NUCLEAR EXPLOTION

Alexander Ermolov (@flothrone)
Ruslan Zakirov (@ttbr0)

ZERO NIGHTS 2018

2^3 EDITION

#whoarewe

Security Research team at @ embedi

- Intel ME
  - https://embedi.com/resources/what-you-need-to-know-about-the-intel-amt-vulnerability/
- Intel Boot Guard
  - https://embedi.com/blog/bypassing-intel-boot-guard/
- UEFI BIOS (SMM)
  - https://embedi.com/blog/uefi-bios-holes-so-much-magic-dont-come-inside/

EMBEDI

ZERO NIGHTS 2018

2³ EDITION

#whoarewe

Solutions Support

intel

Report a Vulnerability Product Support

Summary:

A potential security vulnerability in Intel® NUC EBU firmware update executable may allow denial of service or information disclosure. Intel is releasing firmware kit updates to mitigate this potential vulnerability.

Intel® NUC EBU firmware

from INTEL-SA-00168 security advisory

https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00168.html

2^3 EDITION

#agenda

- BIOS Security
- Intel NUC BIOS update analysis
  - Update process architecture
  - Pwning
- Bypassing Intel Boot Guard
- Attacking from userland

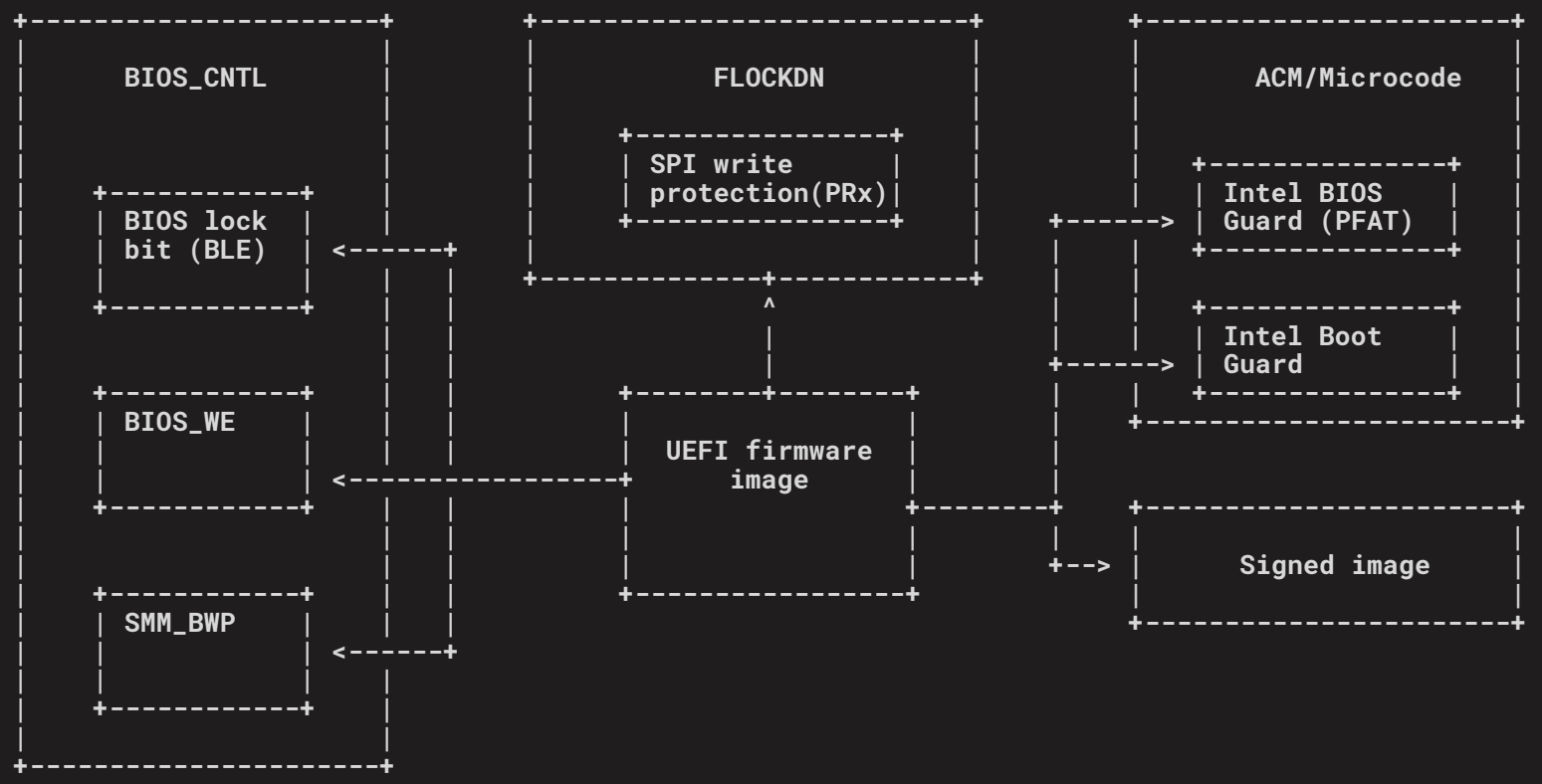
2^3 EDITION

BIOS SECURITY

EDITION

#bios_security

- Low level PC/server attack papers collection by Xeno Kovah @xenokovah
  https://timeglider.com/timeline/5ca2daa6078caaf4
- UEFI Firmware Rootkits: Myths and Reality by Alex Matrosov @matrosov and Eugene Rodionov @vxradius
  https://www.blackhat.com/docs/asia-17/materials/asia-17-Matrosov-The-UEFI-Firmware-Rootkits-Myths-And-Reality.pdf
- LOJAX - First UEFI rootkit found in the wild by ESET @ESET
  https://www.welivesecurity.com/wp-content/uploads/2018/09/ESET-LoJax.pdf



#bios_security

Betraying the BIOS by Alex Matrosov @matrosov
https://github.com/REhints/Publications/tree/master/Conferences/Betraying%20the%20BIOS

2^3 EDITION

BIOS UPDATE PROCESS

ZERO NIGHTS 2018

2^3 EDITION

Our sufferer

Intel® NUC Kit NUC7i3BNH

Intel® Core™ i3-7100U Processor (3M Cache, 2.40 GHz) – Kaby Lake

AMI BIOS

2^3 EDITION

NUC BIOS write protections

- ✓ BIOS Write Enable (BIOS_WE / BLE)
- ✓ SMM BIOS Write Protection (SMM_BWP)
- ✓ SPI Protected Ranges (PRx)
- ✓ Signed Capsule

```
BIOS Region Write Protection
------------------------------------------------------------
[*] BC = 0x00000AAA << BIOS Control (b:d.f 00:31.5 + 0xDC)
    [00] BIOSWE           = 0 << BIOS Write Enable
    [01] BLE              = 1 << BIOS Lock Enable
    [02] SRC              = 2 << SPI Read Configuration
    [04] TSS              = 0 << Top Swap Status
    [05] SMM_BWP          = 1 << SMM BIOS Write Protection
    [06] BBS              = 0 << Boot BIOS Strap
    [07] BILD             = 1 << BIOS Interface Lock Down

SPI Protected Ranges
------------------------------------------------------------
PRx (offset) | Value    | Base     | Limit    | WP? | RP?
------------------------------------------------------------
PR0 (84)     | 87FF0240 | 00240000 | 007FFFFF | 1   | 0
PR1 (88)     | 00000000 | 00000000 | 00000000 | 0   | 0
PR2 (8C)     | 00000000 | 00000000 | 00000000 | 0   | 0
PR3 (90)     | 00000000 | 00000000 | 00000000 | 0   | 0
PR4 (94)     | 00000000 | 00000000 | 00000000 | 0   | 0
```

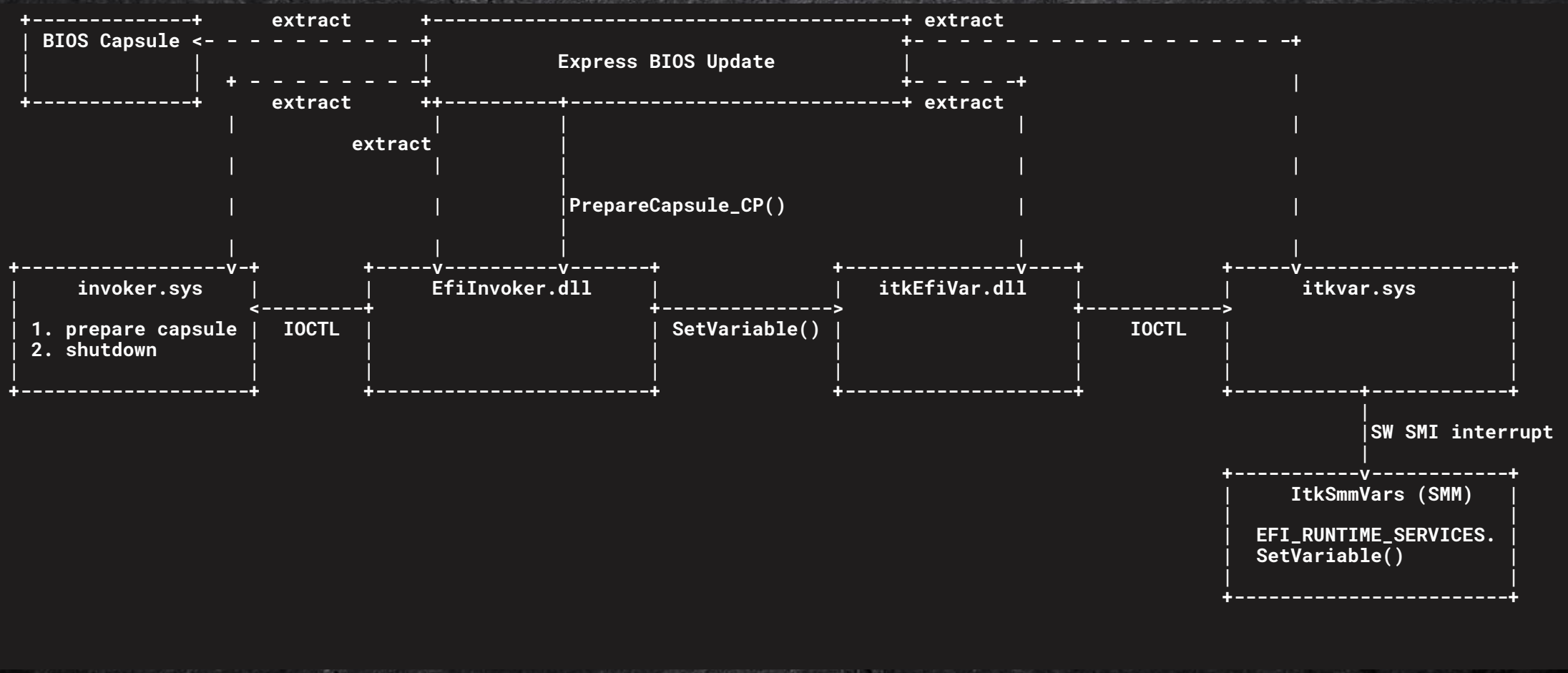
CHIPSEC framework

https://github.com/chipsec/chipsec


ZERO
NIGHTS
2018

2 3
EDITION

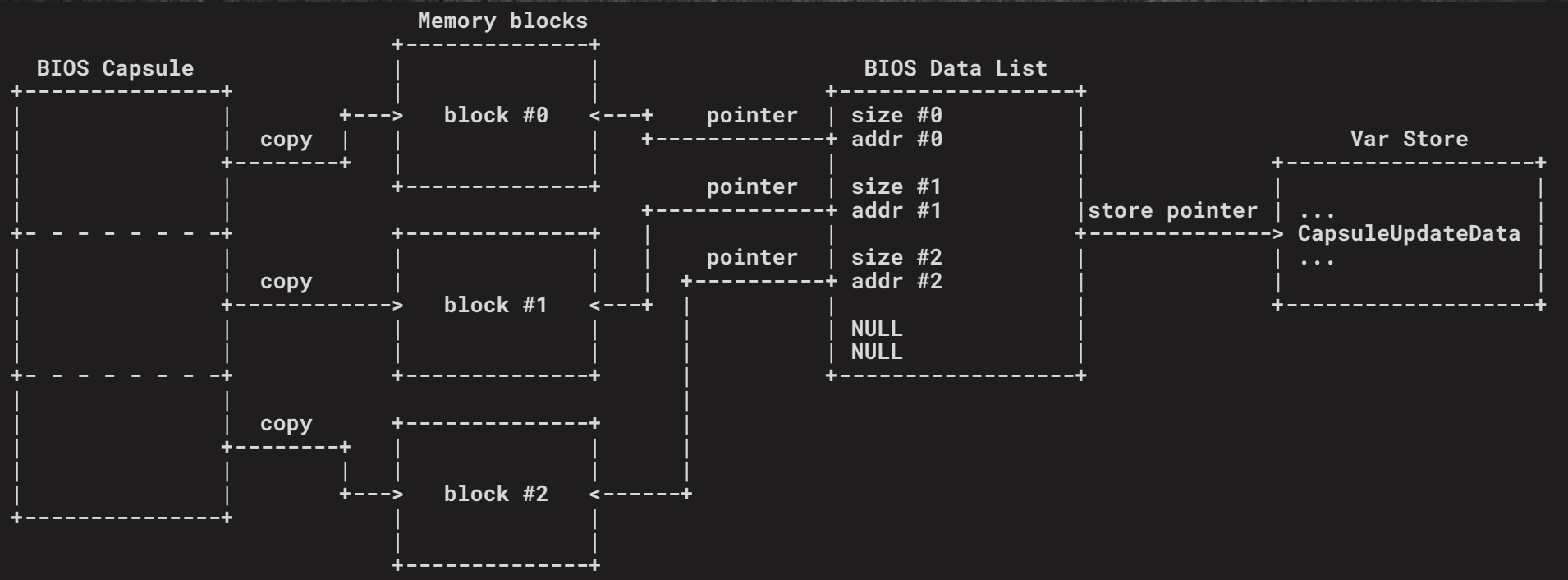
#bios_update


BIOS Capsule
extract
extract
extract
extract
Express BIOS Update
extract
PrepareCapsule_CP()
invoker.sys
1. prepare capsule
2. shutdown
IOCTL
EfiInvoker.dll
SetVariable()
itkEfiVar.dll
IOCTL
itkvar.sys
SW SMI interrupt
ItkSmmVars (SMM)
EFI_RUNTIME_SERVICES.
SetVariable()

2^3 EDITION

#bios_update

- Express BIOS Update executable
  - extracts itkEfiVar.dll, EfiInvoker.dll, invoker.sys, itkvar.sys and their 64-bit brothers
  - calls `PrepareCapsule_CP()` of EfiInvoker.dll
- EfiInvoker.dll
  - installs invoker.sys driver and calls it through `DeviceIoControl()`
  - calls `SetVariable()` of itkEfiVar.dll
  - shuts down the system with the help of invoker.sys
- invoker.sys
  - prepares “BIOS Data List”
- itkEfiVar.dll
  - installs itkvar.sys to call `SetVariable()` through SW SMI


Memory blocks
BIOS Capsule
BIOS Data List
Var Store
copy
copy
copy
block #0
block #1
block #2
pointer
pointer
pointer
size #0
addr #0
size #1
addr #1
size #2
addr #2
NULL
NULL
store pointer
...
CapsuleUpdateData
...

#bios_update

#bios_update

- Divide the signed BIOS capsule into several blocks
- Save the physical addresses of blocks into a special structure - BIOS Data List
- Store the physical address of this structure into “CapsuleUpdateData” EFI variable
- Shutdown the system (looks like reboot)
- Enjoy the BIOS firmware update process

ZERO NIGHTS 2018

2^3 EDITION

CVE-2018-12158

INTEL-SA-00168

A tribute to: What makes OS drivers dangerous for BIOS?
by Alex Matrosov @matrosov
https://medium.com/@matrosov/dangerous-update-tools-c246f7299459

https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00168.html

EDITION

#itkvar.sys

```
case 0x9C402418:
      IOCTL_STRUCT *IoctlStruct = (ICTL_STRUCT *) Irp->AssociatedIrp.SystemBuffer;
      DWORD dest = IoctlStruct->dest;
      LARGE_INTEGER src = IoctlStruct->src;
      DWORD size = IoctlStruct->size;

      if (MmIsAddressValid(dest))
      {
        ptr = MmMapIoSpace(src, size, 0);
        if (ptr)
        {
          memmove(dest, ptr, size);
          MmUnmapIoSpace(ptr, size);
        }
```

```
typedef struct
{
        LARGE_INTEGER   src;
        DWORD           dest;
        DWORD           reserved;
        DWORD           size;
        DWORD           status;
} IOCTL_STRUCT;
```

ZERO NIGHTS 2018

2³ EDITION

CVE-2018-12176

INTEL-SA-00176

https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00176.html

EDITION

#the_great_updater

- Took UEFITool and Resource Hacker
- Tried to repack the updater executable with custom (modified) BIOS update capsule…

UEFITool https://github.com/LongSoft/UEFITool

Resource Hacker http://www.angusj.com/resourcehacker/

ZERO NIGHTS 2018

2^3 EDITION

The story of success

Flash update has completed successfully.

```
Flashing motherboard firmware:

Current revision:     BNKBL357.86A.0061.2017.1221.1952
Updating to revision: BNKBL357.86A.0063.2018.0413.1542

Preparing image for Intel(R) Management Engine firmware ... [done]
Preparing image for BackUp Recovery Block firmware ... [done]
Preparing image for Boot Block firmware ... [done]
Preparing image for Recovery Block firmware ... [done]
Preparing image for Main Block firmware ... [done]
Preparing image for Graphic firmware ... [done]
Preparing image for FV Data firmware ... [done]
Flashing image for Intel(R) Management Engine firmware ... [done]
Flashing image for BackUp Recovery Block firmware ... [done]
Flashing image for Boot Block firmware ... [done]
Flashing image for Recovery Block firmware ... [done]
Flashing image for Main Block firmware ... [done]
Flashing image for Graphic firmware ... [done]
Flashing image for FV Data firmware ... [done]

Flash update has completed successfully.
```

2^3 EDITION

#the_great_updater

Meet THE GREAT UPDATER

- PoC (updater) based on Python and CHIPSEC framework
  - with both Windows and Linux support!
- Uses ItkSmmVars SMM-driver to avoid usage of EFI-provided routine `SetVariable()`
  - works even if UEFI mode is off
- Will be available on GitHub (https://github.com/embedi)

DEMO 1

@author

2^3 EDITION

#the_great_updater

- Allows to modify SEC, PEI, DXE, SMM code
- Bypasses BIOS_WE / BLE / SMM_BWP / PRx
- Compromised BIOS won't be executed if Intel Boot Guard enabled (on some NUCs)

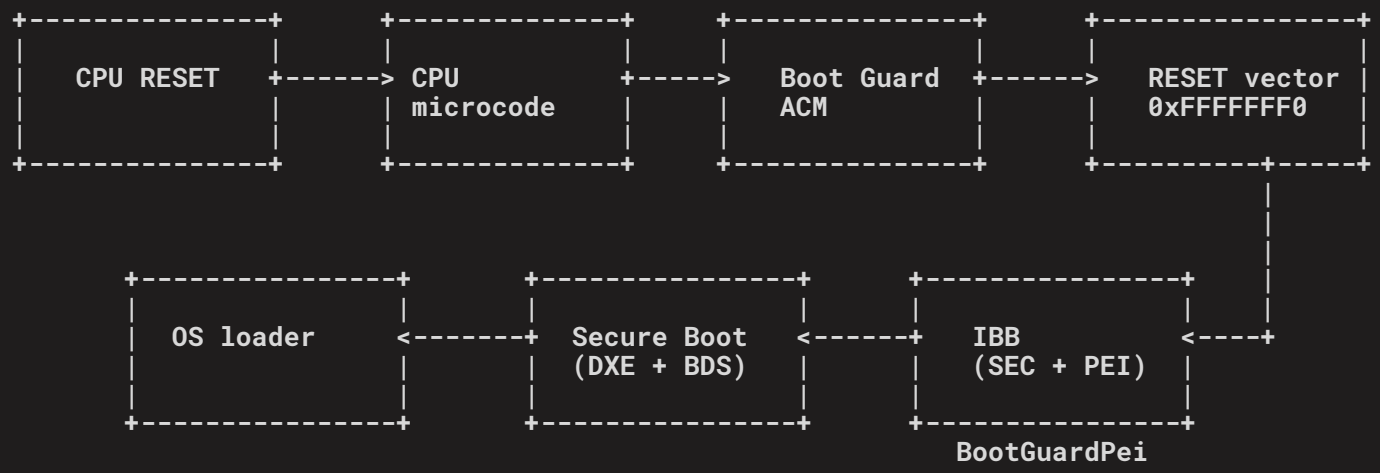
ZERO NIGHTS 2018

2^3 EDITION

Intel® Boot Guard

Bypass. Again.

2018.ZERONIGHTS.ORG



Betraying the BIOS by Alex Matrosov @matrosov

https://github.com/REhints/Publications/tree/master/Conferences/Betraying%20the%20BIOS

#boot_guard

2^3 EDITION

#boot_guard_bypass

- Bypassing Intel Boot Guard by Embedi @_embedi_
  - https://embedi.com/blog/bypassing-intel-boot-guard/
- Who Watch BIOS Watchers? by Alex Matrosov @matrosov
  - https://medium.com/@matrosov/bypass-intel-boot-guard-cc05edfca3a9

ZERO NIGHTS 2018

2^3 EDITION

CVE-2018-3623

INTEL-SA-?????

2^3 EDITION

#BootGuardPei

```
AmiPeiEndOfMrcCallBack(EFI_PEI_SERVICES **PeiServices)
{
            ...    // HOB GUID = {B60AB175-...}
      CreateHob(PeiServices, EFI_HOB_TYPE_GUID_EXTENSION, 0x19u, &BootGuardPeiHob);

      // Hash container GUID = {98DB68E0-5AB6-4A48-80C8-EAC6C51180FC}
      FindObjectInImageByGuid(&gBootGuardDxeHashContainerGuid, &HashContainer);

      Sha256Init(Buffer);
      Sha256Calc(Buffer, HashContainer.BlockBaseAddress, HashContainer.BlockSize);
      Sha256Out(Buffer, &CalculatedHash);

      if (memcmp(&HashContainer.BlockHash, &CalculatedHash, SHA256_DIGEST_SIZE))
            *(BootGuardPeiHob + 0x18) = 0;
      else
            *(BootGuardPeiHob + 0x18) = 0x10;
```

```
UEFI capsule        Caps… UEFI …
 UEFI image         Image UEFI
  EfiFirmware…      Volu… FFSv2
   D1157A19-7…      File  Volum…
    24400798…       Sect… GUID …
     Volume …       Sect… Volum…
      EfiFi…        Volu… FFSv2
       29FF…        File  DXE d…  DescUpdate
       3D93…        File  Freef…
       E002…        File  DXE d…  UpdateArea
       098D…        File  Freef…
       D005…        File  Freef…
       94B5…        File  DXE d…  FirmwareProgrammer
       6A46…        File  DXE d…  FirmwareTopSwap
       E75C…        File  DXE d…  BackUpRecoveryAreas
       ADB9…        File  Freef…
       E449…        File  DXE d…  BootBlockAreas
       AFCC…        File  Freef…
       CB7F…        File  DXE d…  RecoveryAreas
       5BA2…        File  Freef…
       9D8C…        File  DXE d…  MainAreas
       A90A…        File  Raw
       27DC…        File  DXE d…  GraphicAreas
       1150…        File  Freef…
       C3DB…        File  DXE d…  FlexUpdate
       7BA6…        File  DXE d…  FVDataAreas
       E011…        File  Freef…
       7898…        File  DXE d…  EcUpdateArea
       698C…        File  Freef…
```

ZERO NIGHTS 2018

2³ EDITION

#BootGuardPei

```
UnknownEventCallBack(EFI_PEI_SERVICES **PeiServices)
{
            ...     // HOB GUID = {B60AB175-...}
      BootGuardPeiHob = FindGuidExtensionHobInHobListByGuid(&BootGuardPeiHobGuid);

      // Hash container GUID = {CBC91F44-A4BC-4A5B-8696-703451D0B053}
      FindObjectInImageByGuid(&gBootGuardDxeHashContainer2Guid, &HashContainer);

      Sha256Init(Buffer);
      Sha256Calc(Buffer, HashContainer.BlockBaseAddress, HashContainer.BlockSize);
      Sha256Out(Buffer, &CalculatedHash);

      if (memcmp(&HashContainer.BlockHash, &CalculatedHash, SHA256_DIGEST_SIZE))
            *(BootGuardPeiHob + 0x18) = 0;  // The stored value (verification result)
      else                                  // is ignored!
            // Start Recovery!
```

```
UEFI capsule          Caps… UEFI …
 UEFI image           Image UEFI
  EfiFirmware…        Volu… FFSv2
   D1157A19-7…        File  Volum…
    24400798…         Sect… GUID …
     Volume …         Sect… Volum…
      EfiFi…          Volu… FFSv2
       29FF…          File  DXE d…  DescUpdate
       3D93…          File  Freef…
       E002…          File  DXE d…  UpdateArea
       098D…          File  Freef…
       D005…          File  Freef…
       94B5…          File  DXE d…  FirmwareProgrammer
       6A46…          File  DXE d…  FirmwareTopSwap
       E75C…          File  DXE d…  BackUpRecoveryAreas
       ADB9…          File  Freef…
       E449…          File  DXE d…  BootBlockAreas
       AFCC…          File  Freef…
       CB7F…          File  DXE d…  RecoveryAreas
       5BA2…          File  Freef…
       9D8C…          File  DXE d…  MainAreas
       A90A…          File  Raw
       27DC…          File  DXE d…  GraphicAreas
       1150…          File  Freef…
       C3DB…          File  DXE d…  FlexUpdate
       7BA6…          File  DXE d…  FVDataAreas
       E011…          File  Freef…
       7898…          File  DXE d…  EcUpdateArea
       698C…          File  Freef…
```

2^3 EDITION

BIOS PWNING

"user-friendly"

2^3 EDITION

#bios_pwning

- UACME to bypass UACM
  - https://github.com/hfiref0x/UACME
- Load signed kernel driver (RWEverything) for accessing R/W routines (both virtual and physical memory) in kernel space
  - http://rweverything.com/
  - https://github.com/Cr4sh/fwexpl
- Prepare "BIOS Data List" in physical memory
- Set "CapsuleUpdateData" EFI variable
- Reboot

DEMO 2

@author

#conclusions

Multiple vulnerabilities in BIOS update scheme:

- CVE-2018-12158 (kernel driver)
- CVE-2018-12176 (update process)
- CVE-2018-3623 (Intel Boot Guard)

Can be easily exploited from user space to compromise BIOS.

ZERO NIGHTS 2018

2^3 EDITION

#mitigations

- Blacklist potentially vulnerable drivers (keep your OS updated)
- Keep your BIOS updated 😉

Thanks

@author