Vulnerability Research on SMM

AMOSSYS

Overview

- What is the SMM?
- Security mechanisms
- Where to start?
- What to look for?
- Example
- Resources

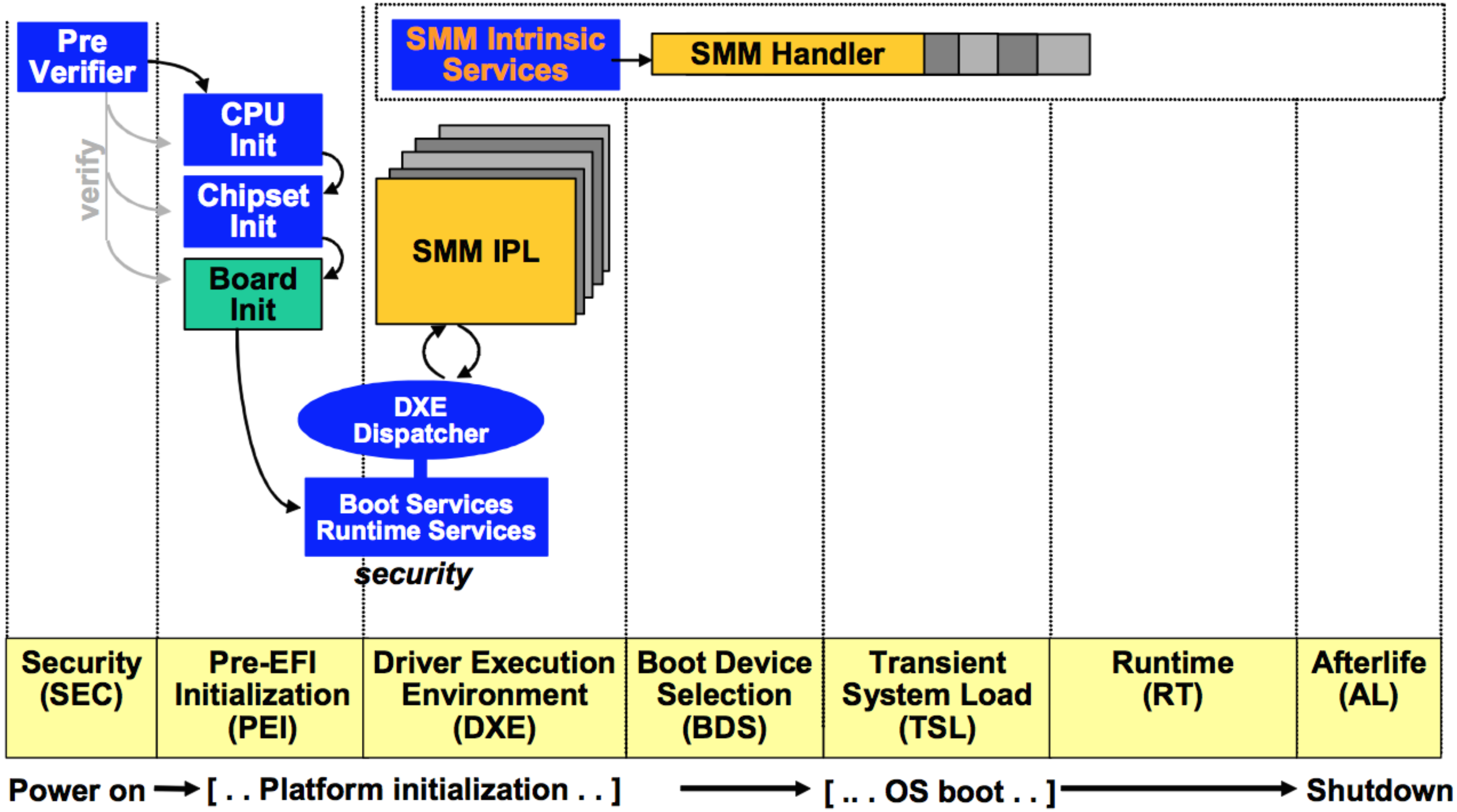
What is the SMM?

- **System Management Mode, Intel processors**
- **From Intel SDM Vol. 3C (chapter 34)**

SMM provides an alternate operating environment that can be used to monitor and manage various system resources for more efficient energy usage, to control system hardware, and/or to run proprietary code.

- **Interrupt mechanisms to switch mode**
  - System Management Interrupt (SMI)
  - Can be HW or SW SMI
- **Registers content saved into SMRAM**

What is the SMM?



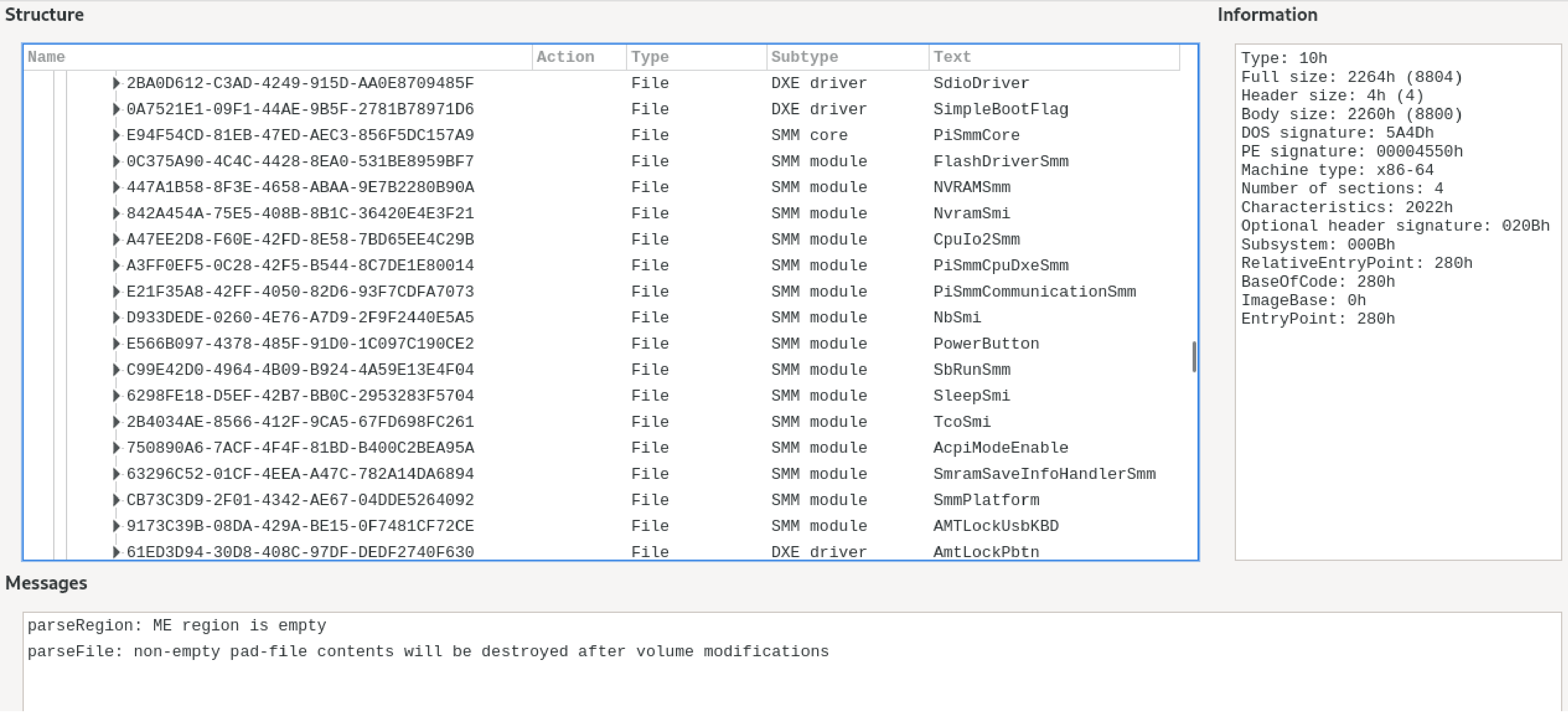
Security mechanisms

- **SMRAM isolation**
  - SMRAM Control Register (**SMRAMC**): *D_LCK* and *D_OPEN*
- **Block code execution outside of SMRAM**
  - MSR_SMM_FEAUTRE_CONTROL: *SMM_CODE_CHK_EN*
- **Other HW mitigations**
  - *TOLUD* register: separation between DRAM and MMIO
  - *SMM_BWP*: bit of the SMRAMC to prevent BIOS flash
  - *TSEG/BGSM*: protection against DMA
- **No software protections**

Where to start?

- **Get the firmware**
  - CHIPSEC or hardware flash reader
- **Identify the part related to the SMM**
  - UEFITool and UEFIExtract

Where to start?



Structure

Name	Action	Type	Subtype	Text
2BA0D612-C3AD-4249-915D-AA0E8709485F		File	DXE driver	SdioDriver
0A7521E1-09F1-44AE-9B5F-2781B78971D6		File	DXE driver	SimpleBootFlag
E94F54CD-81EB-47ED-AEC3-856F5DC157A9		File	SMM core	PiSmmCore
0C375A90-4C4C-4428-8EA0-531BE8959BF7		File	SMM module	FlashDriverSmm
447A1B58-8F3E-4658-ABAA-9E7B2280B90A		File	SMM module	NVRAMSmm
842A454A-75E5-408B-8B1C-36420E4E3F21		File	SMM module	NvramSmi
A47EE2D8-F60E-42FD-8E58-7BD65EE4C29B		File	SMM module	CpuIo2Smm
A3FF0EF5-0C28-42F5-B544-8C7DE1E80014		File	SMM module	PiSmmCpuDxeSmm
E21F35A8-42FF-4050-82D6-93F7CDFA7073		File	SMM module	PiSmmCommunicationSmm
D933DEDE-0260-4E76-A7D9-2F9F2440E5A5		File	SMM module	NbSmi
E566B097-4378-485F-91D0-1C097C190CE2		File	SMM module	PowerButton
C99E42D0-4964-4B09-B924-4A59E13E4F04		File	SMM module	SbRunSmm
6298FE18-D5EF-42B7-BB0C-2953283F5704		File	SMM module	SleepSmi
2B4034AE-8566-412F-9CA5-67FD698FC261		File	SMM module	TcoSmi
750890A6-7ACF-4F4F-81BD-B400C2BEA95A		File	SMM module	AcpiModeEnable
63296C52-01CF-4EEA-A47C-782A14DA6894		File	SMM module	SmramSaveInfoHandlerSmm
CB73C3D9-2F01-4342-AE67-04DDE5264092		File	SMM module	SmmPlatform
9173C39B-08DA-429A-BE15-0F7481CF72CE		File	SMM module	AMTLockUsbKBD
61ED3D94-30D8-408C-97DF-DEDF2740F630		File	DXE driver	AmtLockPbtn

Information

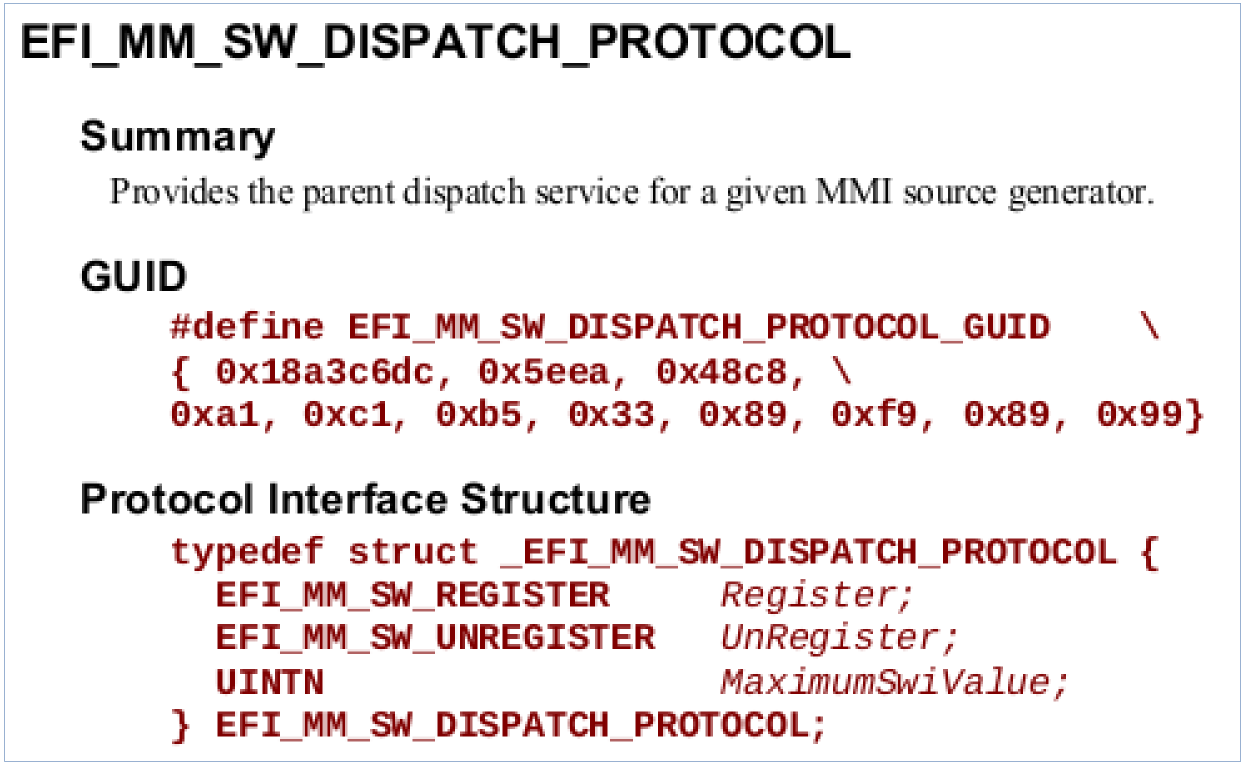
```
Type: 10h
Full size: 2264h (8804)
Header size: 4h (4)
Body size: 2260h (8800)
DOS signature: 5A4Dh
PE signature: 00004550h
Machine type: x86-64
Number of sections: 4
Characteristics: 2022h
Optional header signature: 020Bh
Subsystem: 000Bh
RelativeEntryPoint: 280h
BaseOfCode: 280h
ImageBase: 0h
EntryPoint: 280h
```

Messages

```
parseRegion: ME region is empty
parseFile: non-empty pad-file contents will be destroyed after volume modifications
```

What to look for?

- **Protocols and GUIDs**



EFI_MM_SW_DISPATCH_PROTOCOL

Summary

Provides the parent dispatch service for a given MMI source generator.

GUID

```
#define EFI_MM_SW_DISPATCH_PROTOCOL_GUID    \
{ 0x18a3c6dc, 0x5eea, 0x48c8, \
0xa1, 0xc1, 0xb5, 0x33, 0x89, 0xf9, 0x89, 0x99}
```

Protocol Interface Structure

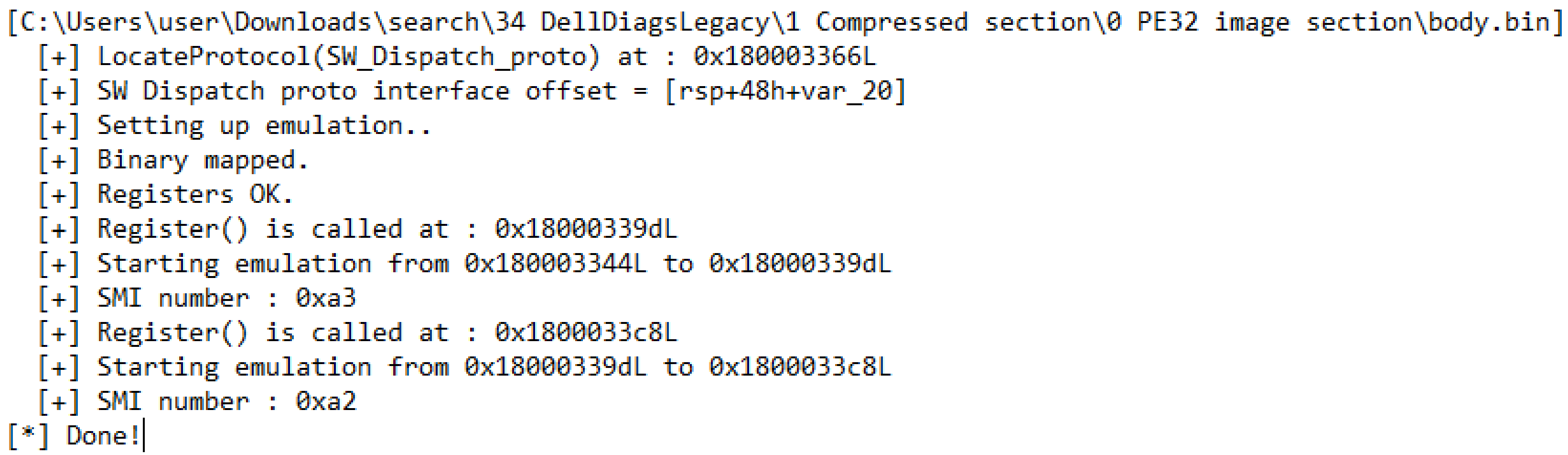
```
typedef struct _EFI_MM_SW_DISPATCH_PROTOCOL {
  EFI_MM_SW_REGISTER      Register;
  EFI_MM_SW_UNREGISTER    UnRegister;
  UINTN                   MaximumSwiValue;
} EFI_MM_SW_DISPATCH_PROTOCOL;
```

What to look for?

- **Look for modules which:**
  - register SMI(s);
  - are using a given protocol;
  - are manipulating data from OS (e.g. registers).

What to look for?

- **Custom scripts**
  - Extract subfolders of binaries containing a given GUID
    - Grep like
  - SMI extraction automation
    - Static analysis (IDA API) + Emulation (Unicorn Engine)



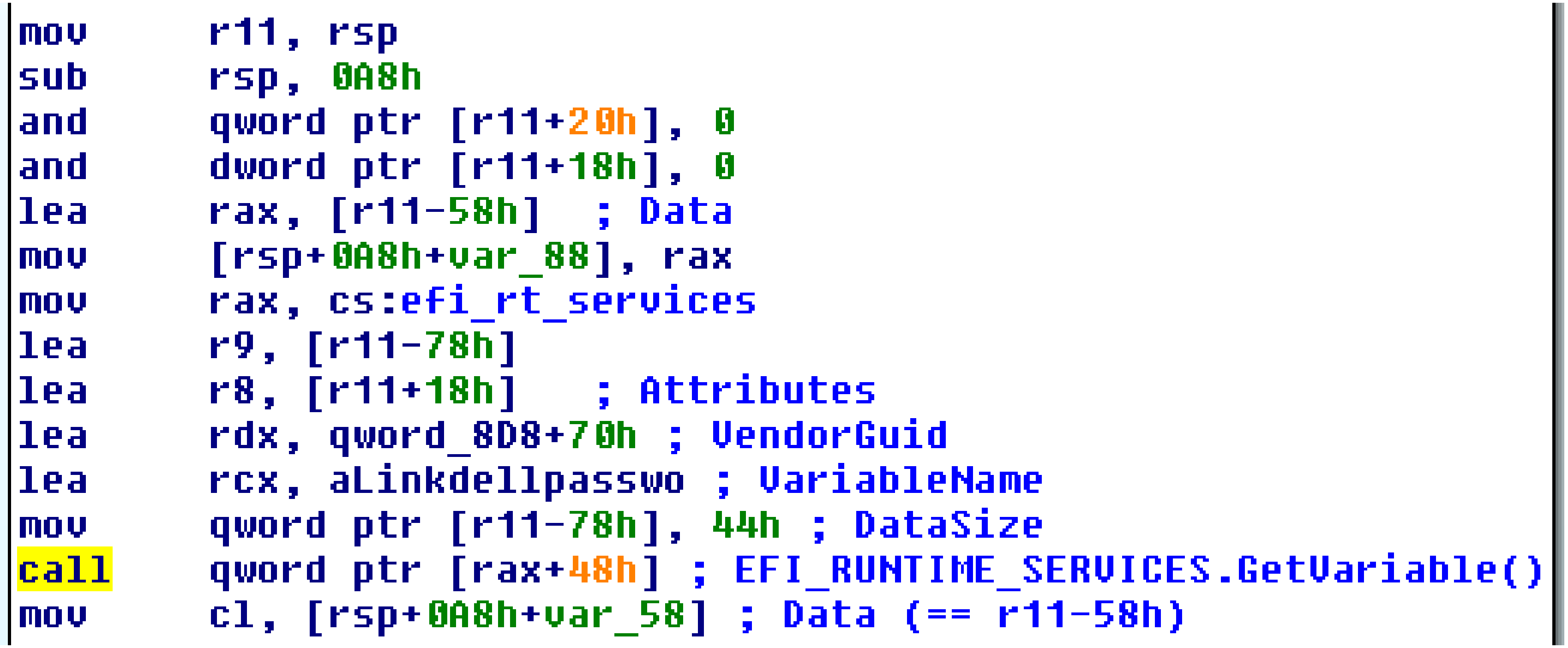
```
[C:\Users\user\Downloads\search\34 DellDiagsLegacy\1 Compressed section\0 PE32 image section\body.bin]
  [+] LocateProtocol(SW_Dispatch_proto) at : 0x180003366L
  [+] SW Dispatch proto interface offset = [rsp+48h+var_20]
  [+] Setting up emulation..
  [+] Binary mapped.
  [+] Registers OK.
  [+] Register() is called at : 0x18000339dL
  [+] Starting emulation from 0x180003344L to 0x18000339dL
  [+] SMI number : 0xa3
  [+] Register() is called at : 0x1800033c8L
  [+] Starting emulation from 0x18000339dL to 0x1800033c8L
  [+] SMI number : 0xa2
[*] Done!
```

Example

- **Dell laptop firmware**
  - 437 modules
    - 295 DXE drivers
    - 142 SMM modules! (33 SW SMI)

Example

- **OemLinkDellPwdLib**
  - *SMM_CODE_CHK_EN* = enabled
  - Exception if code outside of SMRAM is executed



```
mov     r11, rsp
sub     rsp, 0A8h
and     qword ptr [r11+20h], 0
and     dword ptr [r11+18h], 0
lea     rax, [r11-58h]  ; Data
mov     [rsp+0A8h+var_88], rax
mov     rax, cs:efi_rt_services
lea     r9, [r11-78h]
lea     r8, [r11+18h]   ; Attributes
lea     rdx, qword_8D8+70h ; VendorGuid
lea     rcx, aLinkdellpasswo ; VariableName
mov     qword ptr [r11-78h], 44h ; DataSize
call    qword ptr [rax+48h] ; EFI_RUNTIME_SERVICES.GetVariable()
mov     cl, [rsp+0A8h+var_58] ; Data (== r11-58h)
```

Example

- **Call to a function from the UEFI Runtime Service table**
- **Should trigger an MCE right?**
- **Investigate from an UEFI shell**
  - Get address of the function
  - Replace instruction by shellcode => executed!
  - Let's read *IA32_SMBASE* and dump the SMRAM => Nope..

Example

- **Subtelty in the UEFI specs**
  - Set of functions that may be called after MCE, INIT and NMI
  - GetTime(), ***GetVariable()***, UpdateCapsule(), etc.
- **So no code execution afterall..**
- **The MCE handler must switch processor mode**

Conclusion

- Massive amount of code
- Lots of intricacies
- Actual functionalities are not obvious
- Very dense ecosystem

Resources

- **Blogs**
  - http://blog.cr4.sh/
  - https://www.synacktiv.com/posts/exploit/code-checkmate-in-smm.html
- **Documentation**
  - Intel Software Developer Manual 3C (chapter 34)
  - UEFI Specification
  - UEFI Platform Initialization (PI) Specification
- **Tools**
  - EDKII (for UEFI development + lots of papers)
  - UEFITools & UEFIExtract
  - CHIPSEC
  - github.com/mdolmen/smm_research

Thank you!

- **Contact**
  - mathieu [dot] dolmen [at] gmail.com
  - Twitter: @StrikeBhack