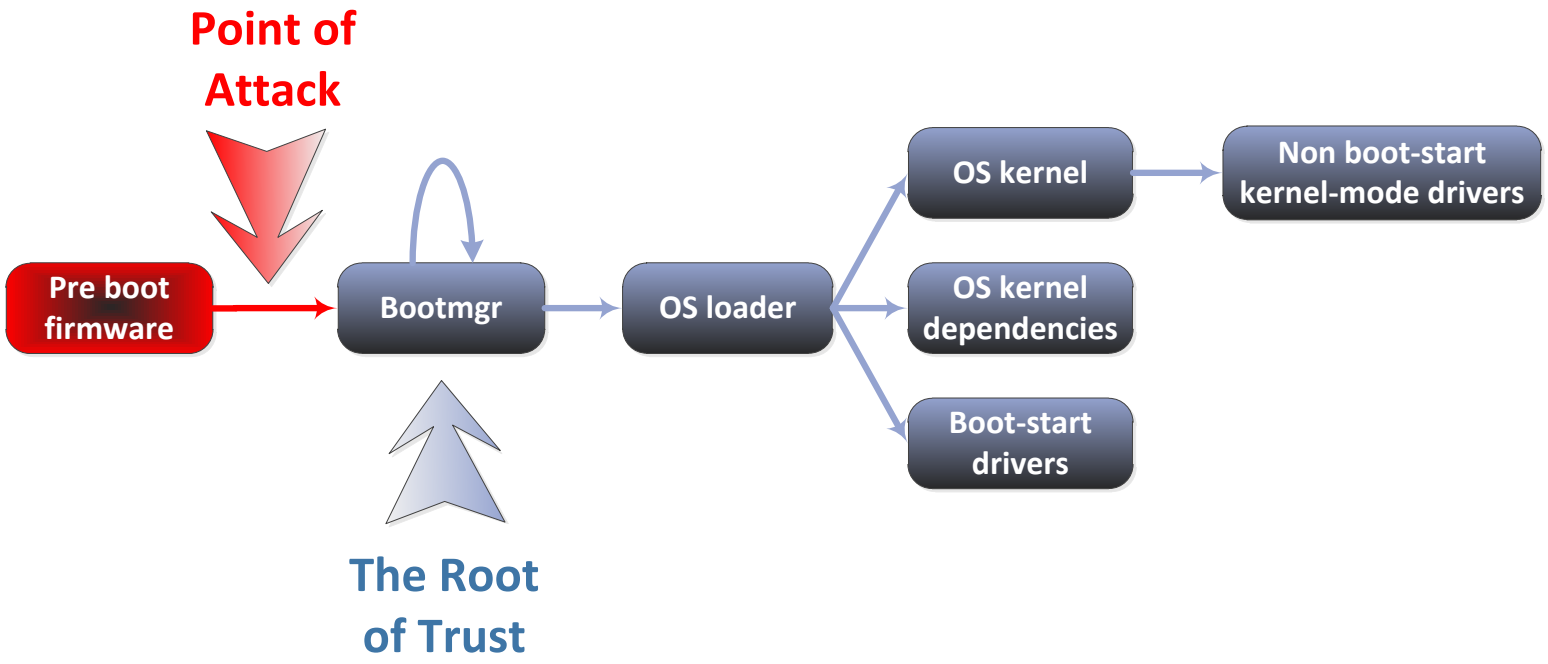
- **Microsoft Windows Bootloader Components:**

| Component Name | Processor Execution Mode |
|---|---|
| Bootstrap code | real mode |
| Bootmgr | real mode/protected mode |
| Winload.exe/Winresume.exe | protected mode |

- **Surviving processor's execution mode switching**
  - ✓ Malware has to retain control after execution mode switching
  - ✓ IDT and GDT are most frequently abused data structures

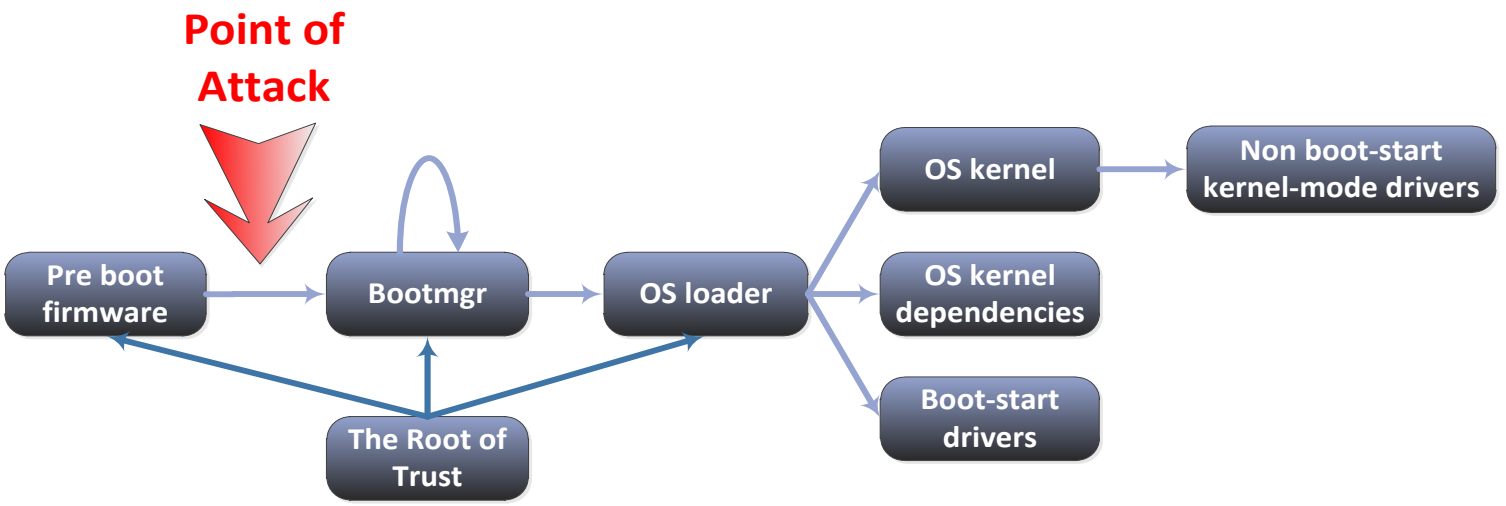
# What Facilitates the Attack Vector?

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



# How to Defend Against the Attack?

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



# Conclusion

- ✓ **Bootkits → ability to bypass KMCSP**
- ✓ **Return of old-school techniques → MBR infections**
- ✓ **Win64/Olmarik (TDL4) → 1st widely spread Win64 rootkit**
- ✓ **Win64/Rovnix → debugging facilities to subvert KMCSP**
- ✓ **Untrusted platform facilitates bootkit techniques**

# References

- ✓ **“The Evolution of TDL: Conquering x64”**

  http://www.eset.com/us/resources/white-papers/The_Evolution_of_TDL.pdf

- ✓ **“Defeating x64: The Evolution of the TDL Rootkit”**

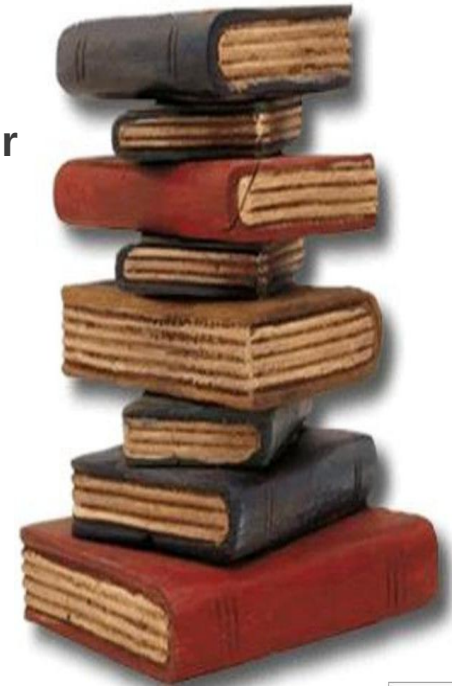
  http://www.eset.com/us/resources/white-papers/TDL4-CONFidence-2011.pdf

- ✓ **“Hasta La Vista, Bootkit: Exploiting the VBR”**

  http://blog.eset.com/2011/08/23/hasta-la-vista-bootkit-exploiting-the-vbr

- ✓ **Follow ESET Threat Blog**

  http://blog.eset.com



# Thank you for your attention ;)


**Aleksandr Matrosov**
**matrosov@eset.sk**
**@matrosov**

**Eugene Rodionov**
**rodionov@eset.sk**
**@vxradius**