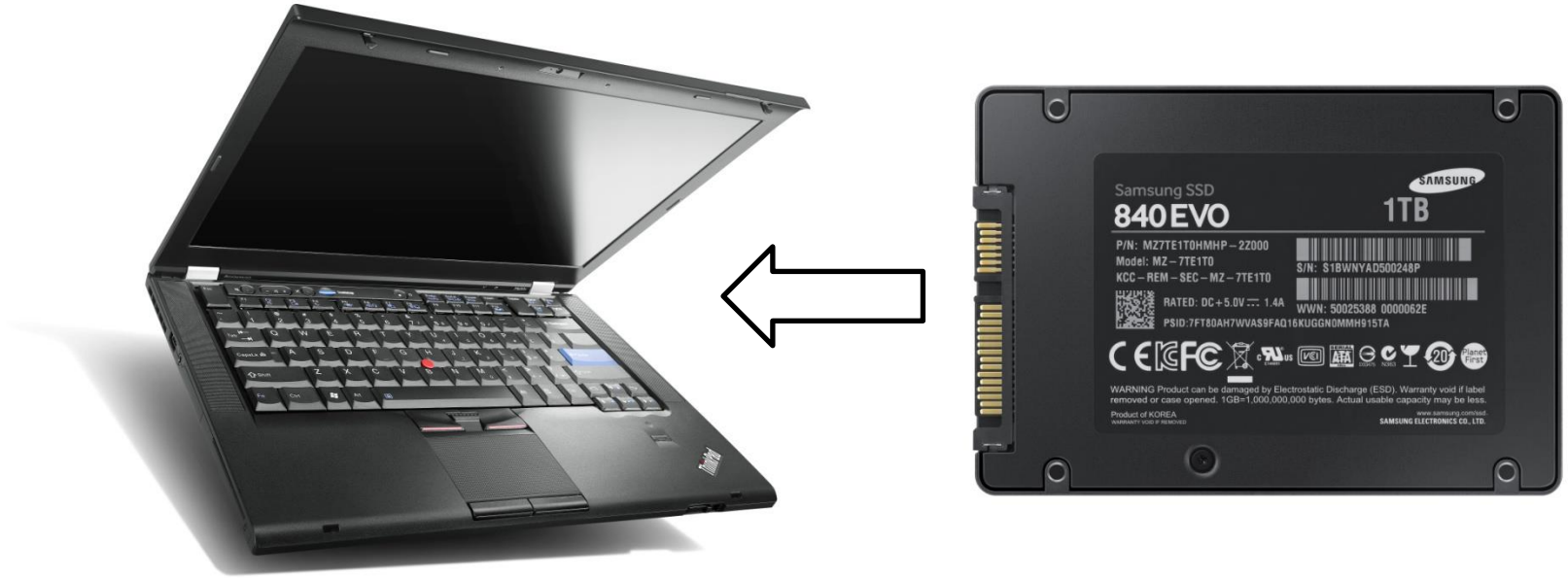
Reverse engineering UEFI by execution

Jethro Beekman

32c3@jbeekman.nl



correct horse battery staple galaxy peace position require house

(64 characters)

correct horse battery staple galaxy peace position require house

(64 characters)

Table 11 – SECURITY UNLOCK data

Word	Content
0	Control word Bit — Field Name — Description 0 — Identifier — 0=compare User password; 1=compare Master password (15:1) — Reserved
1-16	Password (32 bytes)
17-255	Reserved

Password (32 bytes)

Reverse engineering UEFI by execution

-or-

How to turn 64 characters into 32 bytes

UEFI RE challenges

→no debugger

→no system calls

→no dynamic symbols

→**No good starting point!**

LenovoVideoInitDxe.efi
LenovoWmaPciDxe.efi
LenovoWmaPolicyDxe.efi
LenovoWmaUsbDxe.efi
M25PX64FlashPartDxe.efi
M25PX64FlashPartSmm.efi
MeConfig.efi
MeConOutReady.efi
MeFwDowngrade.efi
MePciUpdate.efi
ModulesSmmThunkSmm.efi
MpCpu.efi
MX25L6445EFlashPartDxe.efi
MX25L6445EFlashPartSmm.efi
NationalLpcPc87393.efi
PchEmulateKbc.efi
PchIdeDeviceDetect.efi
PchInitDxe.efi

ThinkPad firmware

281 modules

UEFITool demo

```
EFI_STATUS main(EFI_SYSTEM_TABLE*, ...);

struct EFI_SYSTEM_TABLE {
  EFI_SIMPLE_TEXT_INPUT_PROTOCOL  *ConIn;
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL *ConOut;
  EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL *StdErr;
  EFI_RUNTIME_SERVICES            *RuntimeServices;
  EFI_BOOT_SERVICES               *BootServices;
  ...
};

struct EFI_BOOT_SERVICES {
  EFI_STATUS (*InstallProtocolInterface)(EFI_GUID*, VOID*, ...);
  EFI_STATUS (*LocateProtocol)(EFI_GUID*, VOID**, ...);
  ...
};
```

Executing modules

Compatible instruction set: ✓

Compatible ABI: **efiperun**

Executing modules

Compatible instruction set: ✓

Compatible ABI: **efiperun**

- Many basic EFI API stubs, easy to add more
- Auto-generate missing APIs at runtime
- Memory map annotations
- Use standard debugger (e.g. gdb)
- Uses `cross-stdarg.h`

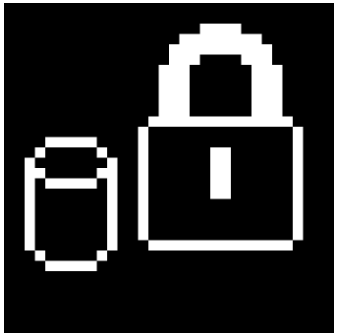
efiperun demo 1


lenovo
T420s

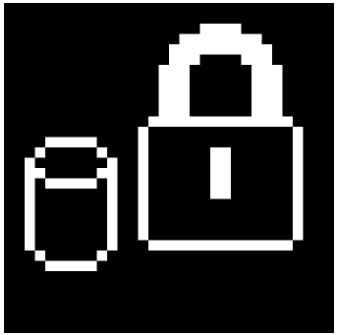
SystemImageDisplayDxe.efi
SystemGraphicsConsoleDxe.efi
SystemHiiImageDisplayDxe.efi
SystemImageDecoderDxe.efi

SystemImageDisplayDxe.efi

LenovoPromptService.efi



- SystemImageDisplayDxe.efi
  - LenovoPromptService.efi
  - LenovoSoundService.efi
  - LenovoTranslateService.efi
  - LenovoCryptService.efi
    - LenovoPasswordCp.efi



efiperun demo 2

How to turn 64 characters into 32 bytes

"correct horse battery staple galaxy peace position require house"

"S1DHNSAFB05849E"

"Samsung SSD 840 EVO 500GB"

$$AtaPassword \leftarrow \mathrm{SHA}_{256}(\mathrm{SHA}_{256}(Password) \parallel AtaIdentity_{SerialNumber} \parallel AtaIdentity_{ModelNumber})$$

How Lenovo turns 64 chars into 32 bytes

$$AtaPassword \leftarrow \mathrm{SHA}_{256}(\mathrm{SHA}_{256}(Password) \parallel AtaIdentity_{SerialNumber} \parallel AtaIdentity_{ModelNumber})$$

actually:

max 96 bits of entropy!

$$PasswordHash \leftarrow \mathrm{SHA}_{256}\left(\left(\mathrm{ToScanCodes}(\mathrm{LowerCase}(Password)) \parallel {}^{\mathrm{NUL}}{}^{64}\right)_{1:64}\right)_{1:12}$$

$$SN \leftarrow AtaIdentity_{SerialNumber} \qquad MN \leftarrow AtaIdentity_{ModelNumber}$$

$$AtaPassword \leftarrow \mathrm{SHA}_{256}(PasswordHash \parallel \mathrm{SwapBytes}(SN) \parallel \mathrm{SwapBytes}(MN))$$

Reverse engineering UEFI by execution

https://github.com/jethrogb/uefireverse (GPL)

https://jbeekman.nl/blog/uefi/

Jethro Beekman

32c3@jbeekman.nl