RSAConference2019
Asia Pacific & Japan

Singapore | 16–18 July | Marina Bay Sands

BETTER.

SESSION ID: FLE-R04

Finding and Analyzing In-the-Wild UEFI Rootkits Assisted by Machine Learning

Jean-Ian Boutin

Head of Threat Research
ESET
@jiboutin

Filip Mazán

Software Engineer
ESET

Why should you care about UEFI malware?

- No longer theoretical
- Difficult to detect
- Difficult to eradicate

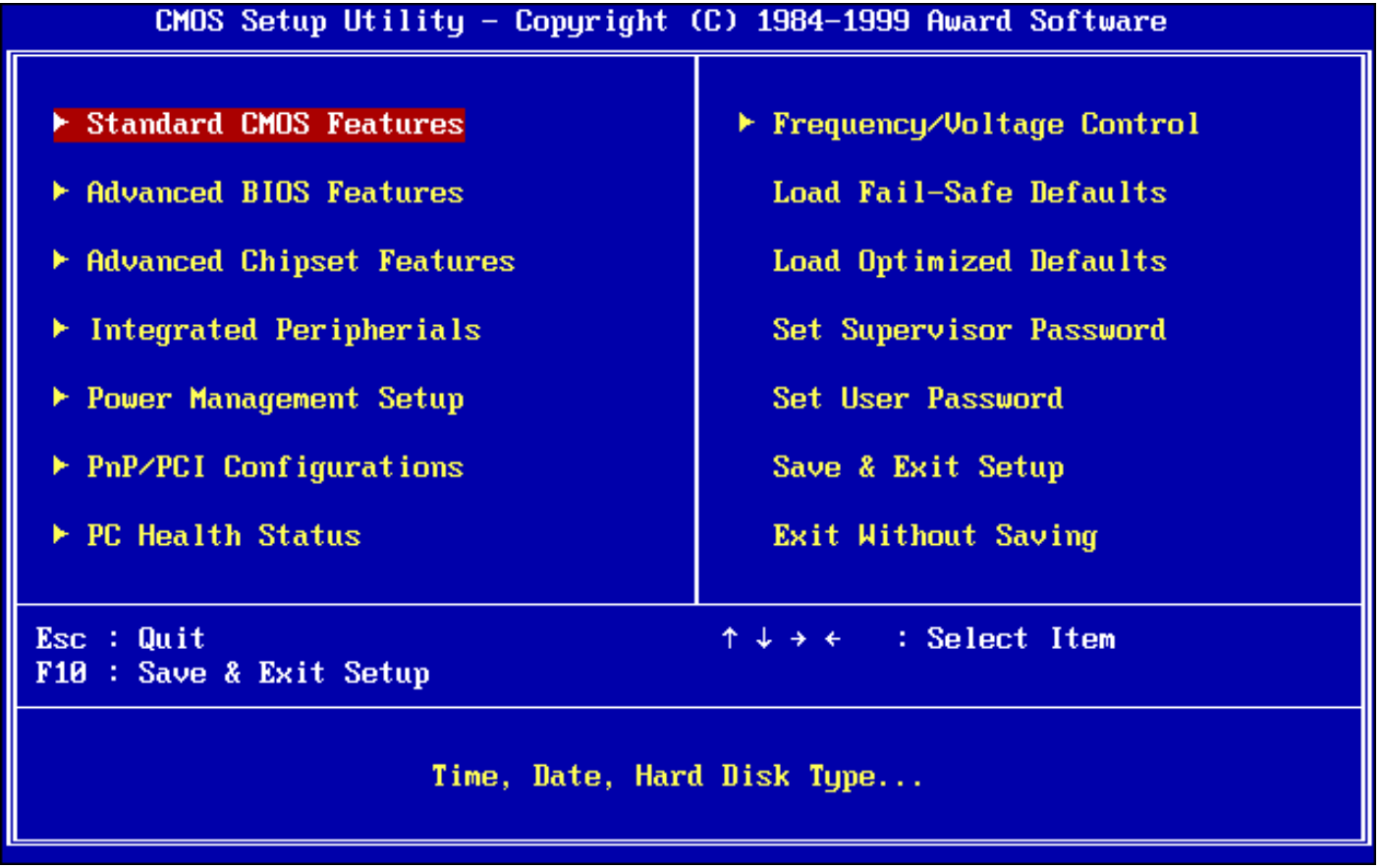
RSAConference2019
Asia Pacific & Japan

UEFI Introduction

UEFI

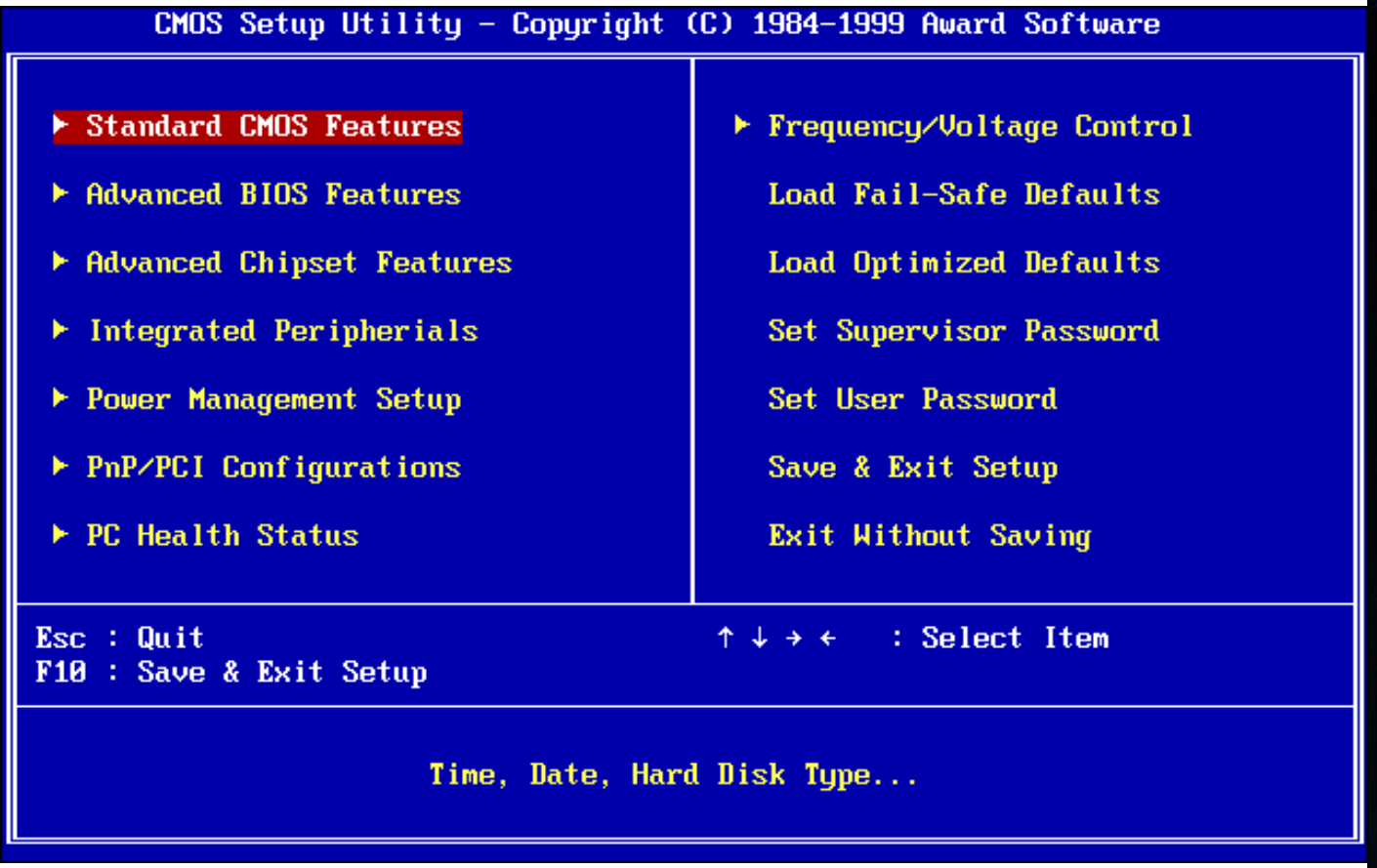
- Evolution
- < 2000: proprietary BIOS
- 2000: Intel creates the Extensible Firmware Interface (EFI)
- 2005: Industry coalition releases Universal Extensible Firmware Interface (UEFI)

BIOS



Source: howtogeek.com

BIOS



UEFI

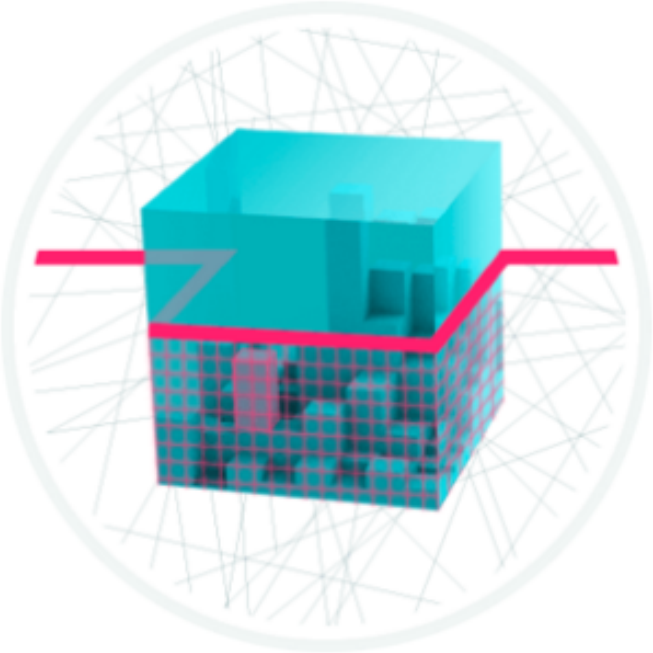
Source: howtogeek.com

Driver Execution Environment (DXE) Drivers

- PE/COFF images
- Abstract the hardware
- Produce UEFI standard interface
- Register new services (protocols)
- Loaded during the DXE phase of the Platform initialization
- Loaded by the DXE dispatcher (DXE Core)

UEFI firmware layout

- Located in the BIOS region of the SPI flash memory
- Contains multiple volumes
  - Volumes contain files identified by GUIDs
  - File contain sections
  - One of these sections is the actual UEFI image
  - It’s more complex than that but it suffices for our purpose



UEFI Scanner

ESET is the first internet security provider to add a dedicated layer into its solution that protects the Unified Extensible Firmware Interface (UEFI). ESET UEFI Scanner checks and enforces the security of the pre-boot environment that is compliant with the UEFI specification. It is designed to monitor the integrity of the firmware and in case modification is detected, it notifies the user.

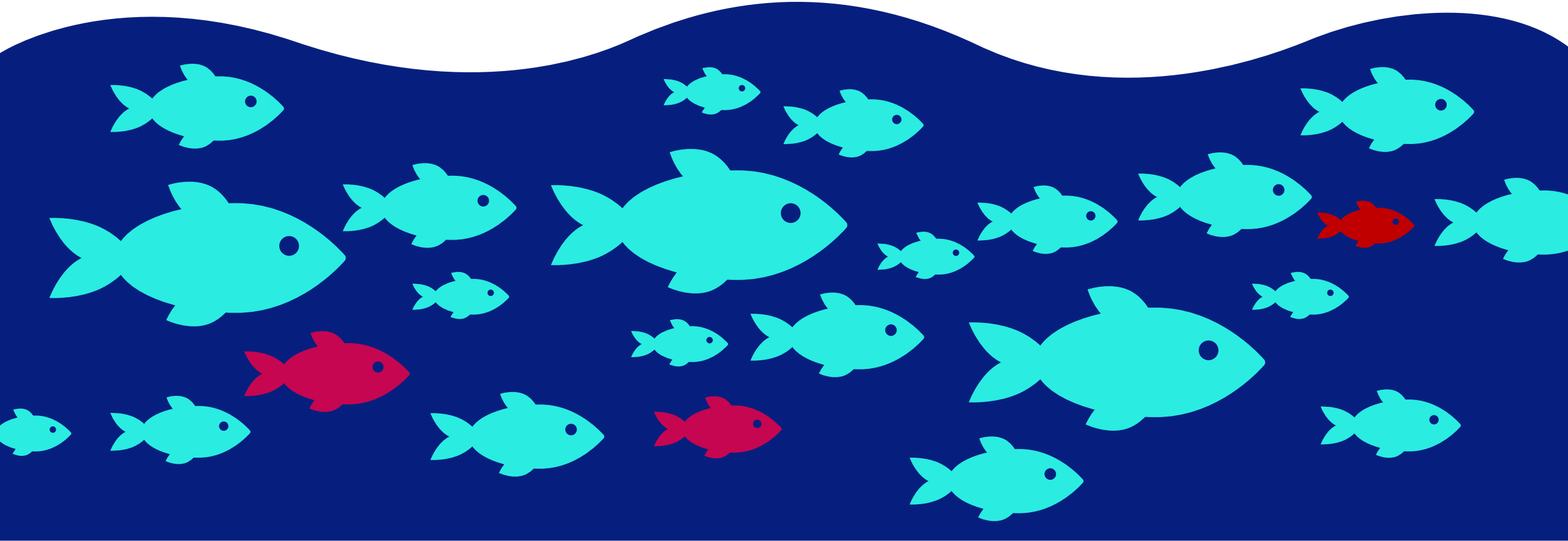
Show more

RSAConference2019
Asia Pacific & Japan

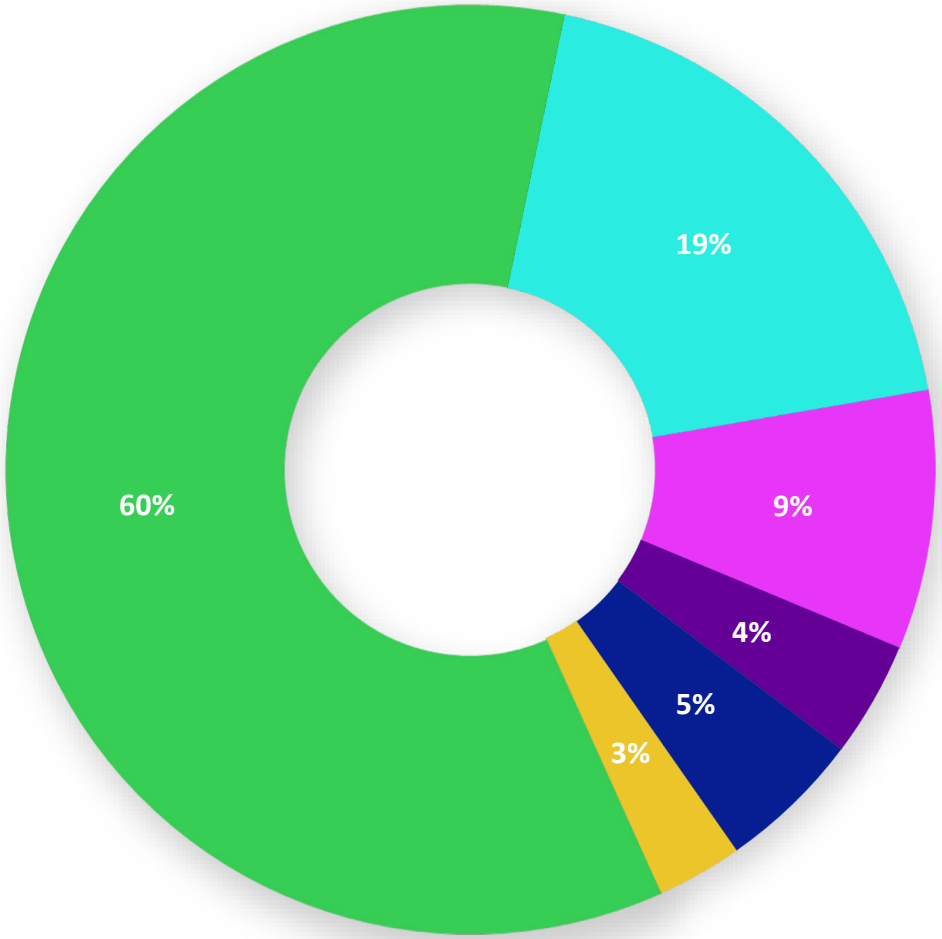
Using machine learning to find needles in a haystack

Motivation

- Millions of samples
- Finding malicious fish in a sea of UEFI executables



Everything is about data

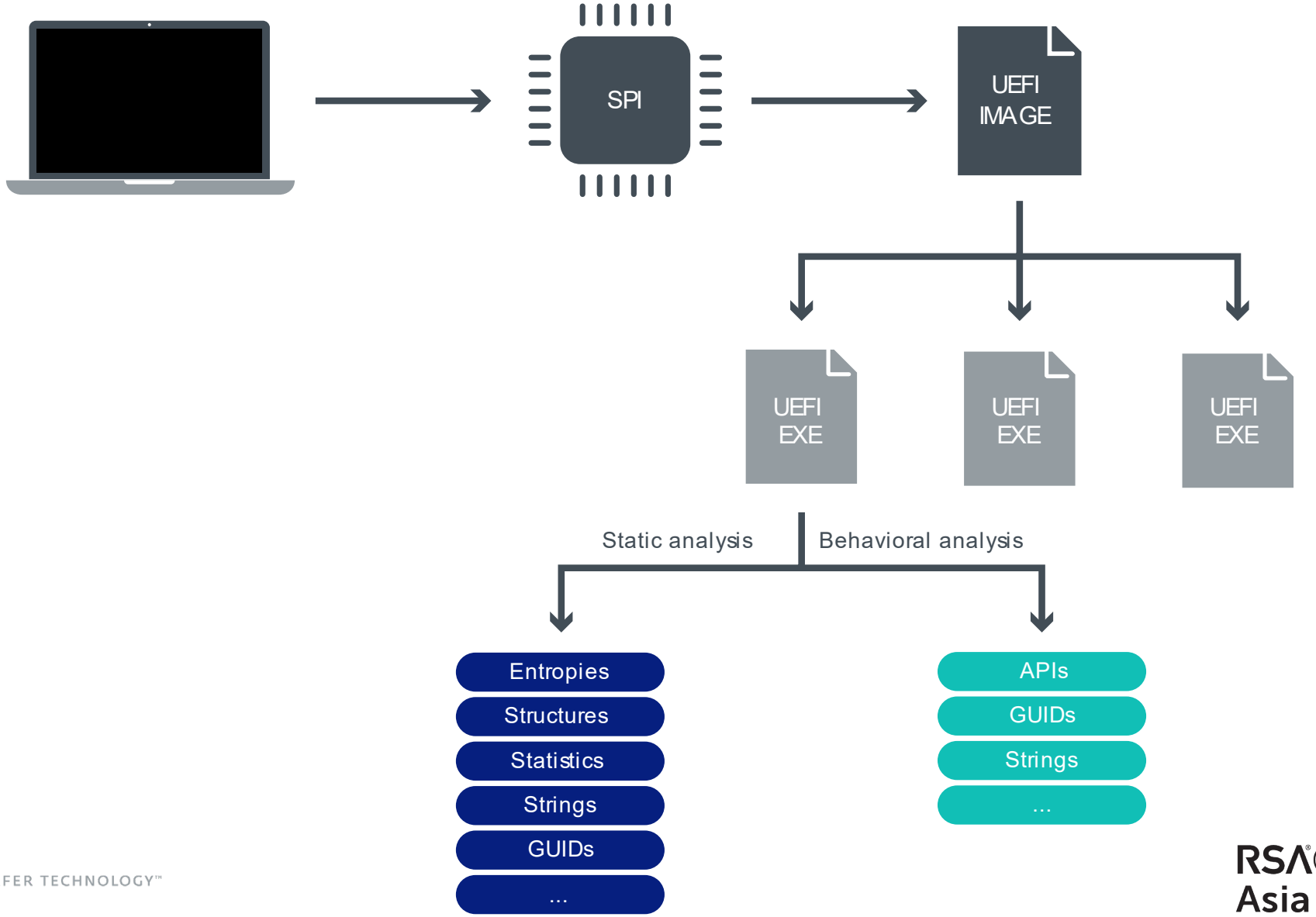


What data scientists spend the most time doing

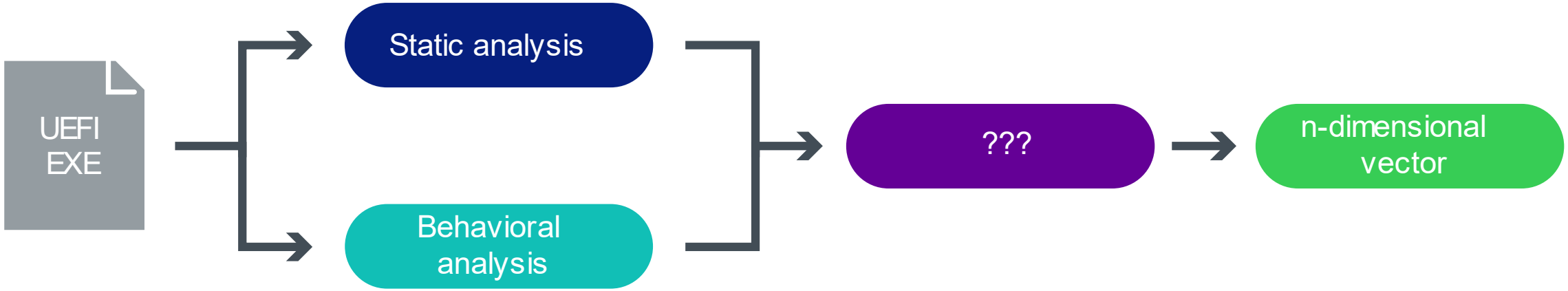
- Building training sets
- Cleaning and organizing data
- Collecting data sets
- Mining data for patterns
- Refining algorithms
- Other

Source: Forbes.com

Everything is about data



Everything is about data



Strings transformation

UEFI STRING DATASET

UEFI executable 1

```
BootCurrent
```

UEFI executable 2

```
SecureBoot
BootOrder
```

Term-occurrence matrix

	boo	oot	cur	...	rde	der	...
UEFI 1	1	1	1		0	0	
UEFI 2	2	2	0		1	1	
...							

LSA

Decomposed matrix

	V1	V2	...	Vn
UEFI 1	0.4	0.1		-0.3
UEFI 2	0.6	0.2		0.4
...				

Procedure statistics

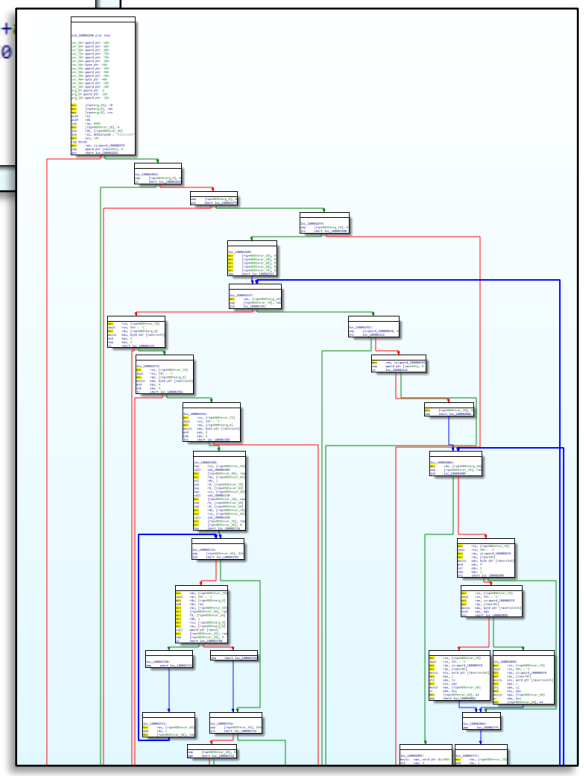
Functions window

Function name	Segment	Start	Length	Locals	Arguments	R	F	L	S	B	T	=
sub_1800002A0	.text	00000001800002A0	000001DD	000000C8	00000040	R	.	.	.	.	.	.
sub_180000480	.text	0000000180000480	000000E1	00000078	00000000	R	.	.	.	.	.	.
sub_180000570	.text	0000000180000570	00000071	00000038	00000010	R	.	.	.	.	.	.
sub_1800005F0	.text	00000001800005F0	0000002B	00000000	00000000	R	.	.	.	.	.	.
sub_180000620	.text	0000000180000620	00000052	00000018	00000011	R	.	.	.	.	.	.
sub_180000680	.text	0000000180000680	00000025	00000028	00000010	R	.	.	.	.	.	.
sub_1800006B0	.text	00000001800006B0	000000CF	00000058	00000014	R	.	.	.	.	.	.
sub_180000780	.text	0000000180000780	00000159	00000238	00000008	R	.	.	.	.	.	.
sub_1800008E0	.text	00000001800008E0	000000B4	00000048	00000010	R	.	.	.	.	.	.
sub_1800009A0	.text	00000001800009A0	000009D6	000000F8	00000010	R	.	.	.	.	.	.
sub_180001380	.text	0000000180001380	000000D3	00000038	00000010	R	.	.	.	.	.	.
sub_180001460	.text	0000000180001460	00000138	00000038	00000018	R	.	.	.	.	.	.
sub_1800015A0	.text	00000001800015A0	00000072	00000018	00000018	R	.	.	.	.	.	.
sub_180001620	.text	0000000180001620	000000F5	00000038	00000010	R	.	.	.	.	.	.
sub_180001720	.text	0000000180001720	00000091	00000000	00000010	R	.	.	.	.	.	.
sub_1800017C0	.text	00000001800017C0	00000054	00000028	00000010	R	.	.	.	.	.	.
sub_180001820	.text	0000000180001820	000000B6	00000038	00000010	R	.	.	.	.	.	.
sub_1800018E0	.text	00000001800018E0	000001A0	00000038	00000028	R	.	.	.	.	.	.
sub_180001A80	.text	0000000180001A80	00000113	00000038	00000019	R	.	.	.	.	.	.
sub_180001BA0	.text	0000000180001BA0	00000098	00000048	00000028	R	.	.	.	.	.	.
sub_180001C40	.text	0000000180001C40	00000098	00000048	00000028	R	.	.	.	.	.	.
sub_180001CE0	.text	0000000180001CE0	00000098	00000048	00000028	R	.	.	.	.	.	.
sub_180001D80	.text	0000000180001D80	000003AE	000000B8	00000018	R	.	.	.	.	.	.
sub_180002130	.text	0000000180002130	00000112	00000048	00000020	R	.	.	.	.	.	.
sub_180002250	.text	0000000180002250	00000031	00000000	00000020	R	.	.	.	.	.	.
sub_180002290	.text	0000000180002290	00000168	00000058	00000030	R	.	.	.	.	.	.
sub_180002400	.text	0000000180002400	00000061	00000058	00000000	R	.	.	.	.	.	.
sub_180002470	.text	0000000180002470	00000059	00000038	00000010	R	.	.	.	.	.	.

```
sub_1800063D8 proc near

arg_10= qword ptr  18h
arg_18= qword ptr  20h

mov     [rsp+arg_10], r8
mov     [rsp+arg_18], r9
sub     rsp, 28h
lea     r9, [rsp+28h+
call    sub_180005FC0
add     rsp, 28h
retn
sub_1800063D8 endp
```



Procedure statistics

Functions window

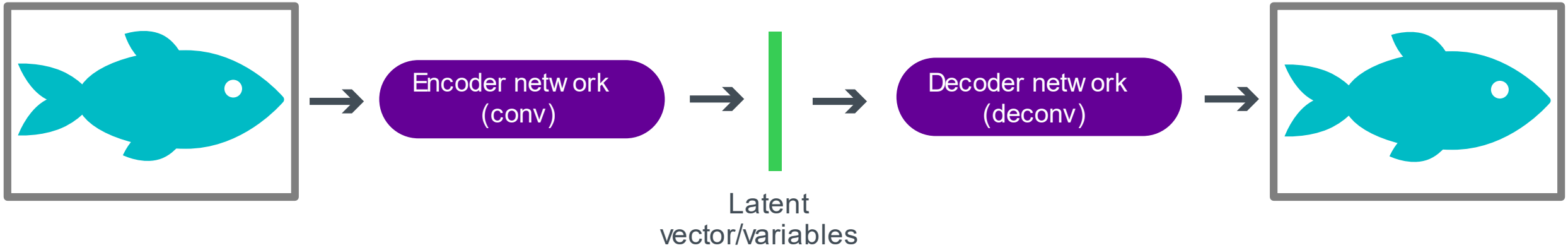
Function name	Segment	Start
sub_1800002A0	.text	00000001800002A0
sub_180000480	.text	0000000180000480
sub_180000570	.text	0000000180000570
sub_1800005F0	.text	00000001800005F0
sub_180000620	.text	0000000180000620
sub_180000680	.text	0000000180000680
sub_1800006B0	.text	00000001800006B0
sub_180000780	.text	0000000180000780
sub_1800008E0	.text	00000001800008E0
sub_1800009A0	.text	00000001800009A0
sub_180001380	.text	0000000180001380
sub_180001460	.text	0000000180001460
sub_1800015A0	.text	00000001800015A0
sub_180001620	.text	0000000180001620
sub_180001720	.text	0000000180001720
sub_1800017C0	.text	00000001800017C0
sub_180001820	.text	0000000180001820
sub_1800018E0	.text	00000001800018E0
sub_180001A80	.text	0000000180001A80
sub_180001BA0	.text	0000000180001BA0
sub_180001C40	.text	0000000180001C40
sub_180001CE0	.text	0000000180001CE0
sub_180001D80	.text	0000000180001D80
sub_180002130	.text	0000000180002130
sub_180002250	.text	0000000180002250
sub_180002290	.text	0000000180002290
sub_180002400	.text	0000000180002400
sub_180002470	.text	0000000180002470

→

	Instr. count	# bytes	...	...	...	...
Proc1	12	20				
Proc2	85	120				
Proc3	4	5				
Proc4	1412	1942				
...						
...						
...						
...						

Huge!

Procedure statistics



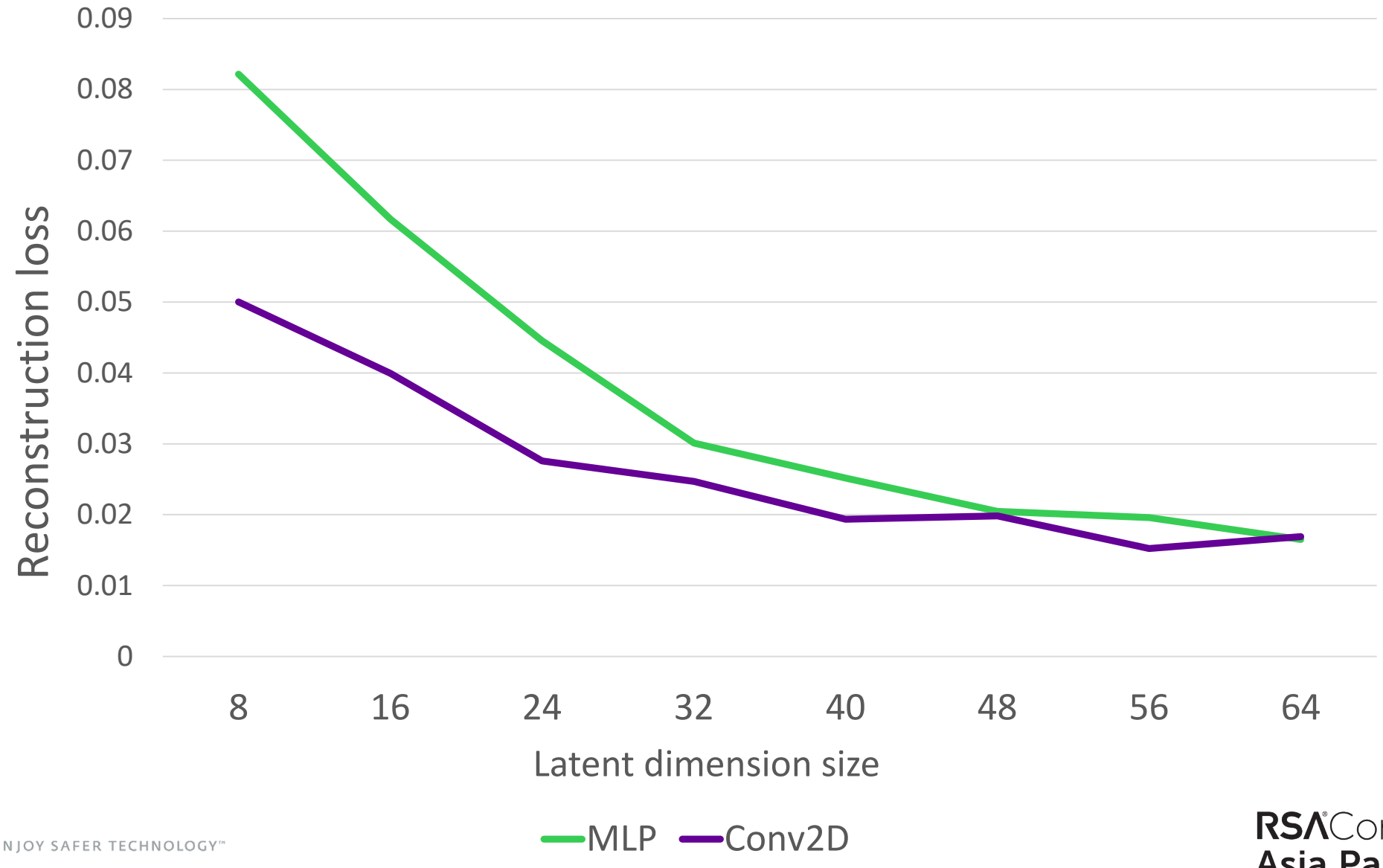
Procedure statistics

	Instr. count	# bytes
Proc1	12	20
Proc2	85	120
Proc3	4	5
Proc4	1412	1942

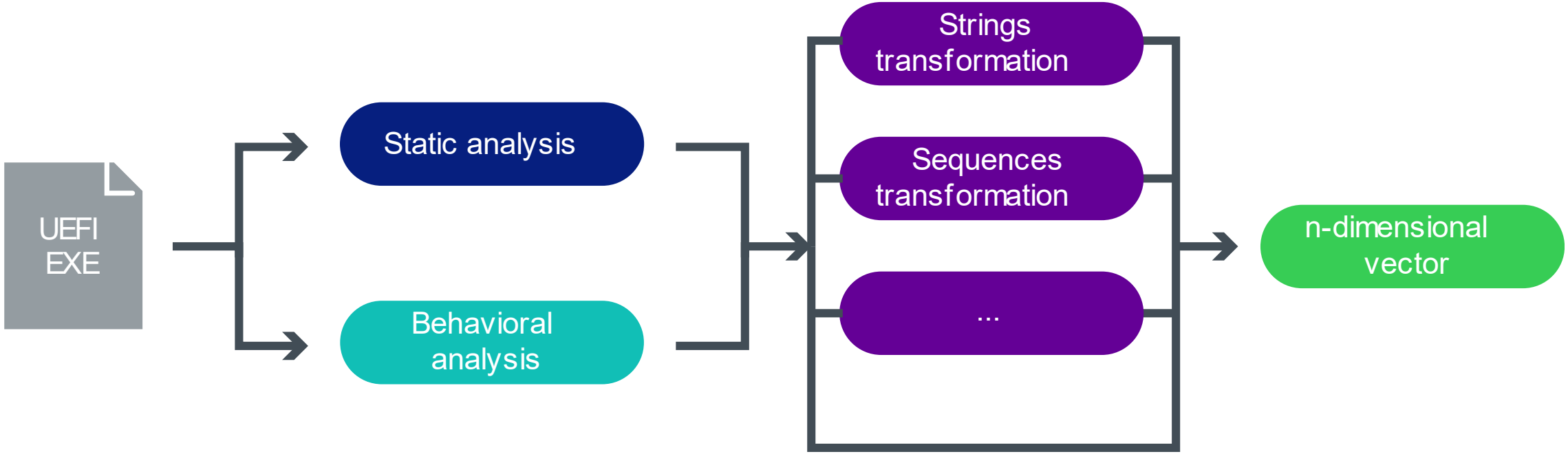
→ Encoder network (conv) → Latent vector/variables → Decoder network (deconv) →

	Instr. count	# bytes
Proc1	12	20
Proc2	85	120
Proc3	4	5
Proc4	1412	1942

Procedure statistics



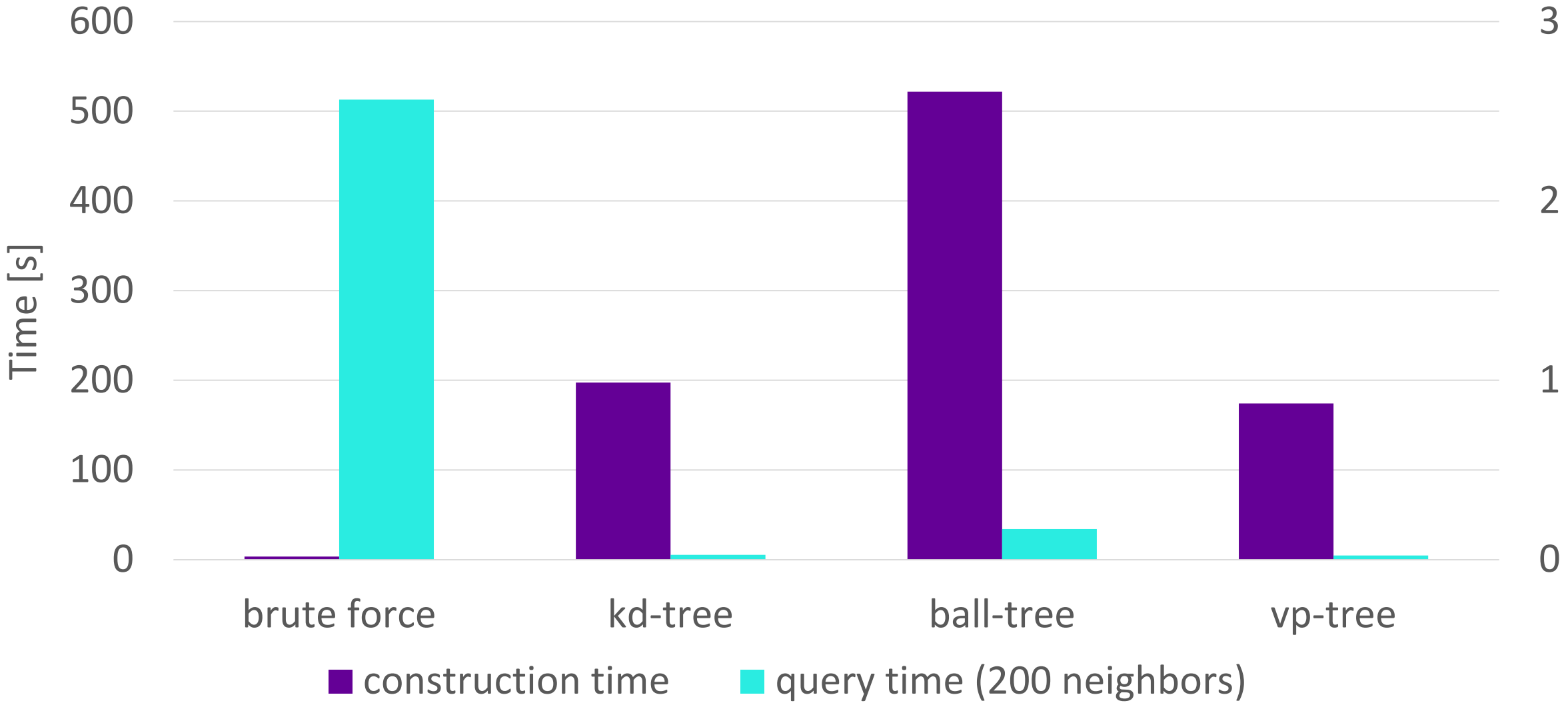
Final vector



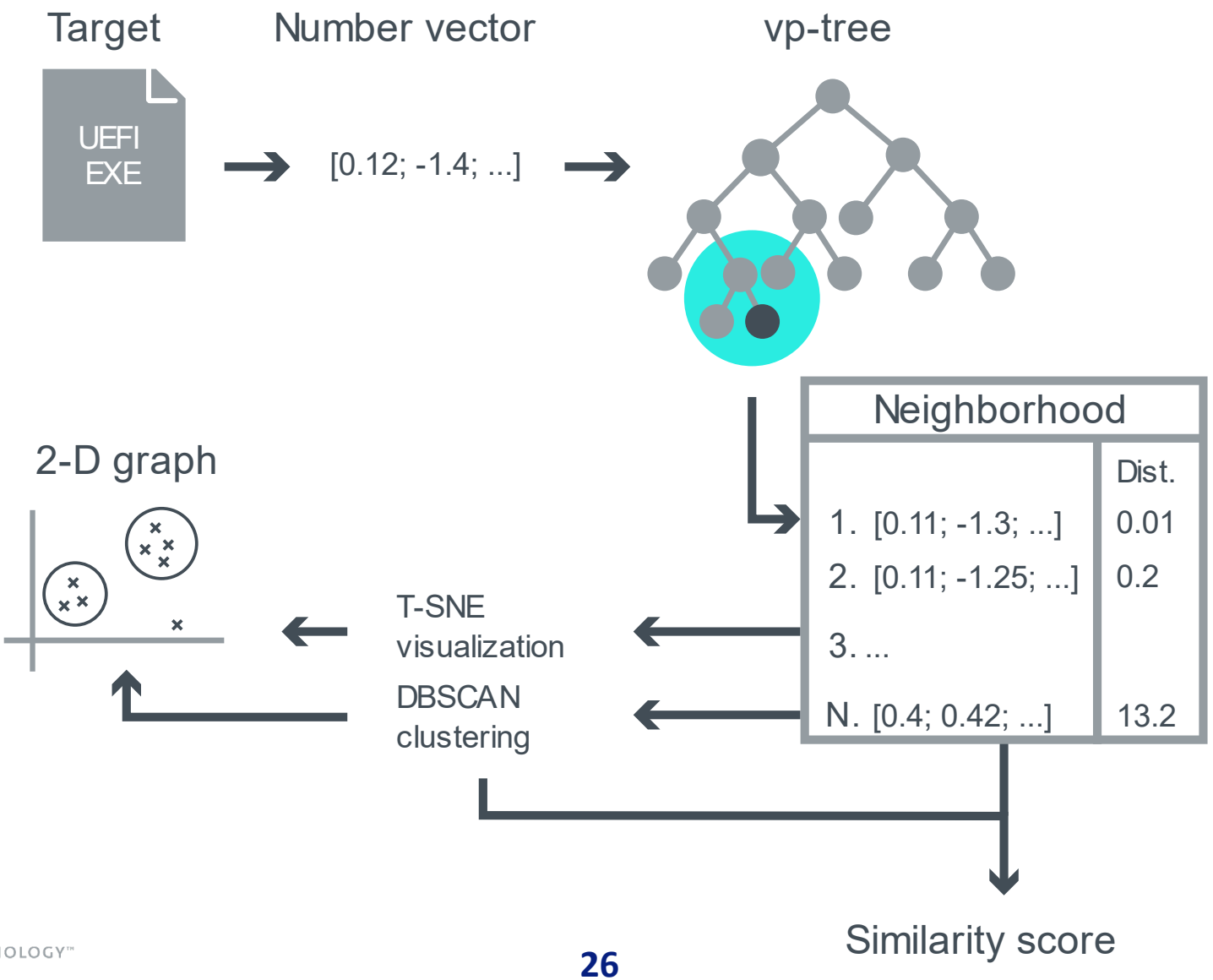
Trees to the rescue!

Image source: https://kajgana.com/sites/default/files/2013/05/30/88787/prirodni-pejzazhi-shto-nema-da-ve-ostavat-ramnodushni_96460.jpg

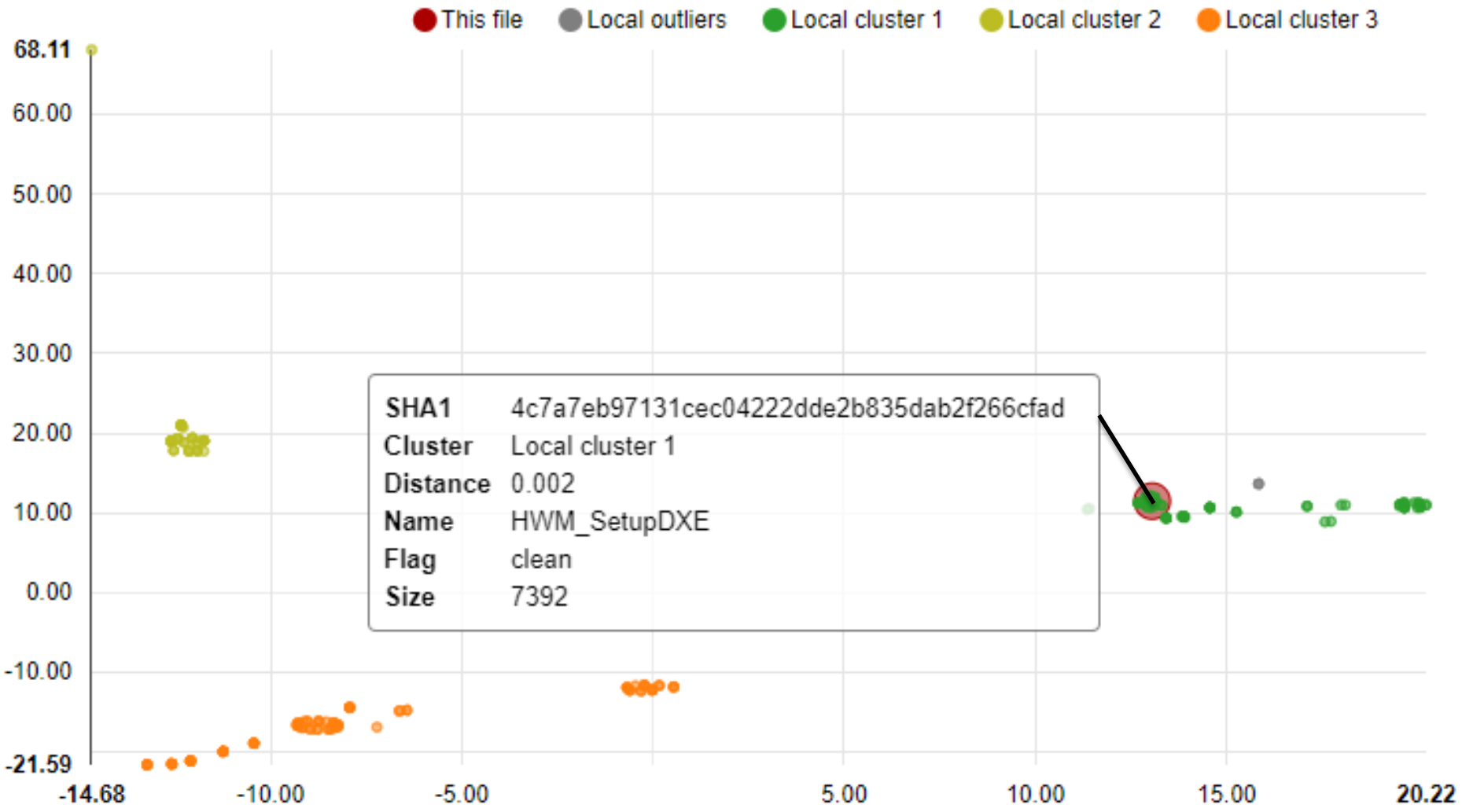
Trees to the rescue!



Processing pipeline

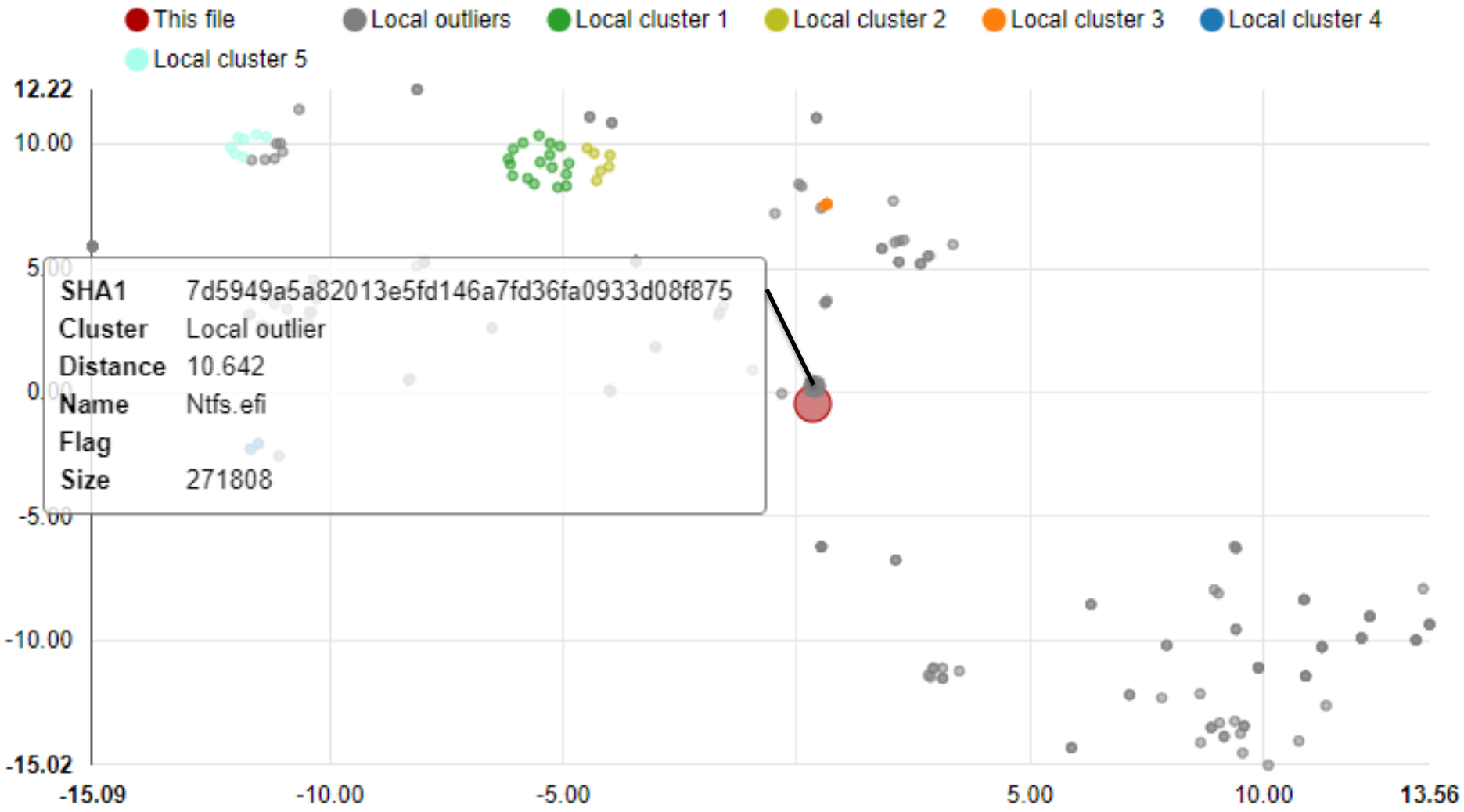


Visualization



Visualization

Visualization



Results in real-time

SHA1	Filename	Size	Flag	Similarity	Time added	
a698de949b9cc93039	ReFlash	8.0 kB	clean		(a few seconds ago)	quick info
abda03a1f64cf995fa	LPC47N207SioDxe.efi	40.9 kB		1	(a minute ago)	quick info
2e796de45feab318d5	WdtDxe.efi	3.1 kB	clean		(a minute ago)	quick info
4dc8a66b6ea13cdd64	OEMDXE	4.7 kB	clean		(2 minutes ago)	quick info
faf4959cf01c0f5b60	SMBIOSUpdateData	1.9 kB	clean		(2 minutes ago)	quick info
4f8cccc72a32e53fdd	SecFlashUpdDxe	4.2 kB	clean		(3 minutes ago)	quick info
ea24cdbc622061fa9f	IntelXtuDxe	2.7 kB	clean		(4 minutes ago)	quick info
3bd29c6bd9d92482ae	MX25L3205AFlashPartDxe.efi	4.2 kB	clean		(4 minutes ago)	quick info
e879b75c41cfc41a06	SBSMI	14.1 kB	clean		(5 minutes ago)	quick info
9e914f895e2a8f06f2	AhciSmm	10.5 kB	clean		(5 minutes ago)	quick info
acc1cc4ccd0f11054e	NetworkStackSetupScreen	3.2 kB	clean		(5 minutes ago)	quick info
9315a4358d2d9075c3	AlertStandardFormatSmm.efi	6.0 kB			(6 minutes ago)	quick info
aa08135cb967664200	AcpiFvi	3.0 kB	clean		(6 minutes ago)	quick info
3fa490b9aee111712b	SmmOemActivation	5.5 kB	clean		(7 minutes ago)	quick info
363f4ec8446791c041	SwitchableGraphicsDxe.efi	6.7 kB	clean		(7 minutes ago)	quick info
df438fe18170093d7e	GD25Q32FlashPartDxe.efi	5.0 kB	clean		(7 minutes ago)	quick info
255032fab37ecaf2f1	PchSpiWrap	3.7 kB		73	(8 minutes ago)	quick info
d6eb71f446a48a6b33	SleepSmi	12.1 kB	clean		(8 minutes ago)	quick info
319c5d82ec3e95281c	AcpiModeEnable	18.1 kB	clean		(9 minutes ago)	quick info
07f19383b23597fd84	6e5228f3-933e-4961-9573-0f1e61b522ac (LENOVO_SMBIOS_VPRO_GUID)	3.5 kB			(9 minutes ago)	quick info
d9af1cd59e2512c875	MeFwDowngrade.efi	4.1 kB			(9 minutes ago)	quick info
06ef2c272cab0dded2	AmiTcgPlatformDxe	34.9 kB	clean	12	(10 minutes ago)	quick info
5973d74fbd06a21c2f	SetTimerPeriodDxe	1.3 kB	clean		(10 minutes ago)	quick info

RSAConference2019
Asia Pacific & Japan

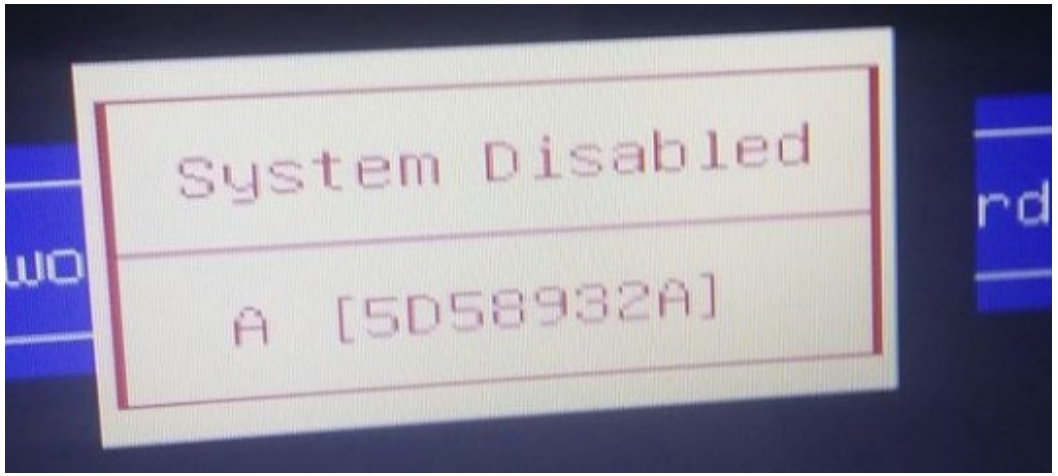
Case studies

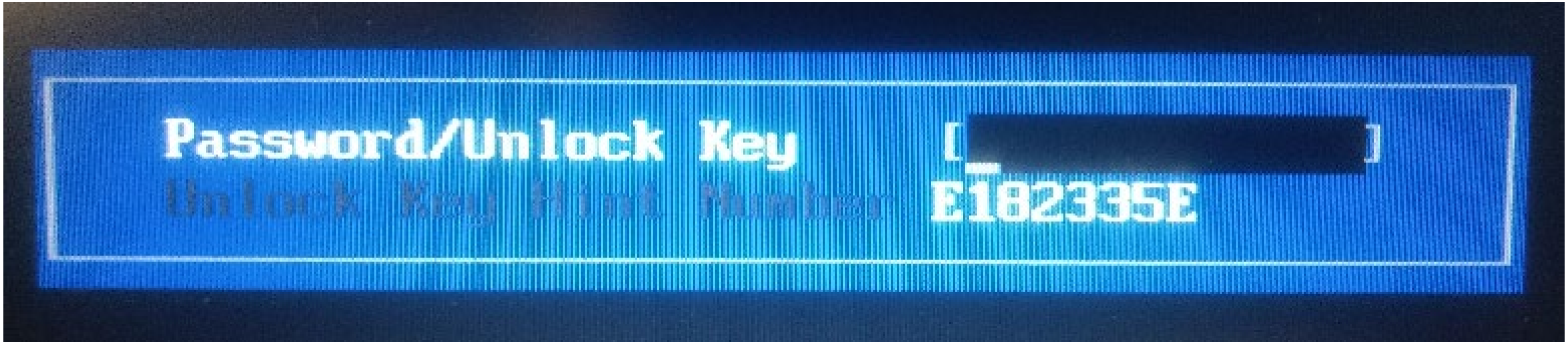
RSAConference2019
Asia Pacific & Japan

UEFI Backdoors & OS Persistence

UEFI Backdoors

- Prevalent recovery mechanisms
- Usually can be triggered by a key combination or too many wrong attempts



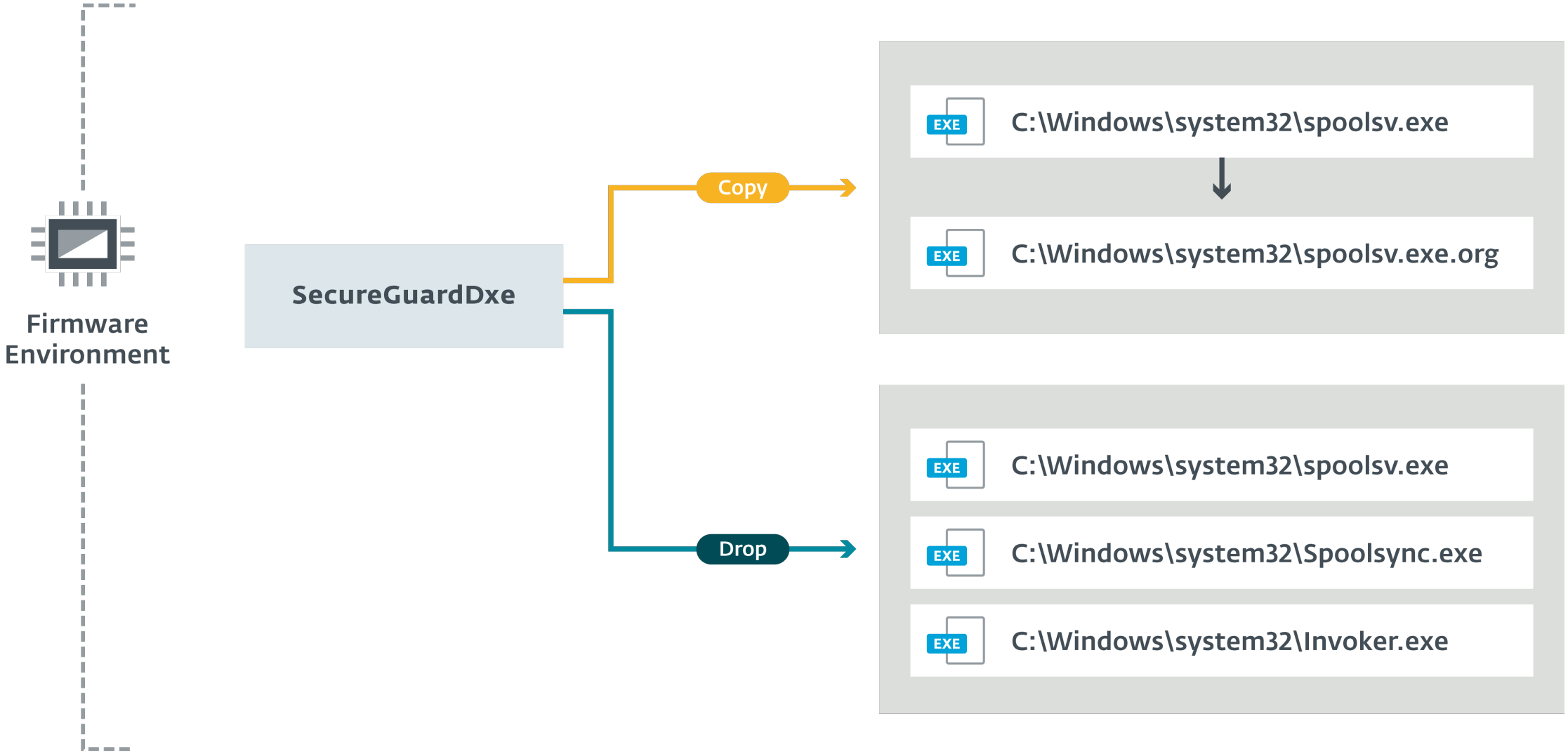


UEFI Backdoors

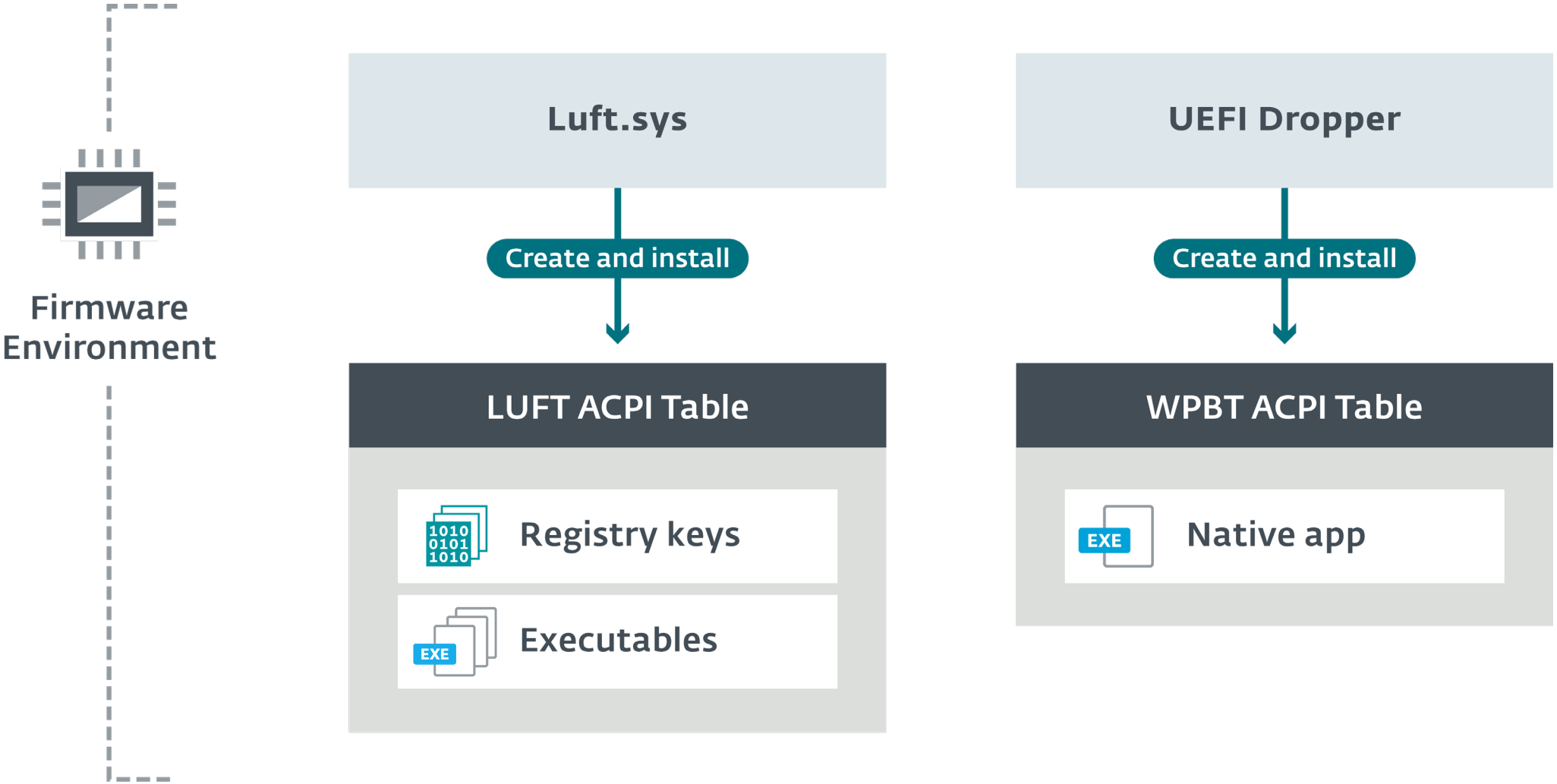
- Many password generators exist for different OEMs firmware

```
Key[0] = 0xB9u;
Key[1] = 0xEDu;
Key[2] = 0xF5u;
Key[3] = 0x69;
Key[4] = 0x9Du;
Key[5] = 0x16;
Key[6] = 0x49;
Key[7] = 0xF9u;
Key[8] = 0x8Cu;
Key[9] = 0x5F;
Key[10] = 0x7C;
Key[11] = 0xB3u;
Key[12] = 0x68;
Key[13] = 0x3C;
Key[14] = 0xD4u;
Key[15] = 0xA7u;
if ( (gRT->GetVariable(L"BackDoor", &VendorGuid, 0i64, &Size, &BackDoor) & 0x8000000000000000ui64) == 0i64 )
{
  StrToDword(&InputDword, UserInput, 0);
  BackDoorChecksum = CalcChecksum(Key, &BackDoor);
  if ( InputDword == BackDoorChecksum )
  {
    v16 = 3i64;
    ResetAMITSE();
  }
}
```

OS Persistence – The Cowboy way



OS Persistence - WPBT



Security Implications – UEFI backdoors

- Physical access can bypass the password set by user
- Settings can be changed
  - SecureBoot
  - Lock device boot order
  - Chassis intrusion
  - etc
- Creates a false sense of security

Security Implications – OS Persistence

- Is this level of persistence really needed?
- System firmware is not updated regularly – vulnerabilities in application can linger for a long time (or forever)

RSAConference2019
Asia Pacific & Japan

Where is the real malware?

RSA Conference 2019
Asia Pacific & Japan

LOJAX

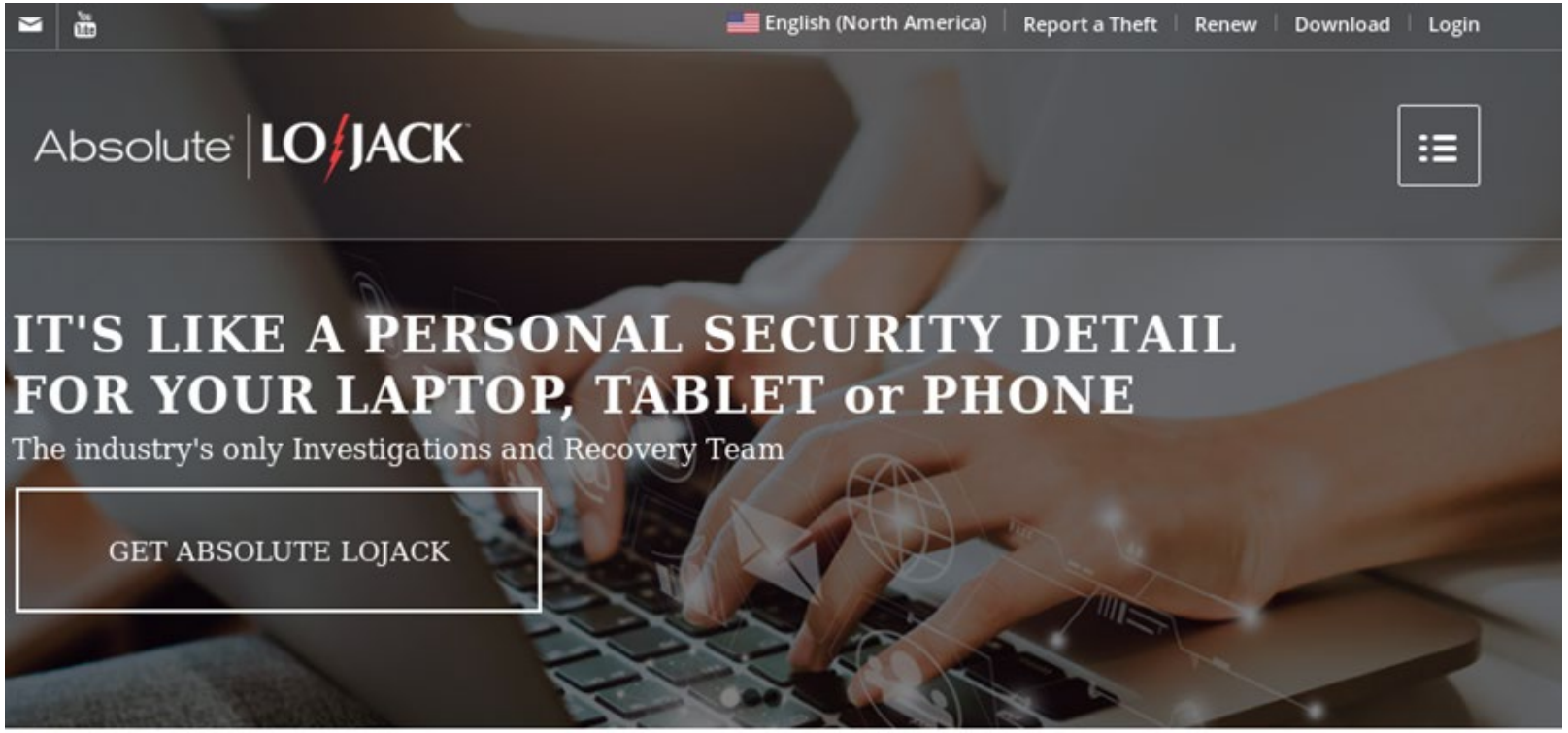
LOJACK

Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

- Espionage group active since the early 2000s
- Tens of custom-built malicious tools
- Regular usage of 0days
- Preferred targets: geopolitics actors, government employees, activists, journalists)

Absolute Software



THE LEADER IN DATA AND DEVICE PROTECTION

Absolute LoJack is the only persistent security solution that can track and recover stolen devices, while providing features that protect your personal information.

ABSOLUTE® | HOME & OFFICE

Built into the BIOS or firmware during the manufacturing process of most major device manufacturers, we are able to provide our customers with the only security solution that can withstand a factory reset, installation of a new OS or even a complete hard drive replacement.

Locate Lock Delete Recover

LoJack Architecture

UEFI/BIOS module executes	Windows early boot	Windows OS running	
1	2	3	4
UEFI/BIOS module	`autochk.exe`	`rpcnetp.exe` - small agent	Normal operation
Contains persistent agent and its dropper	Drops `rpcnetp.exe` - small agent	Injects its DLL into svchost.exe, then Internet Explorer	Full recovery agent is running on the machine
Replaces legitimate `autochk.exe`	Installs it as a service	Communicates with C&C server to download and install full recovery agent	

LoJack Architecture

UEFI/BIOS module executes	Windows early boot	Windows OS running	
1 UEFI/BIOS module	**2** `autochk.exe`	**3** `rpcnetp.exe` - small agent	**4** Normal operation
Contains persistent agent and its dropper	Drops `rpcnetp.exe` - small agent	Injects its DLL into svchost.exe, then Internet Explorer	Full recovery agent is running on the machine
Replaces legitimate `autochk.exe`	Installs it as a service	Communicates with C&C server to download and install full recovery agent	

LoJack Architecture

UEFI/BIOS module executes	Windows early boot	Windows OS running	
1 UEFI/BIOS module	**2** `autochk.exe`	**3** `rpcnetp.exe` - small agent	**4** Normal operation
Contains persistent agent and its dropper	Drops `rpcnetp.exe` - small agent	Injects its DLL into svchost.exe, then Internet Explorer	Full recovery agent is running on the machine
Replaces legitimate `autochk.exe`	Installs it as a service	Communicates with C&C server to download and install full recovery agent	

Black Hat USA 2009

- Exposed design vulnerabilities in agent

Deactivate the Rootkit: Attacks on BIOS anti-theft technologies

Alfredo Ortega, Anibal Sacco, Core Security Technologies

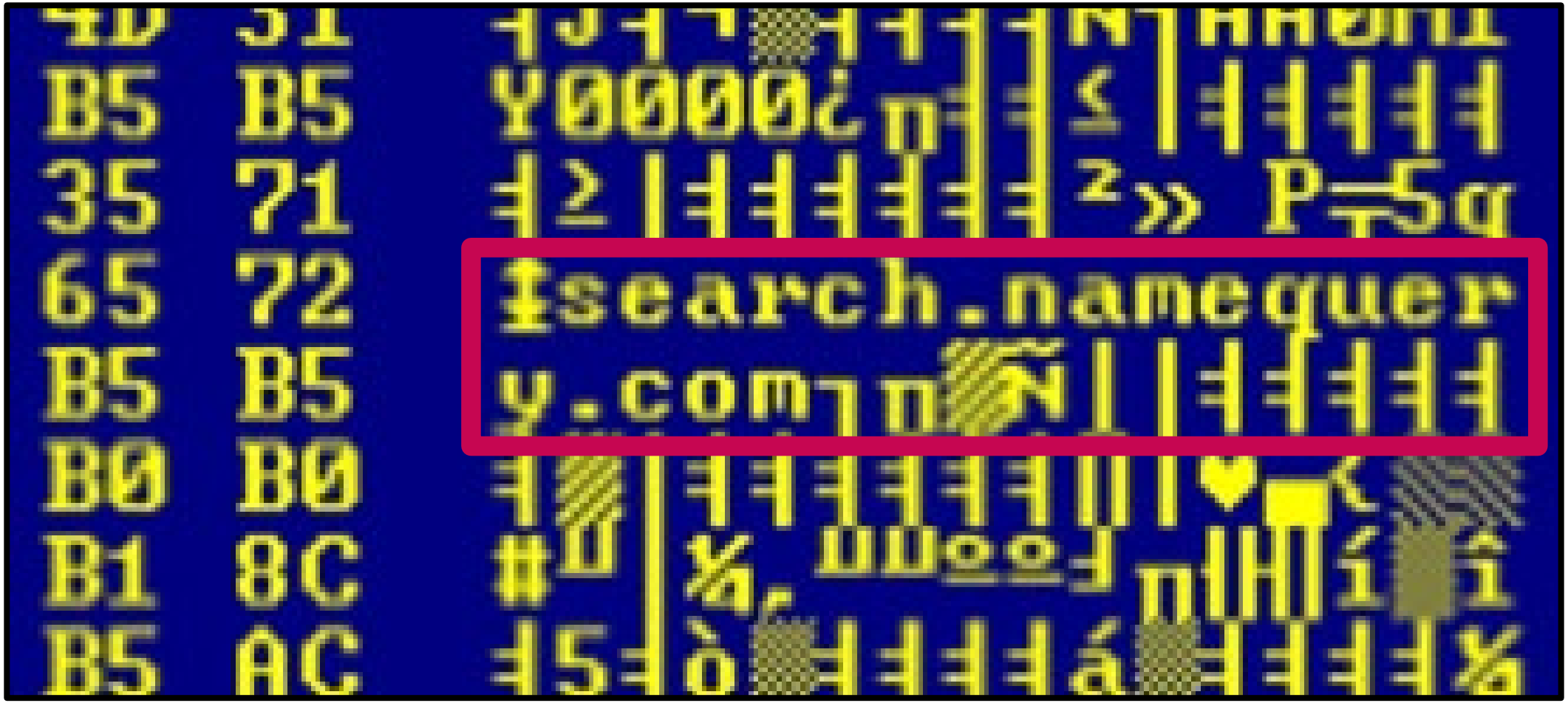
July 24, 2009

Configuration file vulnerability

```
00003C20:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
00003C30:  00 00 00 00-00 00 00 00-B1 B7 B5 B5-35 AB B1 B4
00003C40:  B5 F5 B5 AA-B1 B5 B5 B5-B5 A5 BF 41-41 30 4D 31
00003C50:  59 30 30 30-30 A8 B7 B5-B5 F3 B3 B5-B5 B5 B5 B5
00003C60:  B5 F2 B3 B5-B5 B5 B5 B5-B5 FD AF 00-50 D1 35 71
00003C70:  17 73 65 61-72 63 68 2E-6E 61 6D 65-71 75 65 72
00003C80:  79 2E 63 6F-6D BF B7 B2-A5 B3 B3 B5-B5 B5 B5 B5
00003C90:  B5 B2 B3 B5-B5 B5 B5 B5-B5 BA B3 03-DC 7B B0 B0
00003CA0:  23 BD B3 AC-2C BD BD A7-A7 BE B7 D7-B6 A1 B1 8C
00003CB0:  B5 35 B5 95-B1 B5 B5 B5-B5 A0 B1 B5-B5 B5 B5 AC
00003CC0:  AE B5 B5 B5-B5 B5 B5 B5-B5 B5 B5 B5-B5 B5 B5 B5
00003CD0:  B5 B5 B5 B5-B5 B5 B5 B5-B5 B5 B5 B5-AF B4 B5 AE
00003CE0:  B3 B5 B5 B5-B5 B5 B5 98-B4 0D 98 B4-0D 86 B4 0D
00003CF0:  9E B1 41 54-44 54 9D B6-B5 B5 B5 B4-8D B4 54 58
00003D00:  34 0D 86 B4-0D 9E B1 41-54 44 54 9D-B6 B5 B5 B5
00003D10:  B4 96 B4 00-00 00 00 00-00 00 00 00-00 00 00 00
00003D20:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

```
search.namequer
y.com
```

Configuration file vulnerability



RSAConference2019
Asia Pacific & Japan

LOJAX

Win32/LoJax – The cat is out of the bag

Lojack Becomes a Double-Agent

A ASERT team on May 1, 2018.

- Small agent config modifications
- Domains previously used as SedUploader C2
- SedUploader = Sednit

RSAConference2019
Asia Pacific & Japan

The hunt begins

THE HUNT

- Several military and diplomatic organizations in the Balkans, Central and Eastern Europe hit
- Sednit tools present

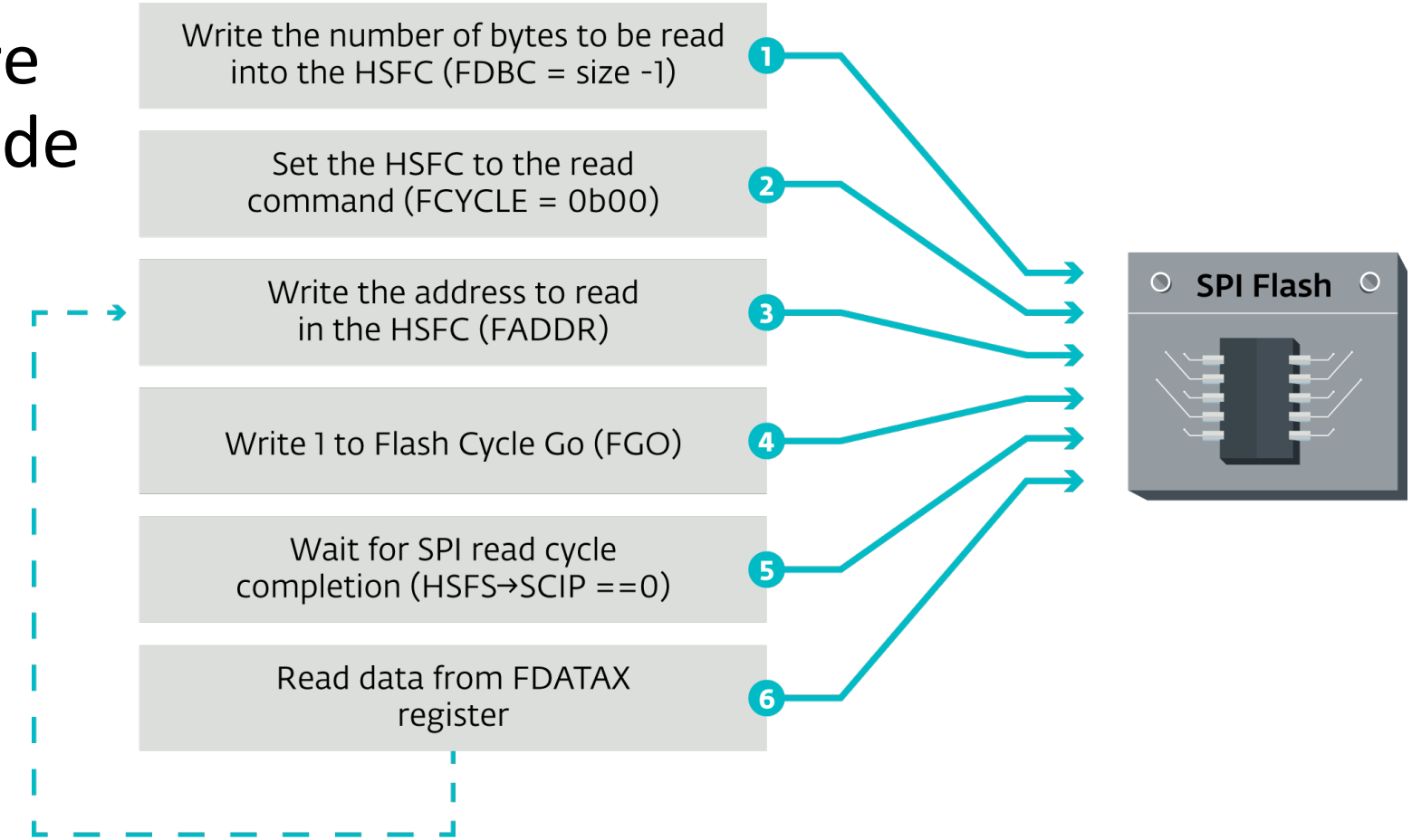
1) info_efi.exe

- Custom tool found alongside LoJax
- Extract hardware information, including UEFI firmware vendor and version

```
Get SMBIOS..
SMBIOS:
^@^X^@^@^A^BRê^C^@■Ð    |^@^@^@^@■^G^D^F^@^@Phoenix Technologies LTD^@6.00^@05/19/2017^@^@^A^[^A^@^A^B^C
^DVM^S■ÍÛW^MÓ\■^T~-(f^F^@^@VMware, Inc.^@VMware Virtual Platform^@None^@VMware-[redacted]
[redacted]^@^@^B^O^B^@^A^B^C^D^@^@^@^@^@^A^@Intel Corporation^@440BX Desktop Reference Platf
orm^@None^@None^@^@^C^U^C^@^A^A^B^C^D^C^C^C^C4^R^@^@^@^@^@^@No Enclosure^@N/A^@None^@No Asset Tag^@^@^D*
^D^@^A^C^B^Bé^F ^@ÿû■^O^C^B^@^@0u¸^KA^D^U^@^V^@ÿÿ^@^@^@^A^A^@$^@^B^@CPU #000^@GenuineIntel^@Intel(R) Cor
e(TM) i5-7400 CPU @ 3.00GHz^@^@^E.^E^@^C^D^C^C^O^L^@^X^E^B^O^F^@^G^@^H^@        ^@
^@^K^@^L^@^M^@^N^@^O^@^P^@^Q^@^R^@^S^@^T^@^D^@^@^F^L^F^@^Aÿ^@^P^A
```

2) ReWriter_read.exe

- Tool to dump firmware content found alongside LoJax sample



3) ReWriter_Binary.exe

- Contains a UEFI rootkit: EFI/LoJax
- Infect the firmware image dumped by ReWriter_read.exe
- Write the trojanized image back to the target

4) EFI/LoJax

- Installs an NTFS driver

```
  ControllerHandle_1 = ControllerHandle;
  EfiDriverBindingProtocol = This;
  LockedByMe = 0;
  if ( fnNtfsAcquireLockOrFail() >= 0 )
    LockedByMe = 1;
  Status = fnInitializeUnicodeCollationSupport(EfiDriverBindingProtocol->DriverBindingHandle);
  if ( Status >= 0 )
  {
    v4 = EFI_OPEN_PROTOCOL_GET_PROTOCOL;
    Status = (gEfiBootServices->OpenProtocol)(
               ControllerHandle_1,
               &gEfiBlockIoProtocolGuid,
               &EfiBlockIoProtocol,
               EfiDriverBindingProtocol->DriverBindingHandle,
               ControllerHandle_1,
               v4);
    if ( Status >= 0 )
    {
      LODWORD(v6) = EFI_OPEN_PROTOCOL_BY_DRIVER;
      Status = (gEfiBootServices->OpenProtocol)(
                 ControllerHandle_1,
                 &gEfiDiskIoProtocolGuid,
                 &EfiDiskIoProtocol,
                 EfiDriverBindingProtocol->DriverBindingHandle,
                 ControllerHandle_1,
                 v6);
      if ( Status >= 0 )
      {
        Status = fnNtfsAllocateVolume(ControllerHandle_1,
            EfiDiskIoProtocol, EfiBlockIoProtocol);
        if ( Status < 0 )
        {
          LODWORD(v7) = 4;
          Status = (gEfiBootServices->OpenProtocol)(
                     ControllerHandle_1,
                     &gEfiSimpleFileSystemProtocolGuid,
                     0i64,
                     EfiDriverBindingProtocol->DriverBindingHandle,
                     ControllerHandle_1,
                     v7);
          if ( Status < 0 )
            (gEfiBootServices->CloseProtocol)(
              ControllerHandle_1,
              &gEfiDiskIoProtocolGuid,
              EfiDriverBindingProtocol->DriverBindingHandle,
              ControllerHandle_1);
        }
      }
    }
  }
  if ( LockedByMe )
    fnNtfsReleaseLock(v3);
  return Status;
}
```

```
  Status = NtfsAcquireLockOrFail ();
  if (!EFI_ERROR (Status)) {
    LockedByMe = TRUE;
  }

  Status = InitializeUnicodeCollationSupport (This->DriverBindingHandle);
  if (EFI_ERROR (Status)) {
    goto Exit;
  }
  Status = gBS->OpenProtocol (
                 ControllerHandle,
                 &gEfiBlockIoProtocolGuid,
                 (VOID **) &BlockIo,
                 This->DriverBindingHandle,
                 ControllerHandle,
                 EFI_OPEN_PROTOCOL_GET_PROTOCOL
                 );
  if (EFI_ERROR (Status)) {
    goto Exit;
  }
  Status = gBS->OpenProtocol (
                 ControllerHandle,
                 &gEfiDiskIoProtocolGuid,
                 (VOID **) &DiskIo,
                 This->DriverBindingHandle,
                 ControllerHandle,
                 EFI_OPEN_PROTOCOL_BY_DRIVER
                 );
  if (EFI_ERROR (Status)) {
    goto Exit;
  }
  Status = NtfsAllocateVolume (ControllerHandle, DiskIo, BlockIo);
  if (EFI_ERROR (Status)) {
    Status = gBS->OpenProtocol (
                   ControllerHandle,
                   &gEfiSimpleFileSystemProtocolGuid,
                   NULL,
                   This->DriverBindingHandle,
                   ControllerHandle,
                   EFI_OPEN_PROTOCOL_TEST_PROTOCOL
                   );
    if (EFI_ERROR (Status)) {
      gBS->CloseProtocol (
             ControllerHandle,
             &gEfiDiskIoProtocolGuid,
             This->DriverBindingHandle,
             ControllerHandle
             );
    }
  }

Exit:
  if (LockedByMe) {
    NtfsReleaseLock ();
  }
```

4) EFI/LoJax

- Writes Win32/LoJax binaries on the system partition

4) EFI/LoJax

- Writes Win32/LoJax binaries on the system partition
- Patch a value in the Windows Registry
- ..
- Profit!

LoJack Architecture

UEFI/BIOS module executes	Windows early boot	Windows OS running	
1 UEFI/BIOS module	**2** `autochk.exe`	**3** `rpcnetp.exe` - small agent	**4** Normal operation
Contains persistent agent and its dropper	Drops `rpcnetp.exe` - small agent	Injects its DLL into svchost.exe, then Internet Explorer	Full recovery agent is running on the machine
Replaces legitimate `autochk.exe`	Installs it as a service	Communicates with C&C server to download and install full recovery agent	

Let's take a step back

- Info_efi.exe
- collect details on firmware
- ReWriter_read.exe
- dump firmware memory
- ReWriter_Binary.exe
- Infect dumped memory with a custom UEFI module
- Write the image back

RSAConference2019
Asia Pacific & Japan

First UEFI rootkit in the wild?

Vault 7: CIA Hacking Tools Revealed

DerStarke 2.0

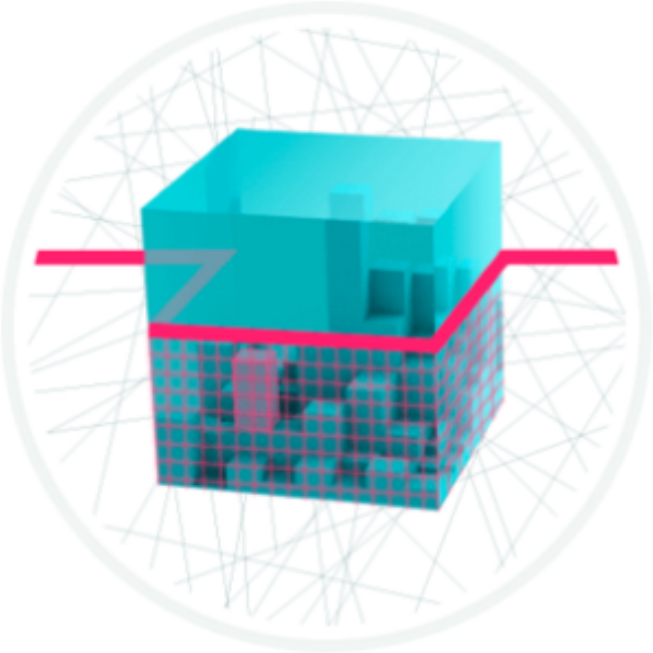
('toc' missing)

Building DerStarke (Developer)
Top-Level Builder (build.py)

Hacking Team Uses UEFI BIOS Rootkit to Keep RCS 9 Agent in Target Systems

Posted on: July 13, 2015 at 10:13 am **Posted in:** Malware, Targeted Attacks

Author: Philippe Lin (Senior Threat Researcher)



UEFI Scanner

ESET is the first internet security provider to add a dedicated layer into its solution that protects the Unified Extensible Firmware Interface (UEFI). ESET UEFI Scanner checks and enforces the security of the pre-boot environment that is compliant with the UEFI specification. It is designed to monitor the integrity of the firmware and in case modification is detected, it notifies the user.

Show more

RSAConference2019
Asia Pacific & Japan

Prevention and Remediation

Prevention

- Keep your firmware up-to-date

Prevention

- Keep your firmware up-to-date
- Verify equipment's UEFI security

Prevention

- Keep your firmware up-to-date
- Verify equipment's UEFI security
- Firmware security assessments can be done with CHIPSEC
- Note that chipsec is a test tool and not intended for use on production systems.

Prevention

- Keep your firmware up-to-date
- Verify equipment's UEFI security
- Firmware security assessments can be done with CHIPSEC
- Note that chipsec is a test tool and not intended for use on production systems.
- Security solution that scans UEFI firmware memory

Remediation

- Reinstall Windows

Remediation

- ~~Reinstall Windows~~

Remediation

- ~~Reinstall Windows~~
- Replace harddrive

Remediation

- ~~Reinstall Windows~~
- ~~Replace harddrive~~

Remediation

- ~~Reinstall Windows~~
- ~~Replace harddrive~~
- Reflash firmware with a clean version from the vendor

Remediation

- ~~Reinstall Windows~~
- ~~Replace harddrive~~
- Reflash firmware with a clean version from the vendor
- If it’s not an option…

Remediation

- ~~Reinstall Windows~~
- ~~Replace harddrive~~
- Reflash firmware with a clean version from the vendor
- If it’s not an option…

Conclusion

- UEFI malware is no longer theoretical.
- It has to be present in current threat models.
- Machine learning is a usable method to find oddities in UEFI landscape

RSAConference2019
Asia Pacific & Japan

Questions?

Big shout-out to: Hamidreza Ebtehaj
Martin Smolár
Frédéric Vachon