



ENJOY SAFER TECHNOLOGY™

# Malware Buried Deep Down the SPI Flash: Sednit's First UEFI Rootkit Found in the Wild

Jean-Ian Boutin | Senior Malware Researcher

Frédéric Vachon | Malware Researcher





Jean-Ian Boutin

Senior Malware Researcher



@jiboutin



Frédéric Vachon

Malware Researcher



@Freddrckk\_

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

- Espionage group active since the early 2000s
- Very visible in the past few years as allegedly behind these notorious hacks

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

- Espionage group active since the early 2000s
- Very visible in the past few years as allegedly behind these notorious hacks
  - Democratic National Committee (DNC)

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

- Espionage group active since the early 2000s
- Very visible in the past few years as allegedly behind these notorious hacks
  - Democratic National Committee (DNC)
  - World Anti-Doping Agency (WADA)

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

- Espionage group active since the early 2000s
- Very visible in the past few years as allegedly behind these notorious hacks
  - Democratic National Committee (DNC)
  - World Anti-Doping Agency (WADA)
  - TV5 Monde
  - etc

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF COLUMBIA

UNITED STATES OF AMERICA

v.

VIKTOR BORISOVICH NETYKSHO,  
BORIS ALEKSEYEVICH ANTONOV,  
DMITRIY SERGEYEVICH BADIN,  
IVAN SERGEYEVICH YERMAKOV,  
ALEKSEY VIKTOROVICH  
LUKASHEV,  
SERGEY ALEKSANDROVICH  
MORGACHEV,  
NIKOLAY YURYEVIKH KOZACHEK,  
PAVEL VYACHESLAVOVICH  
YERSHOV,  
ARTEM ANDREYEVICH  
MALYSHEV,  
ALEKSANDR VLADIMIROVICH  
OSADCHUK,  
ALEKSEY ALEKSANDROVICH  
POTEMKIN, and  
ANATOLIY SERGEYEVICH  
KOVALEV,

Defendants.

CRIMINAL NO.

(18 U.S.C. §§ 2, 371, 1030, 1028A, 1956,  
and 3551 et seq.)

**RECEIVED**

**JUL 13 2018**

*Clerk, U.S. District & Bankruptcy  
Courts for the District of Columbia*

\*\*\*\*\*

**INDICTMENT**

- Espio
- Very
- behin
- De
- Wo
- TV5
- etc

Os  
ly

# Sednit

(AKA Fancy Bear/APT28/STRONTIUM/etc)

UNITED STATES DISTRICT COURT  
WESTERN DISTRICT OF PENNSYLVANIA

UNITED STATES OF AMERICA

v.

Criminal No.

18-263

ALEKSEI SERGEYEVICH MORENETS  
EVGENII MIKHAYLOVICH SEREBRIAKOV  
IVAN SERGEYEVICH YERMAKOV  
ARTEM ANDREYEVICH MALYSHEV  
DMITRIY SERGEYEVICH BADIN  
OLEG MIKHAYLOVICH SOTNIKOV  
ALEXEY VALEREVICH MININ

Defendants.

18 U.S.C. §§ 371, 1030(a)(2)(C),  
1030(a)(5)(A)  
(Conspiracy)  
18 U.S.C. § 1349 and § 3559(g)(1)  
(Conspiracy to Commit Wire Fraud)  
18 U.S.C. § 1343 (Wire Fraud)  
18 U.S.C. § 1028A  
(Aggravated Identity Theft)  
18 U.S.C. § 1956(h)  
(Conspiracy to Launder Money)

[UNDER SEAL]

INDICTMENT

**FILED**

OCT 03 2018

CLERK U.S. DISTRICT COURT  
WEST. DIST. OF PENNSYLVANIA



# Example of phishing email

From [REDACTED]  
Subject **Privacy violation**  
To [REDACTED]

Dear Customer,

Please be informed that your personal data has been found on Google Drive Service. For your privacy purposes we have temporary restricted access to the following page:

<https://docs.google.com/document/d/0B1EY-OHft-ixYWIyMD4hODUdfvYhn2sdE3N=>

We warn you about probability of your personal data unlawful using.

According to Google Privacy Policy we can't restrict access to the page without reasons for more than 24 hours. Therefore please respond to this message to delete the page.

Google Company considers user's confidentiality as first-priority factor. We collect exclusively personal identification data provided yourself. We don't disclose, don't spread and don't share your personal data with other organizations with any purposes.

Google Monitoring Center.

# Example of phishing email

From [REDACTED]  
Subject **Privacy violation**  
To [REDACTED]

Dear Customer,

Please be informed Drive Service. For access to the following link:


<https://docs.google.com/...>

We warn you about [REDACTED]

According to Google [REDACTED] without reasons for [REDACTED] message to delete [REDACTED]


Google Company confidential. We collect [REDACTED] yourself. We don't [REDACTED] data with other [REDACTED]

Google Monitoring [REDACTED]



Please re-enter your password

To help protect your privacy, we will sometimes ask you to verify your password even though you are already signed in.



[REDACTED]  
[REDACTED]@gmail.com

[Sign in](#)

[Need help?](#)

[To Sign in with a different account](#)

ound on Google  
orary restricted

[D4hODUdfvYhn2sdE3N=](#)

nlawful using.

access to the page  
ase respond to this

rst-priority  
n data provided  
hare your personal

# Agenda

- What is LoJack?
- Past research
- Digging in
- Descending through the rings

Computrace/LoJack

# Absolute Software



The image is a screenshot of the Absolute Software website. The background is a dark teal color with a subtle pattern of white dots and lines, resembling a network or data structure. The website content is displayed in a dark grey/black overlay. At the top, there is a navigation bar with a mail icon, a document icon, a US flag, and links for 'English (North America)', 'Report a Theft', 'Renew', 'Download', and 'Login'. Below the navigation bar is the Absolute LO/JACK logo, with 'Absolute' in a serif font and 'LO/JACK' in a bold, sans-serif font. To the right of the logo is a hamburger menu icon. The main headline reads 'IT'S LIKE A PERSONAL SECURITY DETAIL FOR YOUR LAPTOP, TABLET or PHONE' in a bold, white, sans-serif font. Below the headline is the sub-headline 'The industry's only Investigations and Recovery Team' in a smaller, white, sans-serif font. A white-bordered button with the text 'GET ABSOLUTE LOJACK' is positioned below the sub-headline. The bottom section of the screenshot has a white background and contains the text 'THE LEADER IN DATA AND DEVICE PROTECTION' in a bold, black, sans-serif font. Below this is a paragraph of text: 'Absolute Lojack is the only persistent security solution that can track and recover stolen devices, while providing features that protect your personal information.' The words 'track and recover stolen devices' are highlighted in red.

English (North America) | Report a Theft | Renew | Download | Login

Absolute | **LO/JACK**

**IT'S LIKE A PERSONAL SECURITY DETAIL  
FOR YOUR LAPTOP, TABLET or PHONE**

The industry's only Investigations and Recovery Team

GET ABSOLUTE LOJACK

**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.

# Past Research

# 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

# LoJack Architecture back then

UEFI/BIOS module  
executes

Windows  
early boot

Windows  
OS running

1

UEFI/BIOS module

Contains persistent  
agent and its dropper

Replaces legitimate  
`autochk.exe`

2

`autochk.exe`

Drops `rpcnetp.exe`  
- small agent

Installs it as a service

3

`rpcnetp.exe`  
- small agent

Injects its DLL into  
`svchost`, then internet  
explorer

Communicates with C&C  
server to download and  
install full recovery agent

4


Normal operation

Full recovery agent  
is running on the machine



# Configuration file vulnerability

```
.00406020: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00406030: 00 00 00 00-00 00 00 00-04 02 00 00-80 1E 04 01
.00406040: 00 40 00 1F-04 00 00 00-00 10 0A F4-F4 85 F8 84
.00406050: EC 85 85 85-85 1D 02 00-00 46 06 00-00 00 00 00
.00406060: 00 47 06 00-00 00 00 00-00 48 1A B5-E5 64 80 C4
.00406070: A2 C6 D0 D4-C7 D6 DD 9B-DB D4 D8 D0-C4 C0 D0 C7
.00406080: CC 9B D6 DA-D8 0A 02 07-10 06 06 00-00 00 00 00
.00406090: 00 07 06 00-00 00 00 00-00 0F 06 B6-69 CE 05 05
.004060A0: 96 08 06 19-99 08 08 12-12 0B 02 62-03 14 04 39
.004060B0: 00 80 00 20-04 00 00 00-00 15 04 00-00 00 00 19
.004060C0: 1B 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.004060D0: 00 00 00 00-00 00 00 00-00 00 00 00-1A 01 00 1B
.004060E0: 06 00 00 00-00 00 00 2D-01 B8 2D 01-B8 33 01 B8
.004060F0: 2B 04 F4 E1-F1 E1 28 03-00 00 00 01-38 01 E1 ED
.00406100: 81 B8 33 01-B8 2B 04 F4-E1 F1 E1 28-03 00 00 00
.00406110: 01 23 01 00-00 00 00 00-00 00 00 00-00 00 00 00
.00406120: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00406130: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00406140: 00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```



# 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
```

```
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
```

# 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
```



The image shows a hex dump of a configuration file. The left column contains hexadecimal addresses from 00003C20 to 00003D20. The next three columns contain the corresponding hexadecimal bytes. The right column shows the ASCII representation of these bytes. A red box highlights a portion of the ASCII column, specifically the text 'search.namequery.com' which is part of a URL. The surrounding text in the ASCII column is mostly garbled or non-printable characters.

The background is a solid teal color with a subtle, abstract pattern of white lines and dots, resembling a network or a molecular structure. The lines are thin and connect various points, some of which are small white dots. The overall effect is a clean, modern, and technical aesthetic.

Digging in

## LoJax - Cat is out of the bag

### Lojack Becomes a Double-Agent

**A** [ASERT team](#) on May 1, 2018.

- Document small agent modifications
- Links old Sednit domains to Lojax domains

# Where is the attack?

UEFI/BIOS module  
executes

Windows  
early boot

Windows  
OS running

1

UEFI/BIOS module

Contains persistent  
agent and its dropper

Replaces legitimate  
`autochk.exe`

2

`autochk.exe`

Drops `rpcnetp.exe`  
- small agent

Installs it as a service

3

`rpcnetp.exe`  
- small agent

Injects its DLL into  
`svchost`, then internet  
explorer

Communicates with C&C  
server to download and  
install full recovery agent

4

Normal operation

Full recovery agent  
is running on the machine

# Where is the attack?

UEFI/BIOS module executes

Windows early boot

Windows OS running

1

UEFI/BIOS module

Contains persistent agent and its dropper

Replaces legitimate `autochk.exe`

2

`autochk.exe`

Drops `rpcnetp.exe` - small agent

Installs it as a service

3

`rpcnetp.exe`  
- small agent

Injects its DLL into `svchost`, then internet explorer

Communicates with C&C server to download and install full recovery agent

4

Normal operation

Full recovery agent is running on the machine

## Changed only configuration file?

- Almost, and used only one agent version to do so...

```
00003c30: 0000 0000 0000 0000 0402 0000 801e 0401 .....
00003c40: 0040 001f 0400 0000 0010 0af4 f485 f884 .@.....
00003c50: ec85 8585 851d 0200 0046 0600 0000 0000 .....F..... →
00003c60: 0047 0600 0000 0000 0048 1ab5 e564 80c4 .G.....H...d.. ←
00003c70: a2c6 d0d4 c7d6 dd9b dbd4 d8d0 c4c0 d0c7 .....
00003c80: cc9b d6da d80a 0207 1006 0600 0000 0000 .....
00003c90: 0007 0600 0000 0000 000f 06b6 69ce 0505 .....i...
00003ca0: 9608 0619 9908 0812 120b 0262 0314 0439 .....b...9

00003c30: 0000 0000 0000 0000 0402 0000 801e 0401 .....
00003c40: 0040 001f 0400 0000 0010 0af4 f485 f884 .@.....
00003c50: ef85 8585 851d 0200 0046 0600 0000 0000 .....F.....
00003c60: 0047 0600 0000 0000 0048 1ab5 e5e3 df36 .G.....H...6
00003c70: 83d0 d9d4 cdda 9bda c7d2 b5b5 b5b5 b5b5 .....
00003c80: b5b5 b5b5 b50a 0207 1006 0600 0000 0000 .....
00003c90: 0007 0600 0000 0000 000f 06aa fda6 8805 .....
00003ca0: 9608 0619 9908 0812 120b 0262 0314 0439 .....b...9
```



## Changed only configuration file?

- Almost, and used only one agent version to do so...

```
00003c30: 0000 0000 0000 0000 0402 0000 801e 0401 .....
00003c40: 0040 001f 0400 0000 0010 0af4 f485 f884 .@.....
00003c50: ec85 8585 851d 0200 0046 0600 0000 0000 .....F..... → ←
00003c60: 0047 0600 0000 0000 0048 1ab5 e564 80c4 .G.....H...d..
00003c70: a2c6 d0d4 c7d6 dd9b dbd4 d8d0 c4c0 d0c7 .....
00003c80: cc9b d6da d80a 0207 1006 0600 0000 0000 .....
00003c90: 0007 0600 0000 0000 000f 06b6 69ce 0505 .....i...
00003ca0: 9608 0619 9908 0812 120b 0262 0314 0439 .....b...9

00003c30: 0000 0000 0000 0000 0402 0000 801e 0401 .....
00003c40: 0040 001f 0400 0000 0010 0af4 f485 f884 .@.....
00003c50: 505 0505 051d 0200 0046 0600 0000 0000 .....F.....
00003c60: 0047 0600 0000 0000 0048 1ab5 e5e3 df36 .....H...6
00003c70: 83d0 d9d4 cdda 9bda c7d2 b5b5 b5b5 b5b5 .....
00003c80: b5b5 b5b5 b50a 0207 1006 0600 0000 0000 .....
00003c90: 0007 0600 0000 0000 000f 06b6 69ce 0505 .....i...
00003ca0: 9608 0619 9908 0812 120b 0262 0314 0439 .....b...9
```

- Bulk detection now possible – time to dive in

## The Balkans, Central and Eastern Europe victims

- Few organizations hit
- Military and diplomatic organizations
- Presence of several Sednit tools in the organization

# Analyst ramblings

# Clairvoyance?



**Blog**

## Where are all the 'A's in APT?

Posted by  on  Sep 20, 2018

*In a guest blog post by VB2018 gold partner Kaspersky Lab, Costin Raiu, Director of the company's Global Research and Analysis Team, looks critically at the 'A' in APT.*

# Clairvoyance?

The logo for 'virus' features the word in a bold, green, sans-serif font. A small red dot is positioned above the letter 'i'. Below the main text, the phrase 'Covering the global' is partially visible in a smaller, lighter green font.

## So what is missing?

Looking at the discussions and development of sophisticated attack techniques, there is a significant difference between the theory and in-the-wild observations. So what is missing? Here's a list of possible culprits:

- Virtualization / hypervisor malware – although the infamous Blue Pill was discussed as far back as 2006, we haven't seen any in-the-wild (ItW) attacks leveraging this.
- SMM malware – although Dmytro Oleksiuk, a.k.a. Cr4sh, developed an **SMM backdoor** as far back as 2015, this is something yet to be seen in real-world attacks.
- UEFI malware – the hacking of HackingTeam revealed that a UEFI persistence module has been available since at least 2014, but we have yet to observe real-world UEFI malware.

# Clairvoyance?

**virus**

Covering the global

## So what is missing?

Looking at the discussions and development of sophisticated attack techniques, there is a significant difference between the theory and in-the-wild observations. So what is missing? Here's a list of possible culprits:

- Virtualization / hypervisor malware – although the infamous Blue Pill was discussed as far back as 2006, we haven't seen

**but we have yet to observe real-world UEFI malware.**

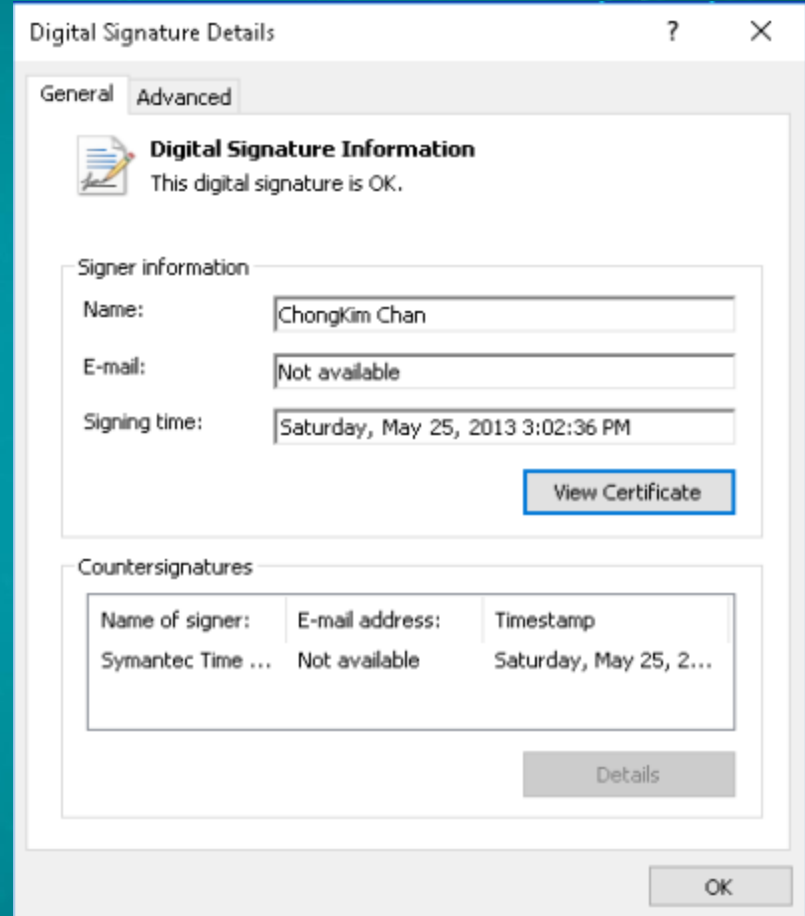
something yet to be seen in real-world attacks.

- UEFI malware – the hacking of HackingTeam revealed that a UEFI persistence module has been available since at least 2014, but we have yet to observe real-world UEFI malware.



# RWEverything

- Legitimate software using legitimate kernel driver
- Not the first time it is reused for other purposes





# RWEverything

- Found on some organizations with LoJax compromise
- info\_efi.exe

```
Get SMBIOS..
SMBIOS:
^@X^@A^BRê^C^@D |^@^@^@^@^G^D^F^@^@Phoenix Technologies LTD^@6.00^@05/19/2017^@^@A^[@^A^A^B^C
^DUM^S^Í^W^M^Ó^T~-(F^F^@^@VMware, Inc.^@VMware Virtual Platform^@None^@VMware-
^@^@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 ^@j^0^@C^B^@^@0u.^KA^D^U^@U^@jj^@^@^@A^A^@S^@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^@Aj^@P^A
```

# autochk.exe mechanism?

UEFI/BIOS module  
executes

Windows  
early boot

Windows  
OS running

1

UEFI/BIOS module

Contains persistent  
agent and its dropper

Replaces legitimate  
`autochk.exe`

2

`autochk.exe`

Drops `rpcnetp.exe`  
- small agent

Installs it as a service

3

`rpcnetp.exe`  
- small agent

Injects its DLL into  
`svchost`, then internet  
explorer

Communicates with C&C  
server to download and  
install full recovery agent

4

Normal operation

Full recovery agent  
is running on the machine

# autochk.exe mechanism?

UEFI/BIOS module  
executes

Windows  
early boot

Windows  
OS running

1

UEFI/BIOS module

Contains persistent  
agent and its dropper

Replaces legitimate  
`autochk.exe`

2

`autochk.exe`

Drops `rpcnetp.exe`  
- small agent

Installs it as a service

3

`rpcnetp.exe`  
- small agent

Injects its DLL into  
`svchost`, then internet  
explorer

Communicates with C&C  
server to download and  
install full recovery agent

4

Normal operation

Full recovery agent  
is running on the machine

# autochk.exe vs. autoche.exe

```
if ( NtOpenKey(&KeyHandle, 0xF003Fu, &ObjectAttributes) < 0 )
{
    NtCreateKey(&KeyHandle, KEY_ALL_ACCESS, &ObjectAttributes, 0u, 0u, 0u, 0u);
    RtlInitUnicodeString(&ValueName, L"DisplayName");
    RtlInitUnicodeString(&v5, L"Remote Procedure Call (RPC) Net");
    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
    {
        RtlInitUnicodeString(&ValueName, L"ObjectName");
        RtlInitUnicodeString(&v5, L"LocalSystem");
        if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
        {
            RtlInitUnicodeString(&ValueName, L"ErrorControl");
            Data = 1;
            if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &Data, 4u) >= 0 )
            {
                RtlInitUnicodeString(&ValueName, L"ImagePath");
                v19 = NtCreateFile(&FileHandle, 1u, &v24, &IoStatusBlock, 0u, 128u, 1u, 1u, 1u, 0u, 0u);
                RtlInitUnicodeString(&v5, L"C:\\Windows\\SysWOW64\\rpcnetp.exe");
                if ( v19 < 0 )
                    RtlInitUnicodeString(&v5, L"C:\\Windows\\System32\\rpcnetp.exe");
                if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 2u, v5.Buffer, v5.MaximumLength) >= 0 )
                {
                    RtlInitUnicodeString(&ValueName, L"Start");
                    v20 = 2;
                    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v20, 4u) >= 0 )
                    {
                        RtlInitUnicodeString(&ValueName, L"Type");
                        v21 = 16;
                        NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v21, 4u);
                    }
                }
            }
        }
    }
}
```

# autochk.exe vs. autoche.exe

```
if ( NtOpenKey(&KeyHandle, 0xF003Fu, &ObjectAttributes) < 0 )
{
    NtSetValueKey(&KeyHandle, KEY_ALL_ACCESS, &ObjectAttributes, 0u, 0u, 0u, 0u);
    RtlInitUnicodeString(&ValueName, L"DisplayName");
    RtlInitUnicodeString(&v5, L"Remote Procedure Call (RPC) Net");
    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
    {
        RtlInitUnicodeString(&ValueName, L"ObjectName");
        RtlInitUnicodeString(&v5, L"LocalSystem");
        if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
        {
            RtlInitUnicodeString(&ValueName, L"ErrorControl");
            Data = 1;
            if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &Data, 4u) >= 0 )
            {
                RtlInitUnicodeString(&ValueName, L"ImagePath");
                v19 = NtCreateFile(&FileHandle, 1u, &v24, &IoStatusBlock, 0u, 128u, 1u, 1u, 1u, 0u, 0u);
                RtlInitUnicodeString(&v5, L"C:\\Windows\\SysWOW64\\rpcnetp.exe");
                if ( v19 < 0 )
                    RtlInitUnicodeString(&v5, L"C:\\Windows\\System32\\rpcnetp.exe");
                if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 2u, v5.Buffer, v5.MaximumLength) >= 0 )
                {
                    RtlInitUnicodeString(&ValueName, L"Start");
                    v20 = 2;
                    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v20, 4u) >= 0 )
                    {
                        RtlInitUnicodeString(&ValueName, L"Type");
                        v21 = 16;
                        NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v21, 4u);
                    }
                }
            }
        }
    }
}
```

# autochk.exe vs. autoche.exe

```
if ( NtOpenKey(&KeyHandle, 0xF003Fu, &ObjectAttributes) < 0 )
{
    NtCreateKey(&KeyHandle, KEY_ALL_ACCESS, &ObjectAttributes, 0u, 0u, 0u, 0u);
    RtlInitUnicodeString(&ValueName, L"DisplayName");
    RtlInitUnicodeString(&v5, L"Remote Procedure Call (RPC) Net");
    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
    {
        RtlInitUnicodeString(&ValueName, L"ObjectName");
        RtlInitUnicodeString(&v5, L"LocalSystem");
        if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 1u, v5.Buffer, v5.MaximumLength) >= 0 )
        {
            RtlInitUnicodeString(&ValueName, L"ErrorControl");
            Data = 1;
            if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &Data, 4u) >= 0 )
            {
                RtlInitUnicodeString(&ValueName, L"ImagePath");
                v19 = NtCreateFile(&FileHandle, 1u, &v24, &IoStatusBlock, 0u, 128u, 1u, 1u, 1u, 0u, 0u);
                RtlInitUnicodeString(&v5, L"C:\\Windows\\SysWOW64\\rpcnetp.exe");
                if ( v19 < 0 )
                {
                    RtlInitUnicodeString(&v5, L"C:\\Windows\\System32\\rpcnetp.exe");
                    if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 2u, v5.Buffer, v5.MaximumLength) >= 0 )
                    {
                        RtlInitUnicodeString(&ValueName, L"Start");
                        v20 = 2;
                        if ( NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v20, 4u) >= 0 )
                        {
                            RtlInitUnicodeString(&ValueName, L"Type");
                            v21 = 16;
                            NtSetValueKey(KeyHandle, &ValueName, 0u, 4u, &v21, 4u);
                        }
                    }
                }
            }
        }
    }
}
```

# autochk.exe vs. autoche.exe

```
NtClose(FileHandle);
RtlInitUnicodeString(&v28, L"\\REGISTRY\\MACHINE\\SYSTEM\\CurrentControlSet\\Control\\Session Manager");
ObjectAttributes.Length = 24;
ObjectAttributes.RootDirectory = 0;
ObjectAttributes.Attributes = 512;
ObjectAttributes.ObjectName = &v28;
ObjectAttributes.SecurityDescriptor = 0;
ObjectAttributes.SecurityQualityOfService = 0;
NtOpenKey(&v23, 0xF003Fu, &ObjectAttributes);
*SourceString = 'u\0a';
v8 = 'o\0t';
v9 = 'h\0c';
v10 = 'c\0e';
v11 = ' \0k';
v12 = 'u\0a';
v13 = 'o\0t';
v14 = 'h\0c';
v15 = ' \0k';
v16 = '*';
v17 = 0;
RtlInitUnicodeString(&ValueName, L"BootExecute");
RtlInitUnicodeString(&v5, SourceString);
NtSetValueKey(v23, &ValueName, 0u, 7u, SourceString, 0x2Au);
return NtTerminateProcess(0xFFFFFFFF, 0);
```

# autochk.exe vs. autoche.exe

```
NtClose(FileHandle);
RtlInitUnicodeString(&v28, L"\\REGISTRY\\MACHINE\\SYSTEM\\CurrentControlSet\\Control\\Session Manager");
ObjectAttributes.Length = 24;
ObjectAttributes.RootDirectory = 0;
ObjectAttributes.Attributes = 512;
ObjectAttributes.ObjectName = &v28;
ObjectAttributes.SecurityDescriptor = 0;
ObjectAttributes.SecurityQualityOfService = 0;
NtOpenKey(&v23, 0x00000000, &ObjectAttributes);
*SourceString = 'u\0a';
v8 = 'o\0t';
v9 = 'h\0c';
v10 = 'c\0e';
v11 = ' \0k';
v12 = 'u\0a';
v13 = 'o\0t';
v14 = 'h\0c';
v15 = ' \0k';
v16 = '*';
v17 = 0;
RtlInitUnicodeString(&ValueName, L"BootExecute");
RtlInitUnicodeString(&v5, SourceString);
NtSetValueKey(v23, &ValueName, 0u, 7u, SourceString, 0x2Au);
return NtTerminateProcess(0xFFFFFFFF, 0);
```



# autochk.exe vs. autoche.exe

```
NtClose(FileHandle);
RtlInitUnicodeString(&v28, L"\\REGISTRY\\MACHINE\\SYSTEM\\CurrentControlSet\\Control\\Session Manager");
ObjectAttributes.Length = 0;
ObjectAttributes.RootDirectory = 0;
ObjectAttributes.Attributes = 512;
ObjectAttributes.ObjectName = &v28;
ObjectAttributes.SecurityDescriptor = 0;
ObjectAttributes.SecurityQualityOfService = 0;
NtOpenKey(&v23, 0xF003Fu, &ObjectAttributes);
*SourceString = 'u\\0a';
v8 = 'o\\0t';
v9 = 'h\\0c';
v10 = 'c\\0e';
v11 = '\\0k';
v12 = 'u\\0a';
v13 = 'o\\0t';
v14 = 'h\\0c';
v15 = '\\0k';
v16 = '*';
v17 = 0;
RtlInitUnicodeString(&ValueName, L"BootExecute");
RtlInitUnicodeString(&v23, SourceString);
NtSetValueKey(v23, &ValueName, 0u, 7u, SourceString, 0x2Au);
return NtTerminateProcess(0xFFFFFFFF, 0);
```

The background is a solid teal color with a subtle, abstract pattern of white dots and thin white lines, resembling a network or molecular structure. The text is centered in the middle of the image.

Down the rings we go

## ReWriter\_read.exe

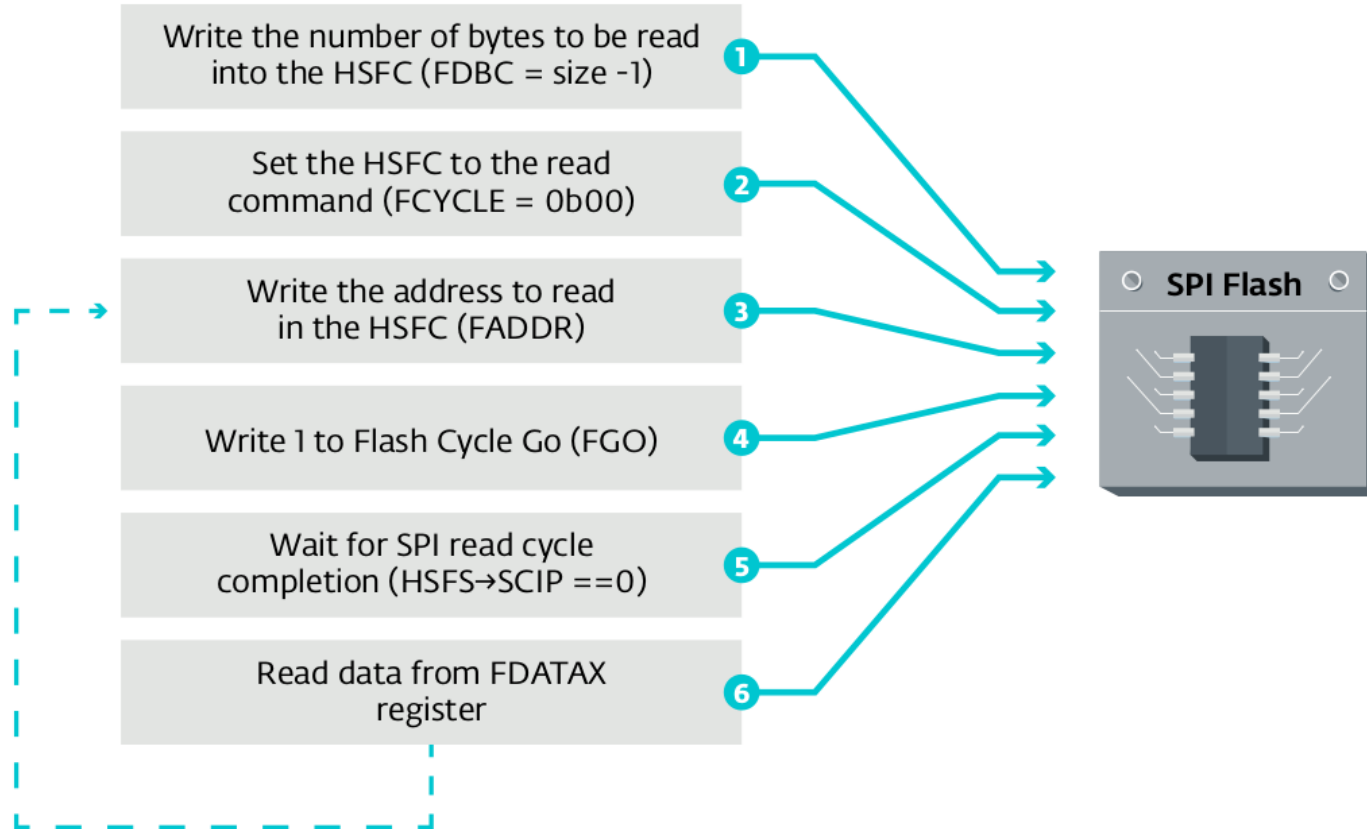
- Tool to dump SPI flash memory content found alongside LoJax sample

IOCTL code	Description
0x22280c	Writes to memory mapped I/O space
0x222808	Reads from memory mapped I/O space
0x222840	Reads a dword from given PCI Configuration Register
0x222834	Writes a byte to given PCI Configuration Register

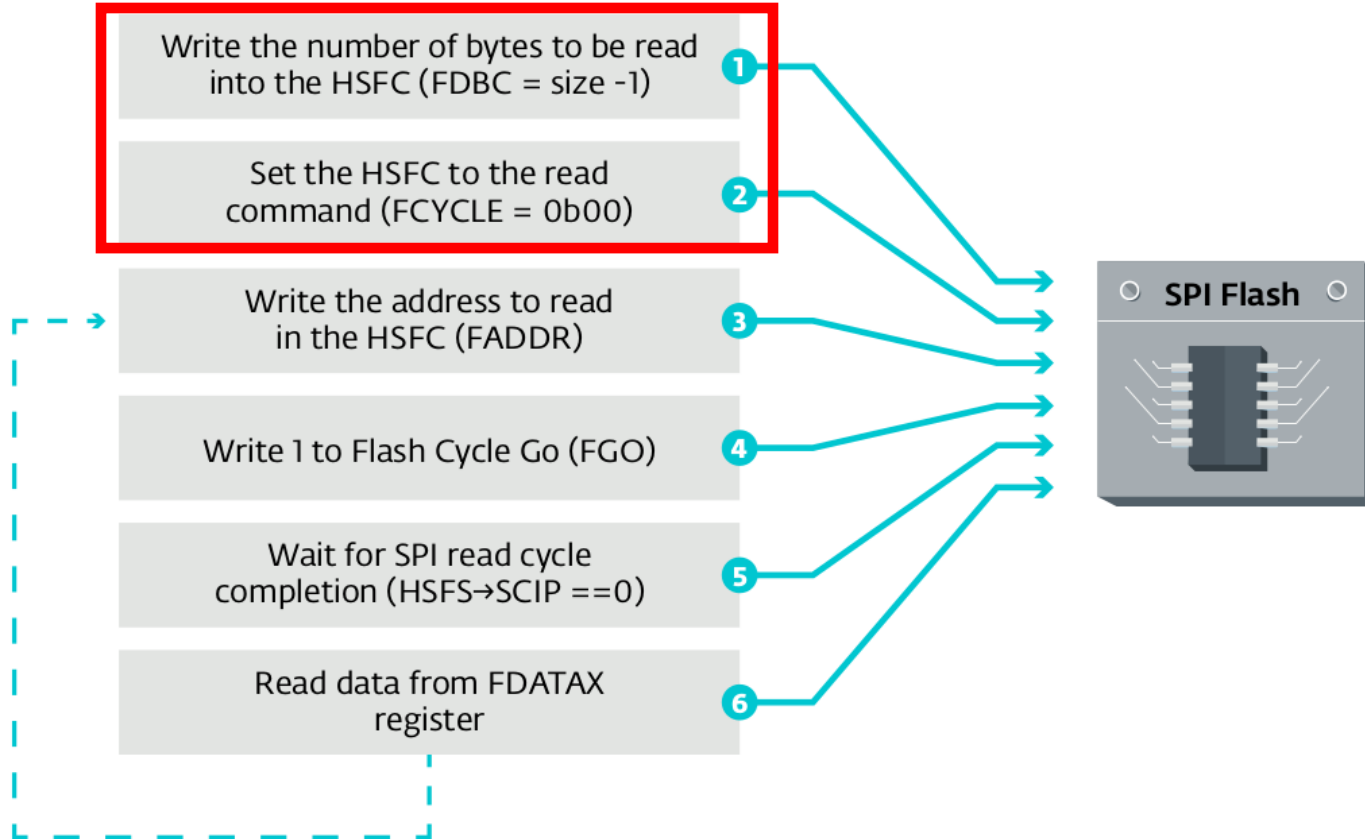
## ReWriter\_read.exe

- Contains *\*lots\** of debug strings
- Consists of the following operations
  - Log information on BIOS\_CNTL register
  - Locate BIOS region base address
  - Read UEFI firmware content and dump it to a file

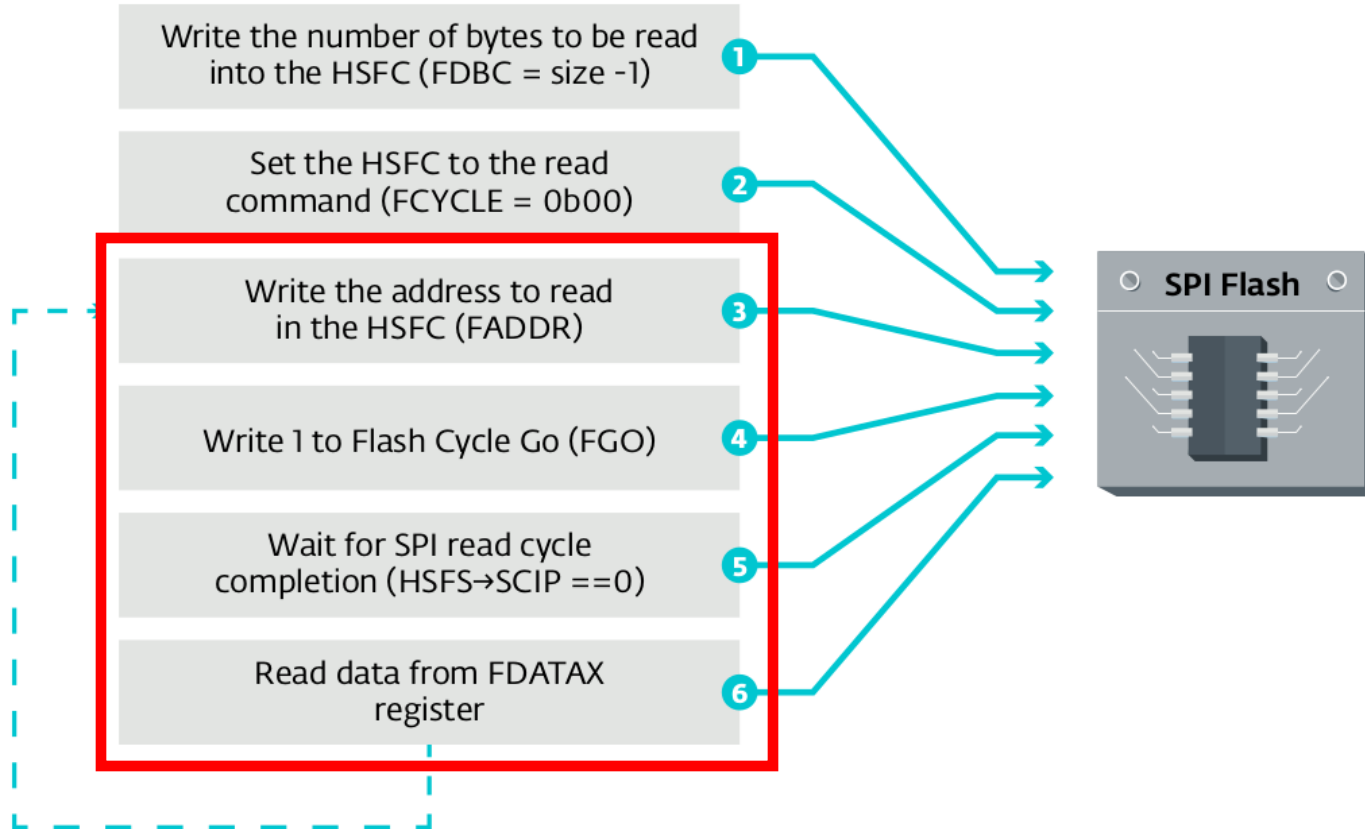
# Reading from the SPI Flash Memory



# Reading from the SPI Flash Memory



# Reading from the SPI Flash Memory



## ReWriter\_binary.exe

- Contains \*lots\* of debug strings
- Uses RWEverything's driver
- Consists of the following operations
  - Add the rootkit to the firmware
  - Write it back to the SPI flash memory



# Patching the UEFI firmware

# Unified Extensible Firmware Interface (UEFI)

- Replacement for the legacy BIOS
- New standard for firmware development
- Provides a set of services to UEFI applications
  - Boot services
  - Runtime services
- No more MBR/VBR

# 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
  - Files contain sections
  - One of these sections is the actual UEFI image
  - It's more complex than that but it suffices for our purpose

# SPI flash memory layout

File Action Help

### Structure

Name	Action	Type	Subtype	Text
▼ Intel image		Image	Intel	
Descriptor region		Region	Descriptor	
ME region		Region	ME	
▼ BIOS region		Region	BIOS	
00504624-8A59-4EEB-BD0F-6B36E96128E0		Volume	Unknown	
▼ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	
▶ 4A538818-5AE0-4EB2-B2EB-488B23657022		File	Volume image	
Volume free space		Free space		
Padding		Padding	Empty (0xFF)	
FFF12B8D-7696-4C8B-A985-2747075B4F50		Volume	Unknown	
Padding		Padding	Non-empty	
▶ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	

### Information

Full size: C00000h  
(12582912)  
Flash chips: 2  
Masters: 3  
PCH straps: 18  
CPU straps: 1

# SPI flash memory layout

The screenshot displays a software interface for SPI flash memory layout. The interface is divided into two main sections: 'Structure' and 'Information'.

**Structure:** A table listing the memory layout components. The 'Descriptor region' is highlighted with a red border.

Name	Action	Type	Subtype	Text
▼ Intel image		Image	Intel	
Descriptor region		Region	Descriptor	
ME region		Region	ME	
▼ BIOS region		Region	BIOS	
00504624-8A59-4EEB-BD0F-6B36E96128E0		Volume	Unknown	
▼ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	
▶ 4A538818-5AE0-4EB2-B2EB-488B23657022		File	Volume image	
Volume free space		Free space		
Padding		Padding	Empty (0xFF)	
FFF12B8D-7696-4C8B-A985-2747075B4F50		Volume	Unknown	
Padding		Padding	Non-empty	
▶ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	

**Information:** A panel on the right side of the interface providing details about the flash memory.

Full size: C00000h (12582912)  
Flash chips: 2  
Masters: 3  
PCH straps: 18  
CPU straps: 1

# SPI flash memory layout

File Action Help

### Structure

Name	Action	Type	Subtype	Text
▼ Intel image		Image	Intel	
Descriptor region		Region	Descriptor	
ME region		Region	ME	
▼ BIOS region		Region	BIOS	
00504624-8A59-4EEB-BD0F-6B36E96128E0		Volume	Unknown	
▼ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	
▶ 4A538818-5AE0-4EB2-B2EB-488B23657022		File	Volume image	
Volume free space		Free space		
Padding		Padding	Empty (0xFF)	
FFF12B8D-7696-4C8B-A985-2747075B4F50		Volume	Unknown	
Padding		Padding	Non-empty	
▶ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	

### Information

Full size: C00000h (12582912)  
Flash chips: 2  
Masters: 3  
PCH straps: 18  
CPU straps: 1

# SPI flash memory layout

The screenshot displays a software interface for SPI flash memory layout. The main window has a menu bar with 'File', 'Action', and 'Help'. The interface is divided into two main panes: 'Structure' on the left and 'Information' on the right.

**Structure Pane:** A table listing memory regions and their properties. The 'BIOS region' is highlighted with a red border.

Name	Action	Type	Subtype	Text
▼ Intel image		Image	Intel	
Descriptor region		Region	Descriptor	
▼ BIOS region		Region	BIOS	
00504624-8A59-4EEB-BD0F-6B36E96128E0		Volume	Unknown	
▼ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	
▶ 4A538818-5AE0-4EB2-B2EB-488B23657022		File	Volume image	
Volume free space		Free space		
Padding		Padding	Empty (0xFF)	
FFF12B8D-7696-4C8B-A985-2747075B4F50		Volume	Unknown	
Padding		Padding	Non-empty	
▶ 7A9354D9-0468-444A-81CE-0BF617D890DF		Volume	FFSv2	

**Information Pane:** Displays flash chip details.

```
Full size: C00000h  
(12582912)  
Flash chips: 2  
Masters: 3  
PCH straps: 18  
CPU straps: 1
```



# BIOS region layout

File Action Help

### Structure

Name	Action	Type	Subtype	Text
▼ 8C8CE578-8A3D-4F1C-9935-896185C32D...		Volume	FFSv2	
▶ FC510EE7-FFDC-11D4-BD41-0080C73C8...		File	Freeform	DXE apriori file
▶ FEDE0A1B-BCA2-4A9F-BB2B-D9FD7DEC2...		File	DXE driver	StatusCodeRuntimeDxe
▶ 80CF7257-87AB-47F9-A3FE-D50B76D89...		File	DXE driver	PcdDxe
▶ B601F8C4-43B7-4784-95B1-F4226CB40...		File	DXE driver	RuntimeDxe
▶ F80697E9-7FD6-4665-8646-88E33EF71...		File	DXE driver	SecurityStubDxe
▶ 53BCC14F-C24F-434C-B294-8ED2D4CC1...		File	DXE driver	DataHubDxe
▶ 13AC6DD0-73D0-11D4-B06B-00AA00BD6...		File	DXE driver	EbcDxe
▶ 79CA4208-BBA1-4A9A-8456-E1E66A814...		File	DXE driver	Legacy8259
▶ A19B1FE7-C1BC-49F8-875F-54A5D5424...		File	DXE driver	CpuIo2Dxe
▼ 1A1E4886-9517-440E-9FDE-3BE44CEE2...		File	DXE driver	CpuDxe
DXE dependency section		Section	DXE dependency	
PE32 image section		Section	PE32 image	
User interface section		Section	User interface	
Version section		Section	Version	
▶ F2765DEC-6B41-11D5-8E71-00902707B...		File	DXE driver	Timer
▶ A510A614-2192-11DF-AF29-2754E86B3...		File	DXE driver	PciExpressHostBridge
▶ 93B80004-9FB3-11D4-9A3A-0090273FC...		File	DXE driver	PciBusDxe
▶ 6B1C5323-297E-4720-B959-56D6F30FE...		File	DXE driver	YieldingDelayDxe
▶ 84562A94-1CFF-11DF-AB3F-FB61AA51C...		File	DXE driver	PmRuntimeDxe
▶ C8339973-A563-4561-B858-D8476F9DE...		File	DXE driver	Metronome
▶ 378D7B65-8DA9-4773-B6E4-A47826A83...		File	DXE driver	PcRtc

### Information

Type: 19h  
Full size: Ch (12)  
Header size: 4h (4)  
Body size: 8h (8)

# BIOS region layout

The screenshot displays a software interface for viewing BIOS region layout. The main window is titled "Structure" and contains a table with the following columns: Name, Action, Type, Subtype, and Text. A red rectangle highlights the first row of the table, which represents a volume. The "Information" panel on the right shows details for the selected volume, including its type, full size, header size, and body size.

Name	Action	Type	Subtype	Text
▼ 8C8CE578-8A3D-4F1C-9935-896185C32D...		Volume	FFSv2	
▶ FEDE0A1B-BCA2-4A9F-BB2B-D9FD7DEC2...		File	DXE driver	StatusCodeRuntimeDxe
▶ 80CF7257-87AB-47F9-A3FE-D50B76D89...		File	DXE driver	PcdDxe
▶ B601F8C4-43B7-4784-95B1-F4226CB40...		File	DXE driver	RuntimeDxe
▶ F80697E9-7FD6-4665-8646-88E33EF71...		File	DXE driver	SecurityStubDxe
▶ 53BCC14F-C24F-434C-B294-8ED2D4CC1...		File	DXE driver	DataHubDxe
▶ 13AC6DD0-73D0-11D4-B06B-00AA00BD6...		File	DXE driver	EbcDxe
▶ 79CA4208-BBA1-4A9A-8456-E1E66A814...		File	DXE driver	Legacy8259
▶ A19B1FE7-C1BC-49F8-875F-54A5D5424...		File	DXE driver	CpuIo2Dxe
▼ 1A1E4886-9517-440E-9FDE-3BE44CEE2...		File	DXE driver	CpuDxe
DXE dependency section		Section	DXE dependency	
PE32 image section		Section	PE32 image	
User interface section		Section	User interface	
Version section		Section	Version	
▶ F2765DEC-6B41-11D5-8E71-00902707B...		File	DXE driver	Timer
▶ A510A614-2192-11DF-AF29-2754E86B3...		File	DXE driver	PciExpressHostBridge
▶ 93B80004-9FB3-11D4-9A3A-0090273FC...		File	DXE driver	PciBusDxe
▶ 6B1C5323-297E-4720-B959-56D6F30FE...		File	DXE driver	YieldingDelayDxe
▶ 84562A94-1CFF-11DF-AB3F-FB61AA51C...		File	DXE driver	PmRuntimeDxe
▶ C8339973-A563-4561-B858-D8476F9DE...		File	DXE driver	Metronome
▶ 378D7B65-8DA9-4773-B6E4-A47826A83...		File	DXE driver	PcRtc

**Information**

Type: 19h  
Full size: Ch (12)  
Header size: 4h (4)  
Body size: 8h (8)

# BIOS region layout

File Action Help

### Structure

Name	Action	Type	Subtype	Text
▼ 8C8CF578-8A3D-4E1C-9935-896185C32D		Volume	EESv2	
▶ FC510EE7-FFDC-11D4-BD41-0080C73C8...		File	Freeform	DXE apriori file
▶ FEDE0A1B-BCA2-4A9F-BB2B-D9FD7DEC2...		File	DXE driver	StatusCodeRuntimeDxe
▶ 80CF7257-87AB-47F9-A3FE-D50B76D89...		File	DXE driver	PcdDxe
▶ B601F8C4-43B7-4784-95B1-F4226CB40...		File	DXE driver	RuntimeDxe
▶ F80697E9-7FD6-4665-8646-88E33EF71...		File	DXE driver	SecurityStubDxe
▶ 53BCC14F-C24F-434C-B294-8ED2D4CC1...		File	DXE driver	DataHubDxe
▶ 13AC6DD0-73D0-11D4-B06B-00AA00BD6...		File	DXE driver	EbcDxe
▶ 79CA4208-BBA1-4A9A-8456-E1E66A814...		File	DXE driver	Legacy8259
▶ A19B1FE7-C1BC-49F8-875F-54A5D5424...		File	DXE driver	CpuIo2Dxe
▼ 1A1E4886-9517-440E-9FDE-3BE44CEE2...		File	DXE driver	CpuDxe
DXE dependency section		Section	DXE dependency	
PE32 image section		Section	PE32 image	
User interface section		Section	User interface	
Version section		Section	Version	
▶ F2765DEC-6B41-11D5-8E71-00902707B...		File	DXE driver	Timer
▶ A510A614-2192-11DF-AF29-2754E86B3...		File	DXE driver	PciExpressHostBridge
▶ 93B80004-9FB3-11D4-9A3A-0090273FC...		File	DXE driver	PciBusDxe
▶ 6B1C5323-297E-4720-B959-56D6F30FE...		File	DXE driver	YieldingDelayDxe
▶ 84562A94-1CFF-11DF-AB3F-FB61AA51C...		File	DXE driver	PmRuntimeDxe
▶ C8339973-A563-4561-B858-D8476F9DE...		File	DXE driver	Metronome
▶ 378D7B65-8DA9-4773-B6E4-A47826A83...		File	DXE driver	PcRtc

### Information

Type: 19h  
Full size: Ch (12)  
Header size: 4h (4)  
Body size: 8h (8)

# BIOS region layout

File Action Help

### Structure

Name	Action	Type	Subtype	Text
▼ 8C8CE578-8A3D-4F1C-9935-896185C32D...		Volume	FFSv2	
▶ FC510EE7-FFDC-11D4-BD41-0080C73C8...		File	Freeform	DXE apriori file
▶ FEDE0A1B-BCA2-4A9F-BB2B-D9FD7DEC2...		File	DXE driver	StatusCodeRuntimeDxe
▶ 80CF7257-87AB-47F9-A3FE-D50B76D89...		File	DXE driver	PcdDxe
▶ B601F8C4-43B7-4784-95B1-F4226CB40...		File	DXE driver	RuntimeDxe
▶ F80697E9-7FD6-4665-8646-88E33EF71...		File	DXE driver	SecurityStubDxe
▶ 53BCC14F-C24F-434C-B294-8ED2D4CC1...		File	DXE driver	DataHubDxe
▶ 13AC6DD0-73D0-11D4-B06B-00AA00BD6...		File	DXE driver	EbcDxe
▶ 79CA4208-BBA1-4A9A-8456-E1E66A814...		File	DXE driver	Legacy8259
▶ A19B1FE7-C1BC-49F8-875F-54A5D5424...		File	DXE driver	CpuIo2Dxe
▼ 1A1E4886-9517-440E-9E9E-3BE44CFE2...		File	DXE driver	CpuDxe
DXE dependency section		Section	DXE dependency	
PE32 image section		Section	PE32 image	
User interface section		Section	User interface	
Version section		Section	Version	
▶ F2765DEC-6B41-11D5-8E71-00902707B...		File	DXE driver	Timer
▶ A510A614-2192-11DF-AF29-2754E86B3...		File	DXE driver	PciExpressHostBridge
▶ 93B80004-9FB3-11D4-9A3A-0090273FC...		File	DXE driver	PciBusDxe
▶ 6B1C5323-297E-4720-B959-56D6F30FE...		File	DXE driver	YieldingDelayDxe
▶ 84562A94-1CFF-11DF-AB3F-FB61AA51C...		File	DXE driver	PmRuntimeDxe
▶ C8339973-A563-4561-B858-D8476F9DE...		File	DXE driver	Metronome
▶ 378D7B65-8DA9-4773-B6E4-A47826A83...		File	DXE driver	PcRtc

### Information

Type: 19h  
Full size: Ch (12)  
Header size: 4h (4)  
Body size: 8h (8)

# Parsing the firmware volumes

- Parses all the firmware volumes of the UEFI firmware
- Looks for 4 specific files
  - Ip4Dxe (8f92960f-2880-4659-b857-915a8901bdc8)
  - NtfsDxe (768bedfd-7b4b-4c9f-b2ff-6377e3387243)
  - SmiFlash (bc327dbd-b982-4f55-9f79-056ad7e987c5)
  - DXE Core

## Ip4Dxe and DXE Core

- Used to find the firmware volume to install the rootkit
- All DXE drivers are usually in the same volume
- DXE Core may be in a different volume
- The chosen volume will be the one with enough free space available

# NtfsDxe and SmiFlash

- NtfsDxe the AMI NTFS driver
- Will be removed if found
- SmiFlash metadata are not used
- SmiFlash is a known-vulnerable DXE driver

# Adding the rootkit

- Creates a FFS file header (EFI\_FFS\_FILE\_HEADER)
- Append the Rootkit file

▼ 682894B5-6B70-4EBA-9E90-A607E5676297	File	DXE driver	SecDxe
▼ Compressed section	Section	Compressed	
PE32 image section	Section	PE32 image	
User interface section	Section	User interface	

- Write it at the end of the DXE drivers volume or the DXE Core volume
  - Checks if there's enough free space available



Write the compromised  
firmware to the SPI Flash  
memory

# BIOS Write Protection Mechanisms

- Platform exposes write protection mechanisms
- Need to be properly configured by the firmware
- We'll only cover relevant protections to our research
  - Won't cover Protected Range Registers
- Exposed via the BIOS Control Register (BIOS\_CNTL)

## 13.1.33 BIOS\_CNTL—BIOS Control Register (LPC I/F—D31:F0)

Offset Address: DCh  
Default Value: 20h  
Lockable: No

Attribute: R/WLO, R/W, RO  
Size: 8 bit  
Power Well: Core

# BIOS Write Protection Mechanisms

- To write to the BIOS region BIOS Write Enable (BIOSWE) must be set to 1
- BIOS Lock Enable (BLE) allows to lock BIOSWE to 0

1	<b>BIOS Lock Enable (BLE)</b> — R/WLO.
---	--

1	0 = Setting the BIOSWE will not cause SMIs.
---	---

1	1 = Enables setting the BIOSWE bit to cause SMIs. Once set, this bit can only be cleared by a PLTRST#
---	---

# BIOS Write Protection Mechanisms

- To write to the BIOS region BIOS Write Enable (BIOSWE) must be set to 1
- BIOS Lock Enable (BLE) allows to lock BIOSWE to 0

1	<b>BIOS Lock Enable (BLE) – R/WLO.</b> 0 = Setting the BIOSWE will not cause SMIs. 1 = Enables setting the BIOSWE bit to cause SMIs. Once set, this bit can only be cleared by a PLTRST#
---	--

# BIOS Write Protection Mechanisms

- The implementation of BLE is vulnerable
- When BIOSWE is set to 1, its value change in BIOS\_CNTL
- A System Management Interrupt (SMI) is triggered
- The SMI handler sets BIOSWE back to 0
  - The SMI handler must be implemented by the firmware

# BIOS Write Protection Mechanisms

- What if we write to the SPI flash memory before the SMI handler sets BIOSWE to 0?
- Race condition vulnerability (Speed racer)
  - A thread continuously set BIOSWE to 1
  - Another thread tries to write data
- Works on multicore processors and single core processors with hyper-threading enabled

# BIOS Write Protection Mechanisms

- Platform Controller Hub family of Intel chipsets introduces a fix for this issue

5	<p><b>SMM BIOS Write Protect Disable (SMM_BWP)</b>— R/WLO.</p> <p>This bit set defines when the BIOS region can be written by the host.</p> <p>0 = BIOS region SMM protection is disabled. The BIOS Region is writable regardless if processors are in SMM or not. (Set this field to 0 for legacy behavior)</p> <p>1 = BIOS region SMM protection is enabled. The BIOS Region is not writable unless all processors are in SMM.</p>
---	--

- The firmware must set this bit

# BIOS Write Protection Mechanisms

- Platform Controller Hub family of Intel chipsets introduces a fix for this issue

	<b>SMM BIOS Write Protect Disable (SMM_BWP)</b> — R/WLO.
	This bit set defines when the BIOS region can be written by the host.
5	0 = BIOS region SMM protection is disabled. The BIOS Region is writable regardless if processors are in SMM or not. (Set this field to 0 for legacy behavior)
	1 = BIOS region SMM protection is enabled. The BIOS Region is not writable unless all processors are in SMM.

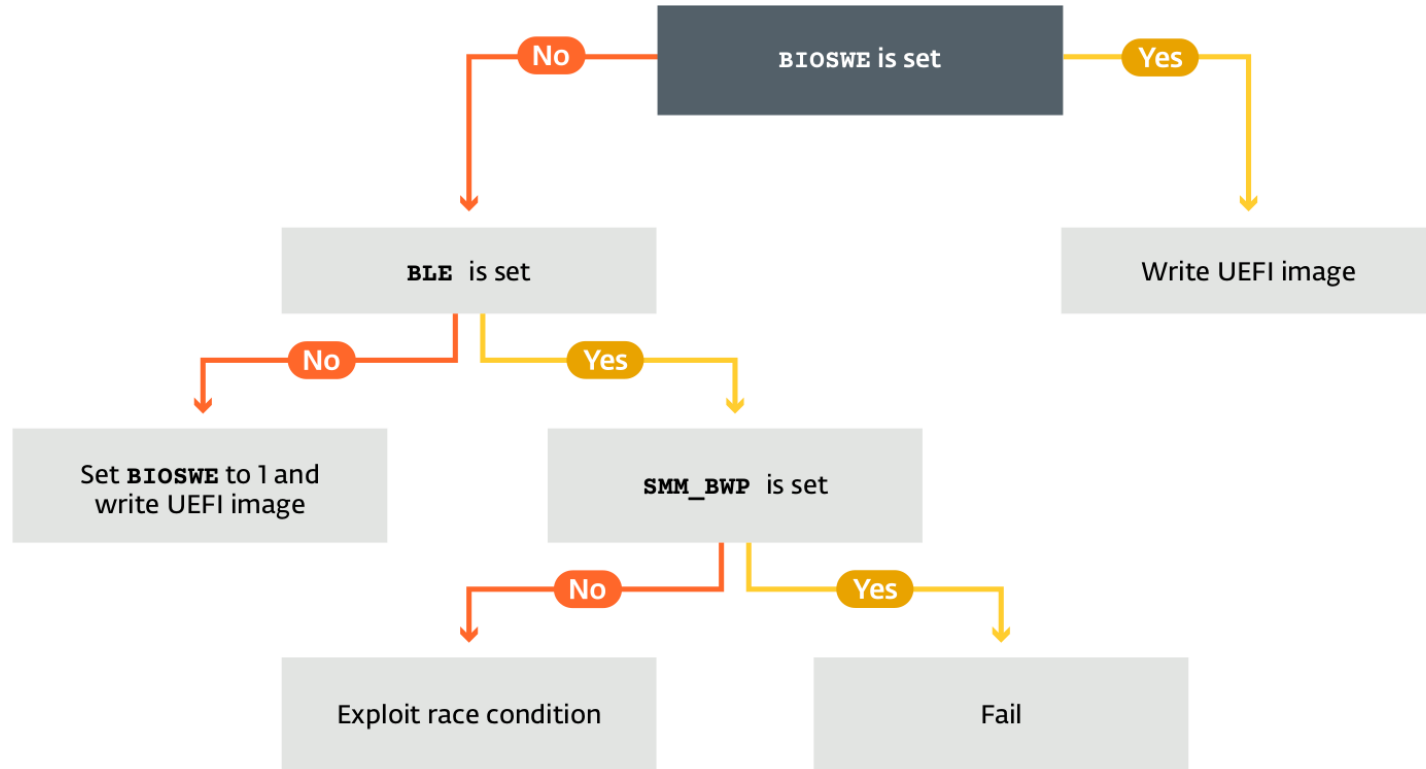
- The firmware must set this bit



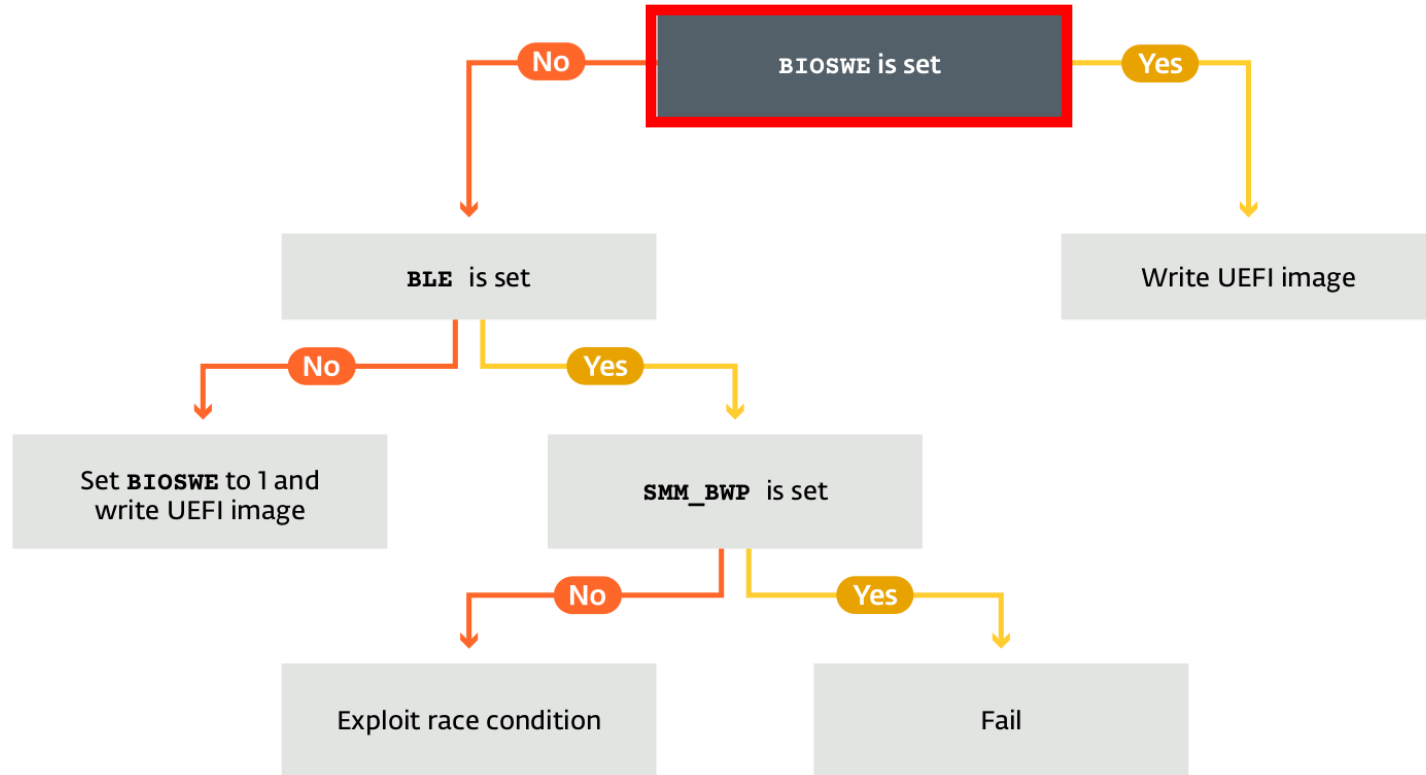
# ReWriter\_Binary.exe

- ReWriter\_Binary.exe checks these settings
- Checks if the platform is properly configured
- Implements the exploit for the race condition

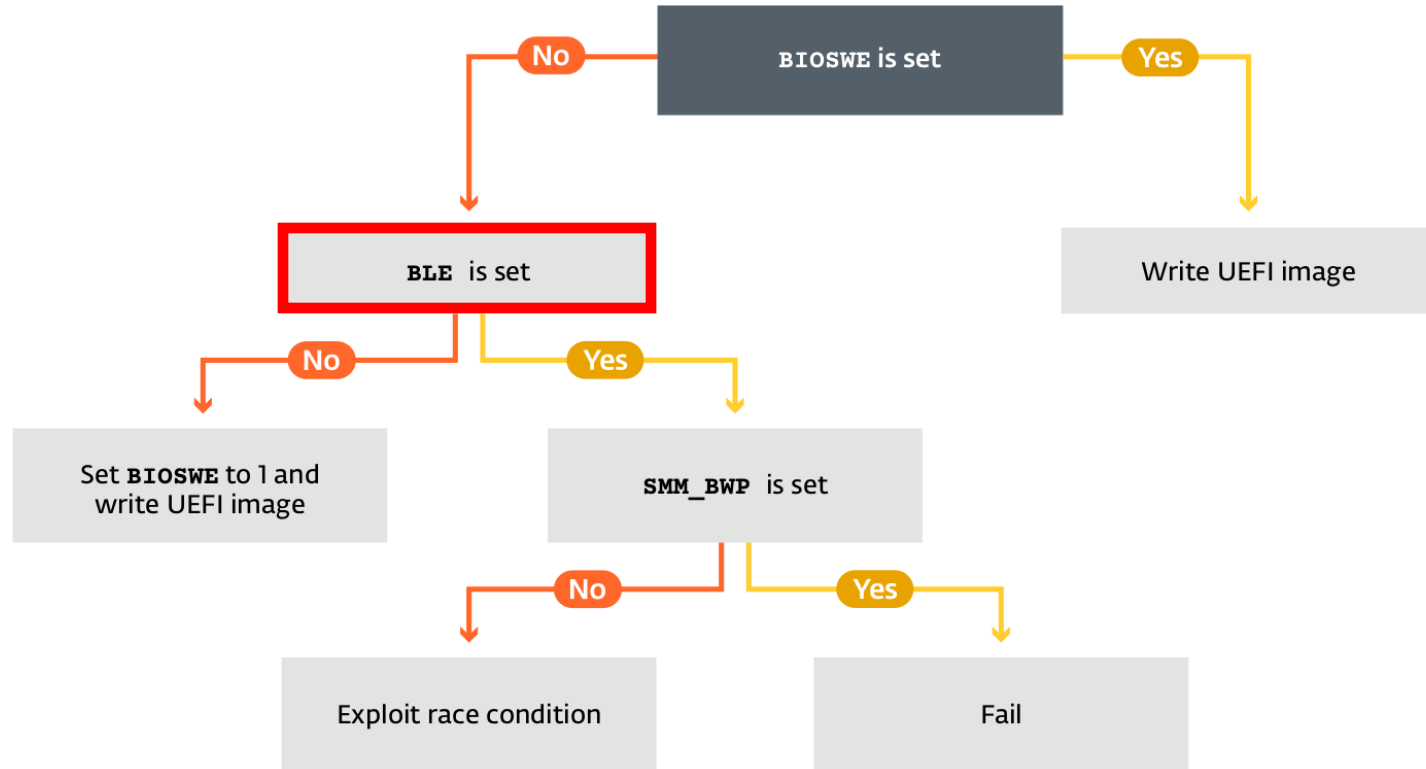
# Writing process decision tree



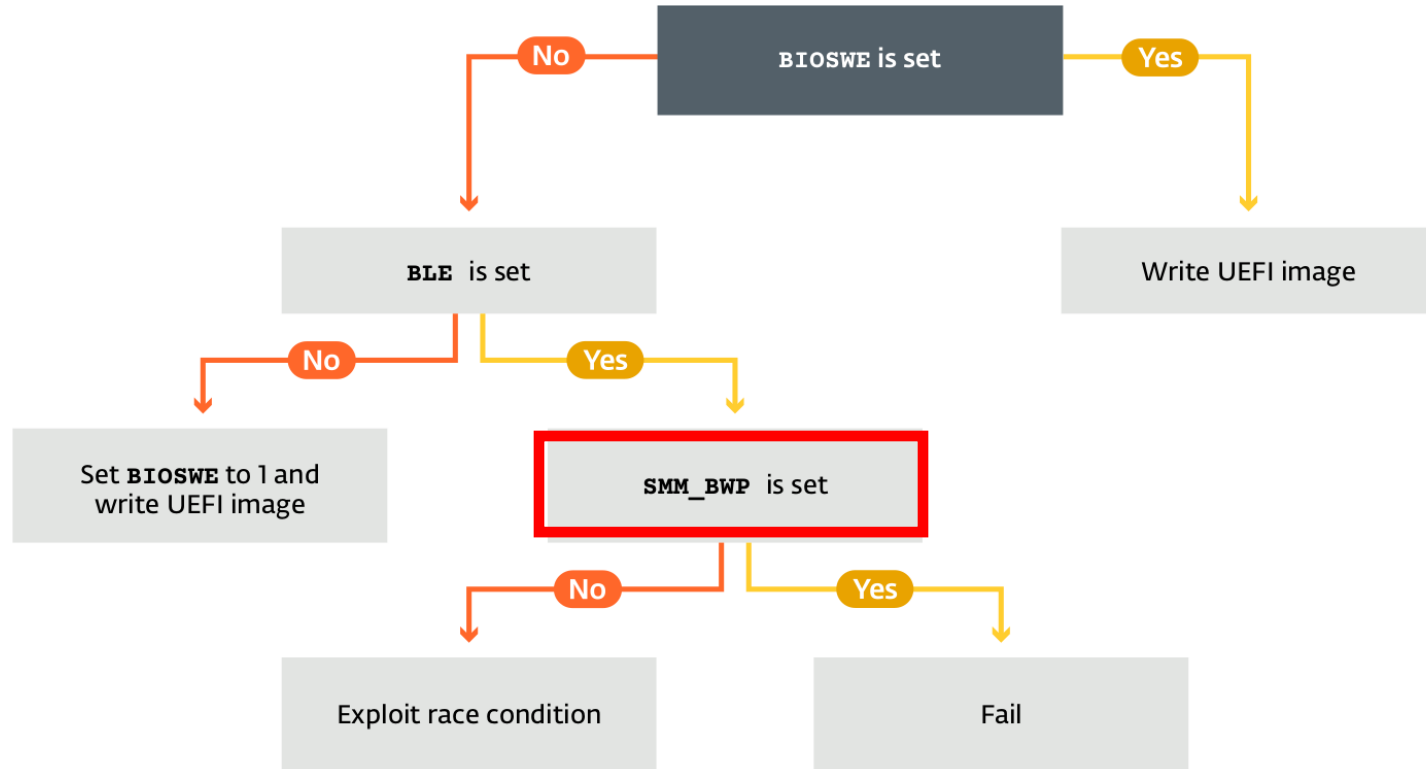
# Writing process decision tree



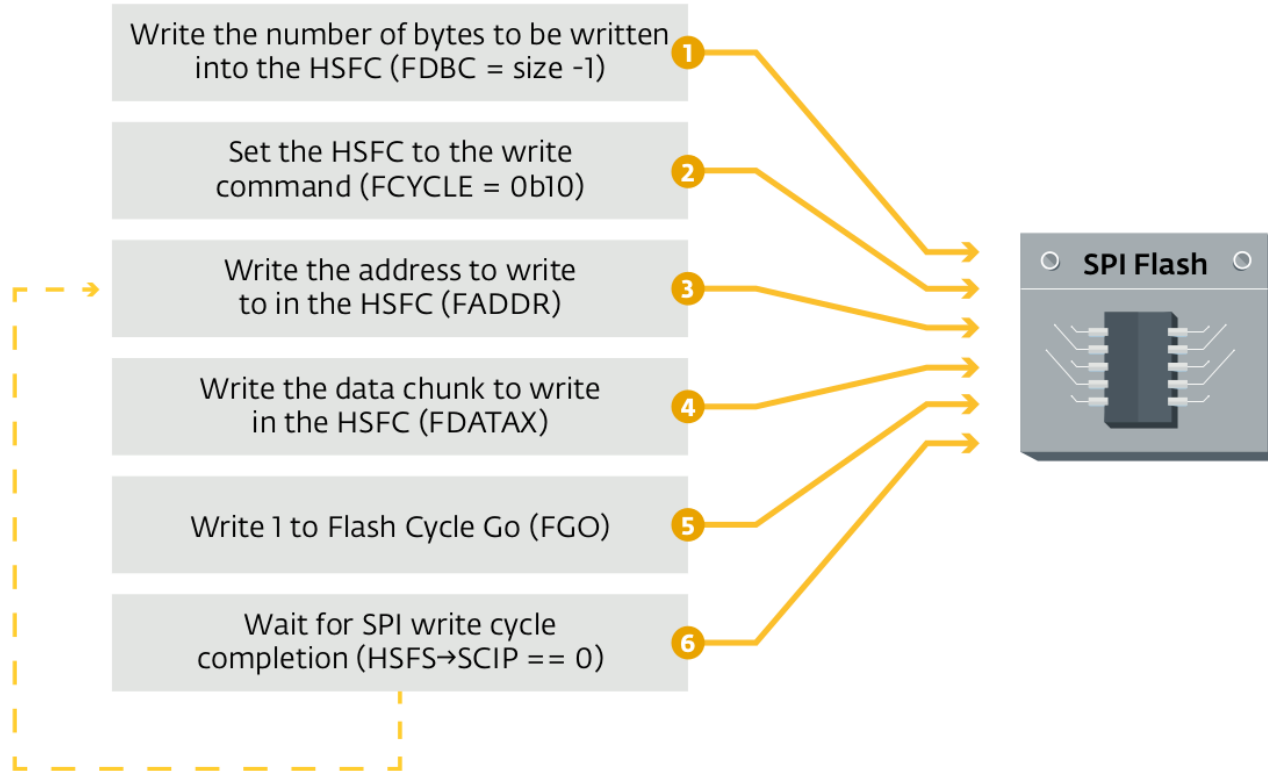
# Writing process decision tree



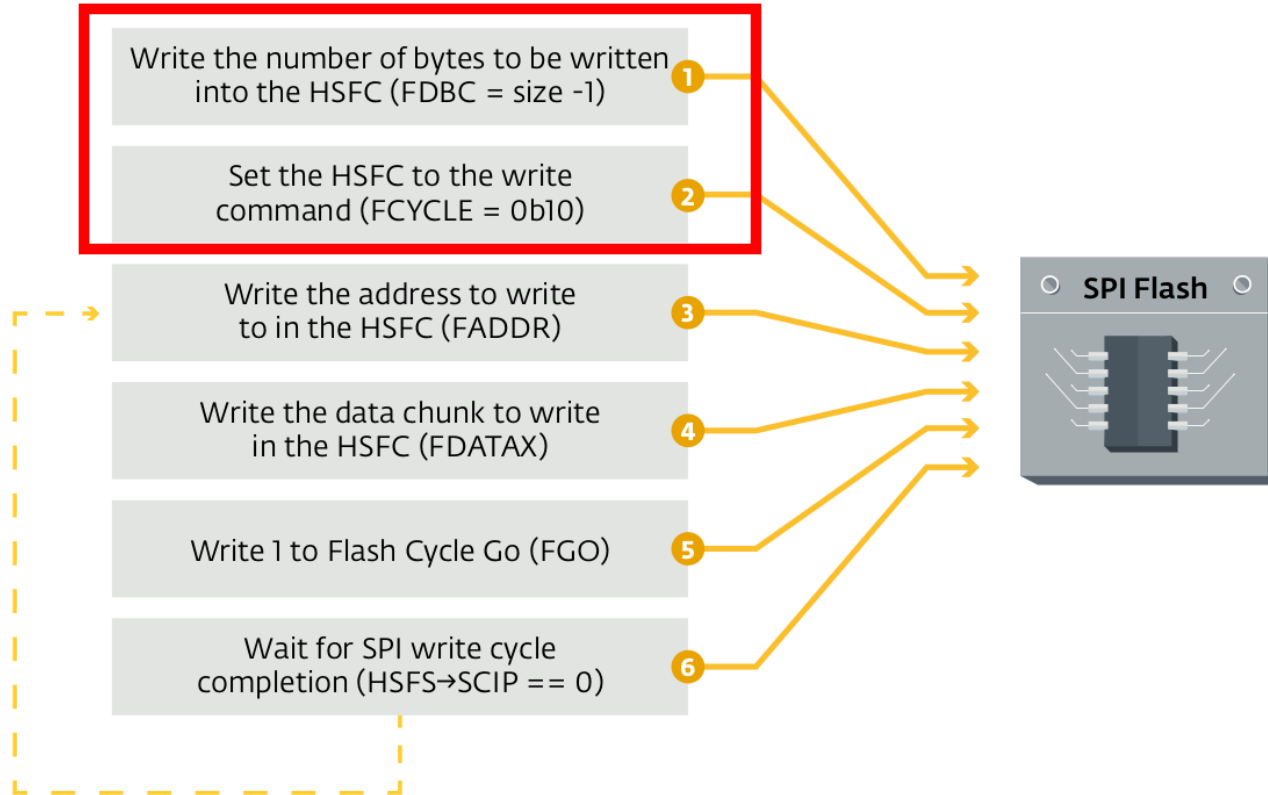
# Writing process decision tree



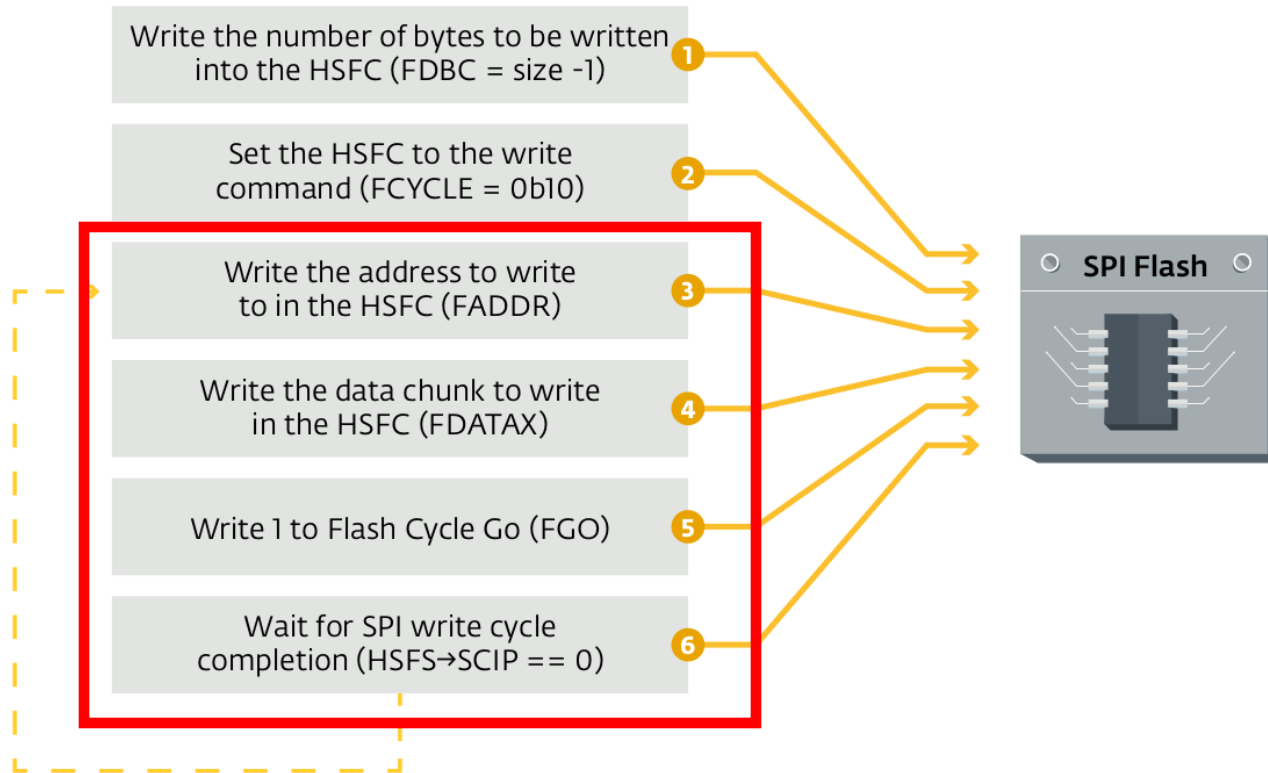
# Writing to the SPI Flash Memory



# Writing to the SPI Flash Memory



# Writing to the SPI Flash Memory





## Let's take a step back

- Software implementation to flash firmware remotely
  - Hacking Team's UEFI rootkit needed physical access
- We extracted the UEFI rootkit
- Looked at ESET's UEFI scanner telemetry
- And...

We're going to Black Hat  
Baby!



We're going to Black Hat  
Baby!

but we have yet to observe real-world UEFI malware.



We're going to Black Hat  
Baby!

~~but we have yet to observe real-world UEFI malware.~~

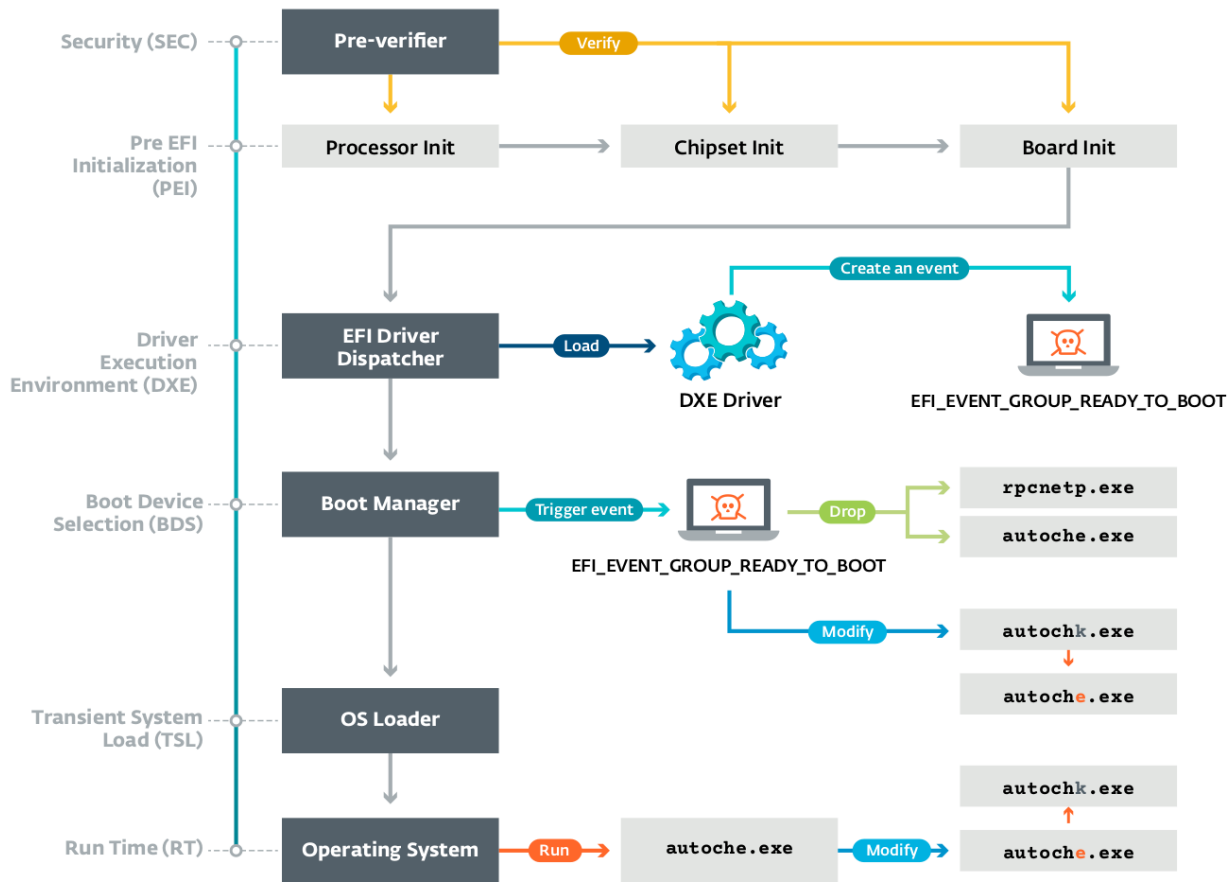


# UEFI Rootkit

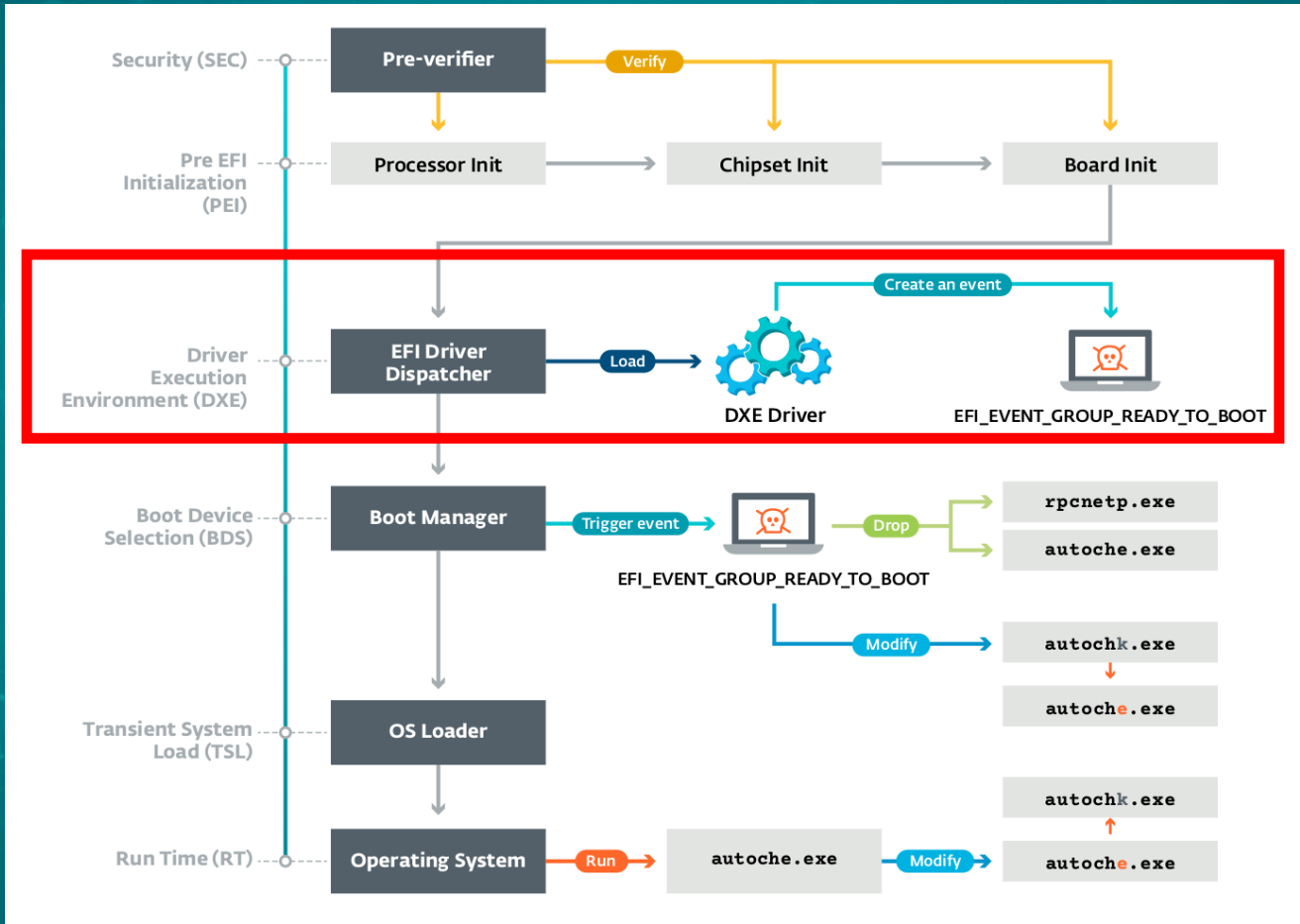
# UEFI Rootkit

- DXE Driver loaded by the DXE Dispatcher
- File Name
  - SecDxe
- File GUID
  - 682894B5-6B70-4EBA-9E90-A607E5676297

# UEFI Rootkit Workflow

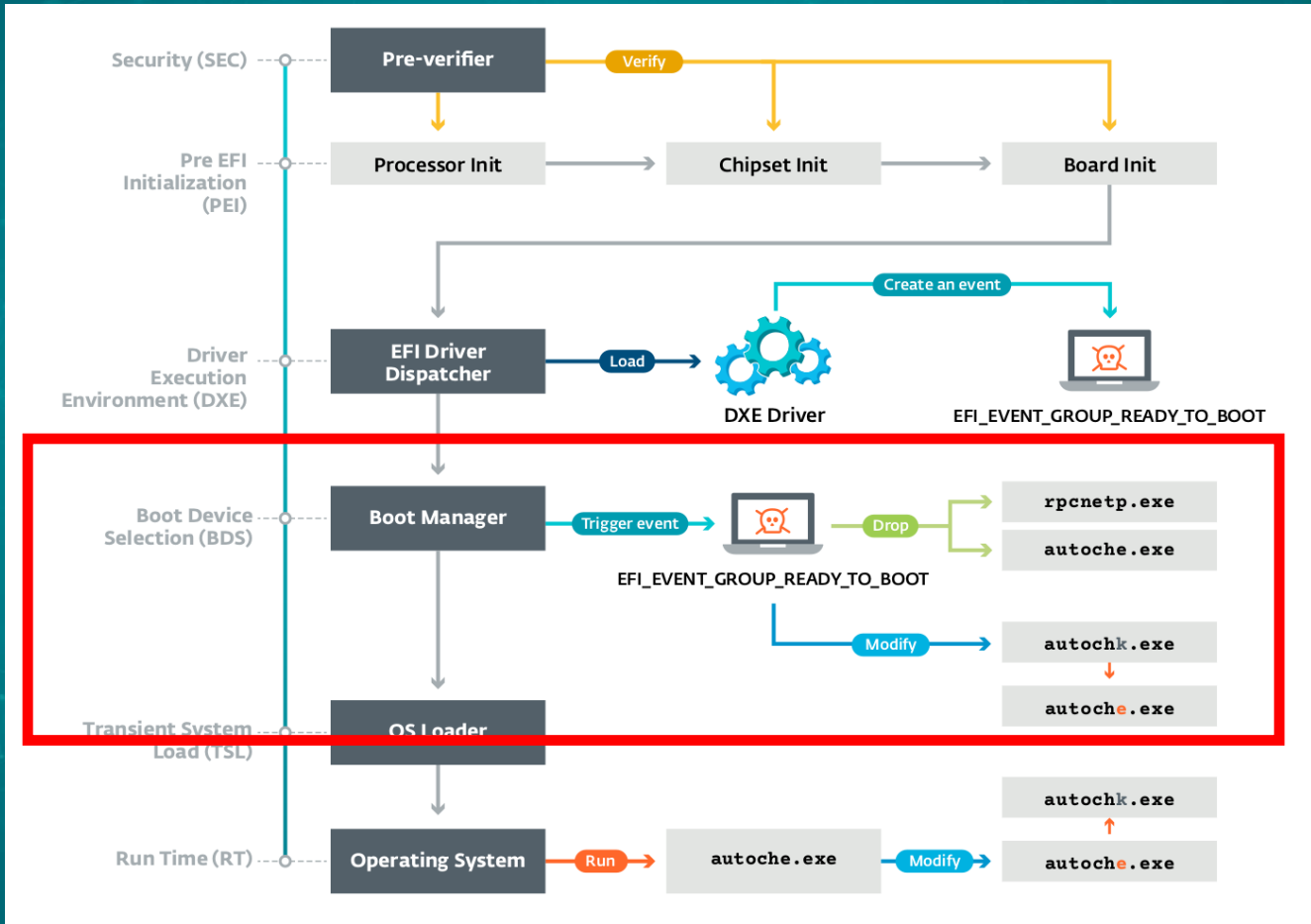


# UEFI Rootkit Workflow





# UEFI Rootkit Workflow



# UEFI Rootkit: SecDxe

- **Notify function**
  - Installs NTFS driver
  - Drops autoche.exe and rpcnetp.exe
  - Patch a value in the Windows Registry

## UEFI Rootkit: NTFS driver

- NTFS driver needed to get file-based access to Windows' partition
- Hacking Team's NTFS driver from HT's leak
  - NtfsDxe project from vector-edk

# UEFI Rootkit: Dropping files

```
else
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", lui64, 0x20ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", 0x8000000000000003ui64, 0x20ui64);
    (*NewHandle)->Write(*NewHandle, &RpcnetpFileSize, &gRpcnetp_exe);
  }
  (*NewHandle)->Close(*NewHandle);
}
v2 = (*WindowsDirHandle)->Open(*WindowsDirHandle, SystemDirHandle, System32Dir, lui64, 0x10ui64);
if ( !v2 )
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", lui64, 6ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", 0x8000000000000003ui64, 6ui64);
    (*NewHandle)->Write(*NewHandle, &AutocheFileSize, &gAutoche_exe);
  }
  v2 = (*NewHandle)->Close(*NewHandle);
}
```

# UEFI Rootkit: Dropping files

```
else
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", lui64, 0x20ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", 0x8000000000000003ui64, 0x20ui64);
    (*NewHandle)->Write(*NewHandle, &RpcnetpFileSize, &gRpcnetp_exe);
  }
  (*NewHandle)->Close(*NewHandle);
}
v2 = (*WindowsDirHandle)->Open(*WindowsDirHandle, SystemDirHandle, System32Dir, lui64, 0x10ui64);
if ( !v2 )
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", lui64, 6ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", 0x8000000000000003ui64, 6ui64);
    (*NewHandle)->Write(*NewHandle, &AutocheFileSize, &gAutoche_exe);
  }
  v2 = (*NewHandle)->Close(*NewHandle);
}
```

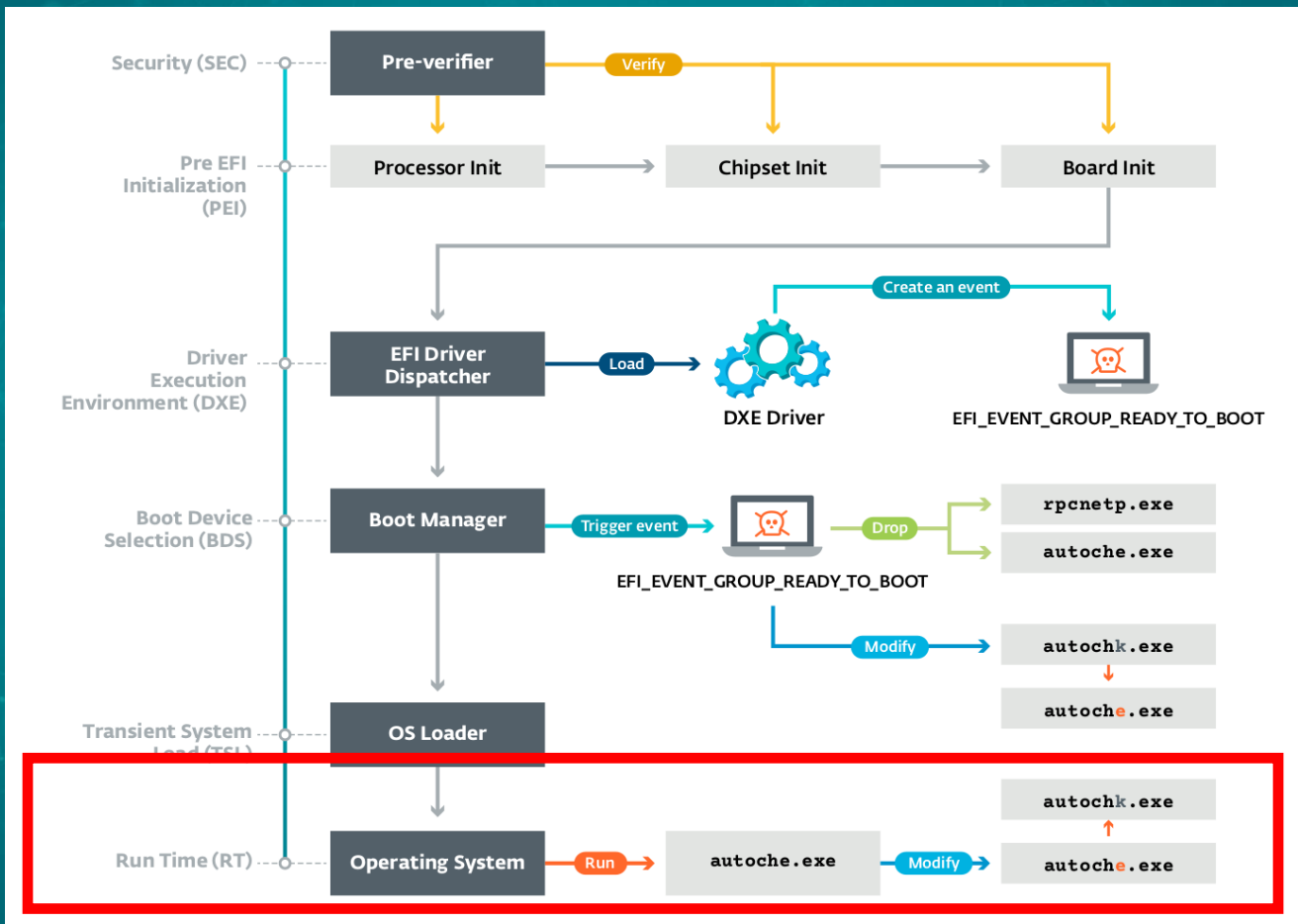
# UEFI Rootkit: Dropping files

```
else
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", lui64, 0x20ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"rpcnetp.exe", 0x8000000000000003ui64, 0x20ui64);
    (*NewHandle)->Write(*NewHandle, &RpcnetpFileSize, &gRpcnetp_exe);
  }
  (*NewHandle)->Close(*NewHandle);
}
v2 = (*WindowsDirHandle)->Open(*WindowsDirHandle, SystemDirHandle, System32Dir, lui64, 0x10ui64);
if ( !v2 )
{
  if ( (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", lui64, 6ui64) )
  {
    (*SystemDirHandle)->Open(*SystemDirHandle, NewHandle, L"autoche.exe", 0x8000000000000003ui64, 6ui64);
    (*NewHandle)->Write(*NewHandle, &AutocheFileSize, &gAutoche_exe);
  }
  v2 = (*NewHandle)->Close(*NewHandle);
}
```

## UEFI Rootkit: Patching Windows Registry Value

- Modifies Windows Registry via  
`%WINDIR%\System32\config\SYSTEM`
- Changes “autocheck autochk \*” to “autocheck autoche \*”
- `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\BootExecute`

# UEFI Rootkit Workflow







Demo

# Prevention and Remediation

## Prevention

- Keep your UEFI firmware up-to-date
- Enable Secure Boot
- Hardware Root of Trust (ex. Intel BootGuard)
- Hope that your firmware configures security mechanisms properly :-)
- Firmware security assessments can be done with CHIPSEC

# Remediation

- You need to reflash your UEFI firmware
- If it's not an option for you then...

# Remediation

- You need to reflash your UEFI firmware
- If it's not an option for you then...



## Conclusion

- UEFI rootkits are real-world threats
- Firmware must be built with security in mind
- Share knowledge about how to prevent and mitigate UEFI-based threats



ENJOY SAFER  
TECHNOLOGY™

Thanks!  
*Questions?*

White paper available at [welivesecurity.com](https://welivesecurity.com)

@jiboutin

@Freddrickk\_