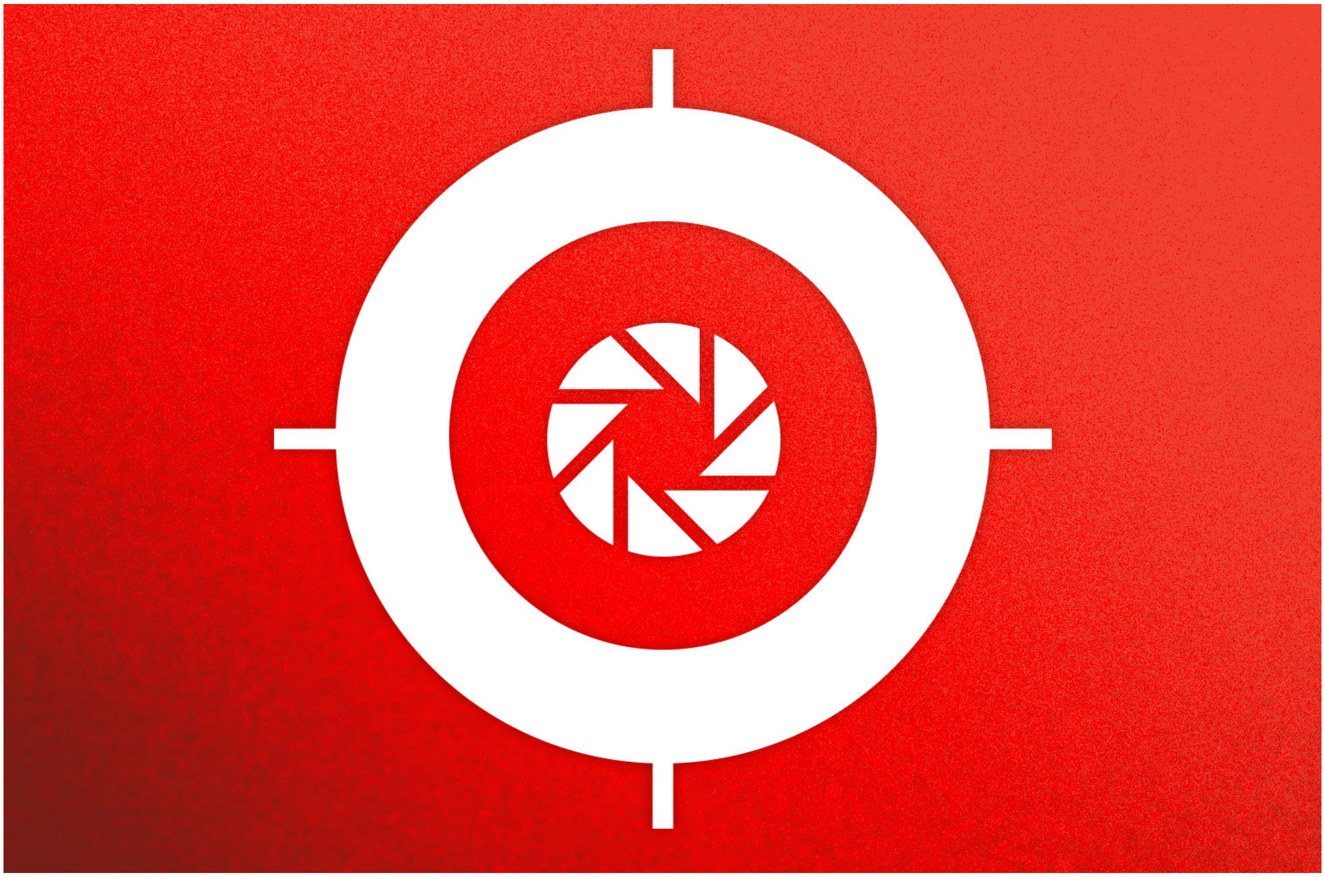
Conti Targets Critical Firmware

eclypsium.com/2022/06/02/conti-targets-critical-firmware/

Eclypsium

June 2, 2022



Introduction

In late February of this year, an unknown individual began leaking internal information and communications from the notorious Conti ransomware organization. These leaks appear to confirm the long-suspected connections between Conti and the Russian FSB, and provide key insight into the development of new threats and techniques.

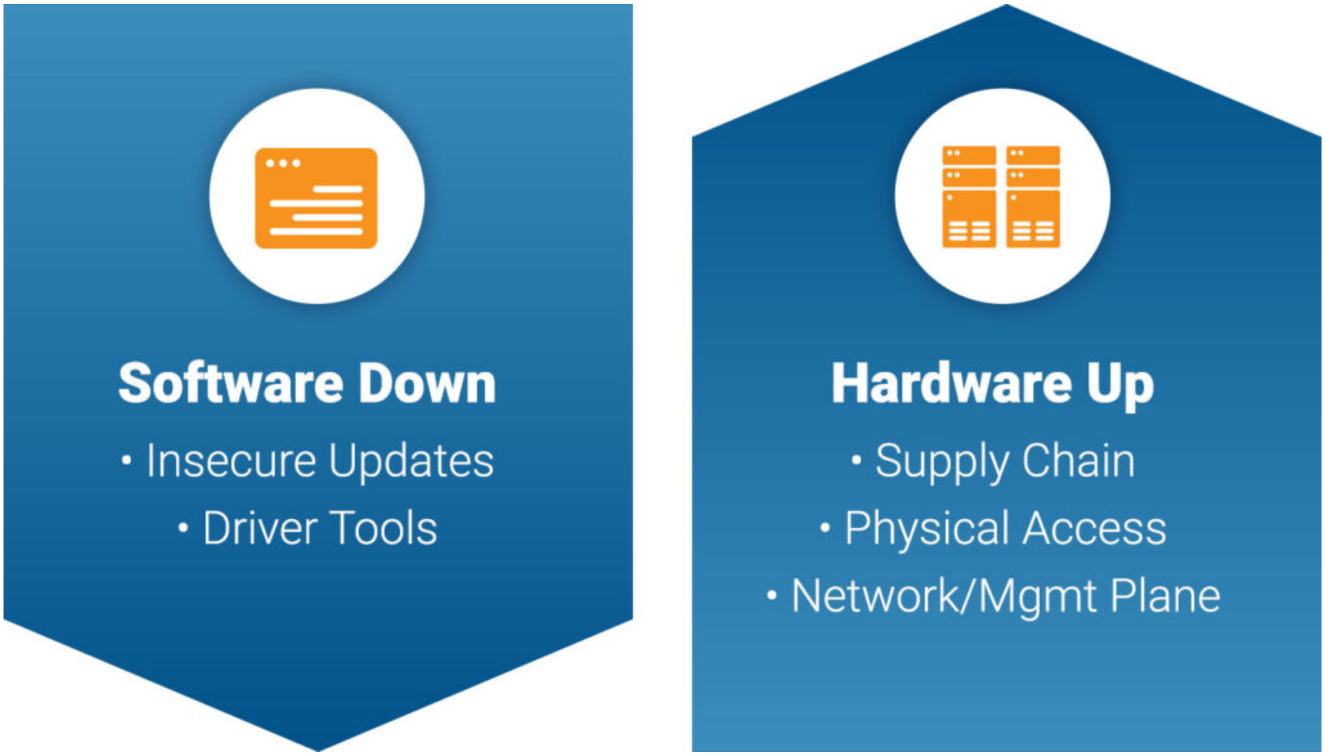
Notably, these leaked chats exposed a new front in the ongoing evolution of firmware-based attacks. In addition to classical attacks that target UEFI/BIOS directly, attackers are now targeting the Intel Management Engine (ME) or Intel Converged Security Management Engine (CSME). ME is a physical microcontroller that is part of the chipset of modern Intel-based systems. It supports a variety of capabilities such as out-of-band management. There are several different variations of this component to be aware of, Management Engine (before Skylake, also sometimes known as Intel Manageability Engine), Intel Converged Security and Management Engine (Skylake and newer), Intel Trusted Execution Environment (Atom platforms). In addition, there's an alternate firmware family used in servers known as Server

Platform Services. But in general, all of these refer to an out-of-band management processor which is built into the Intel chipset and runs independently from the CPU(s). For simplicity, we'll use the term ME firmware or simply, the chipset, to refer to this set of Intel management processors.
It is important to note that no new or unmitigated vulnerabilities have been identified and that Intel chipsets are no more or less vulnerable than any other code. The issue is that most organizations do not update their chipset firmware with the same regularity that they do their software or even the UEFI/BIOS system firmware. This can leave some of the most powerful and privileged code on a device susceptible to attack.

Compromising the Management Engine of a system would have considerable value on its own, but the leaks show that the group is using the unique privileges of the ME firmware as a way to gain indirect access to the UEFI/BIOS, drop additional payloads, and gain runtime control of the system below the operating system using System Management Mode (SMM). Such level of access would allow an adversary to cause irreparable damage to a system or to establish ongoing persistence that is virtually invisible to the operating system.

Leaked conversations indicate that the Conti group had already developed proof-of-concept code for these methods nine months ago. We expect that these techniques will be used in the wild in the near future if they haven't already so we wanted to share our insight into ME firmware in order to help organizations make better threat and impact-informed security decisions. Analysis includes:

- **Chipset Vulnerabilities** – While it is known that these adversaries are actively analyzing the ME for new vulnerabilities, we wanted to identify the known vulnerabilities that attackers could use so that organizations can reduce their firmware attack surface.
- **Attack Flow and Scenarios** – We show different attack scenarios based on the low-level settings and protections set on a system, including in cases where the BIOS is properly write-protected.



- **Attacker Objectives and Impact Analysis** – Review of how attackers can use the ME and system firmware to achieve the greatest possible damage and impact to target organizations.
- **Mitigations and Best Practices** – Key steps that organizations should be taking today to defend their devices from these and similar threats.

About Intel ME and AMT:

As defined by Intel, the Intel® Management Engine (ME) is "*an embedded microcontroller (integrated on some Intel chipsets) running a lightweight microkernel operating system that provides a variety of features and services for Intel® processor–based computer systems*" including out-of-band management services. These remote management capabilities enabled by the ME are known as Intel Active Management Technology (AMT). Thus the ME is the physical controller and AMT is one of the services provided by the ME.

The ME portion of the chipset is conceptually similar to the baseboard management controllers (BMCs) used for out-of-band management of enterprise servers. In addition to having its own kernel, ME has access to its own flash memory stored in the SPI (which also contains the UEFI/BIOS), a dedicated connection to the network interface, and has power that is independent of the operating system. These capabilities allow "*the Intel® Management*

Engine to be up before the main operating system is started," and to "*respond to OOB commands from the IT management console without having to wake up the rest of the system*." And while ME/CSME are conceptually similar to a BMC, it is important to note that these components are integrated into a very wide range of devices, enabling attacks that would be far more scalable and generic than BMC attacks.

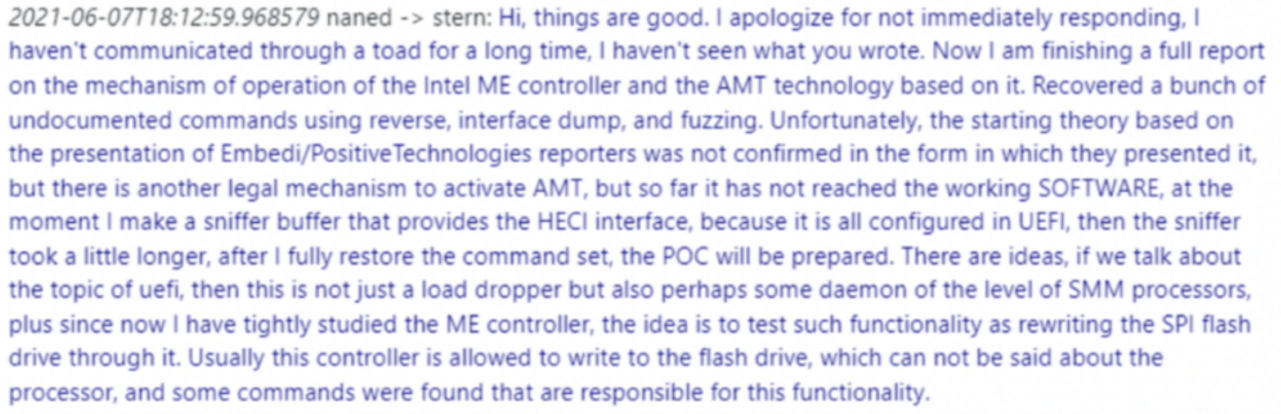
How Adversaries are Targeting Chipset and UEFI Firmware

Analysis of internal Conti communications revealed that attackers were deeply investigating vulnerabilities related to ME firmware as well as BIOS_WP (BIOS Write Protection). This is a significant change in tactics from the most recent firmware threats. For example, well-known firmware threats such as TrickBoot, MosaicRegressor, and LoJax all looked for a well-known vulnerability resulting from the misconfiguration of the BIOS Control register in the SPI controller as a way of writing the SPI and manipulating the UEFI/BIOS firmware of the device. However, this vulnerability is well known and often patched. By shifting focus to Intel ME as well as targeting devices in which the BIOS is write protected, attackers could easily find far more available target devices.

Our analysis found that Conti was focusing research in the following areas.

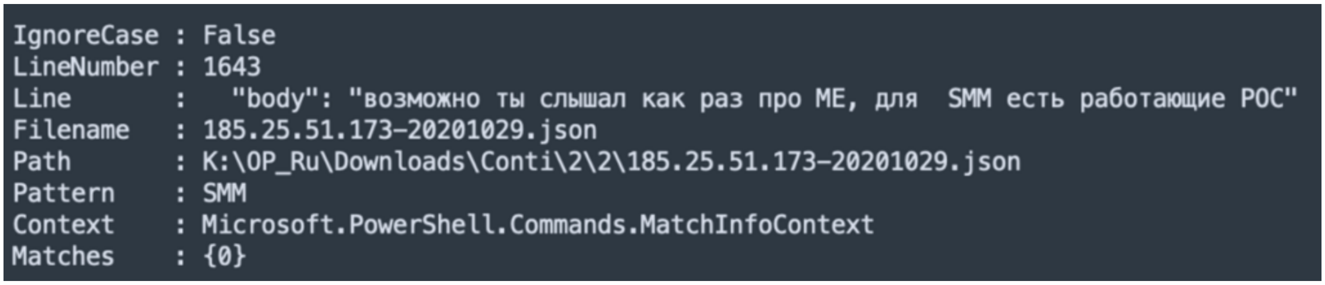
1. Fuzzing the Management Engine Interface and finding undocumented commands and vulnerabilities. Note that the Management Engine Interface (MEI) was previously named the Host Embedded Controller Interface (HECI) and is referred to as HECI within the exposed chats, but both names are still commonly used.
2. Attempting to access SPI (the flash memory used by the UEFI/BIOS system firmware) from the ME in order to generically bypass other protections. Provisioning AMT or changing other ME configurations from the host could expose ME vulnerabilities, leading to code execution. With code execution in the ME, access to SPI flash and other resources can bypass the usual protections.
3. They are considering not only a dropper (placing malware on the host OS) from UEFI but also a System Management Mode (SMM) implant. SMM is a runtime CPU mode controlled by the UEFI/BIOS that is more privileged than the "Ring-0" operating system kernel. The operating system kernel doesn't have the ability to examine SMM code or block it from executing. As a result, an SMM implant could modify the kernel on the fly with complete stealth and without the OS being able to do anything to prevent it.

Excerpts from the chat below illustrate the progression of Conti's investigations and development of proof-of-concept (POC) code.



2021-06-07T18:12:59.968579 naned -> stern: Hi, things are good. I apologize for not immediately responding, I haven't communicated through a toad for a long time, I haven't seen what you wrote. Now I am finishing a full report on the mechanism of operation of the Intel ME controller and the AMT technology based on it. Recovered a bunch of undocumented commands using reverse, interface dump, and fuzzing. Unfortunately, the starting theory based on the presentation of Embedi/PositiveTechnologies reporters was not confirmed in the form in which they presented it, but there is another legal mechanism to activate AMT, but so far it has not reached the working SOFTWARE, at the moment I make a sniffer buffer that provides the HECI interface, because it is all configured in UEFI, then the sniffer took a little longer, after I fully restore the command set, the POC will be prepared. There are ideas, if we talk about the topic of uefi, then this is not just a load dropper but also perhaps some daemon of the level of SMM processors, plus since now I have tightly studied the ME controller, the idea is to test such functionality as rewriting the SPI flash drive through it. Usually this controller is allowed to write to the flash drive, which can not be said about the processor, and some commands were found that are responsible for this functionality.

Translated chat discussing the research and proof-of-concept development of using ME to rewrite flash and gain SMM execution.



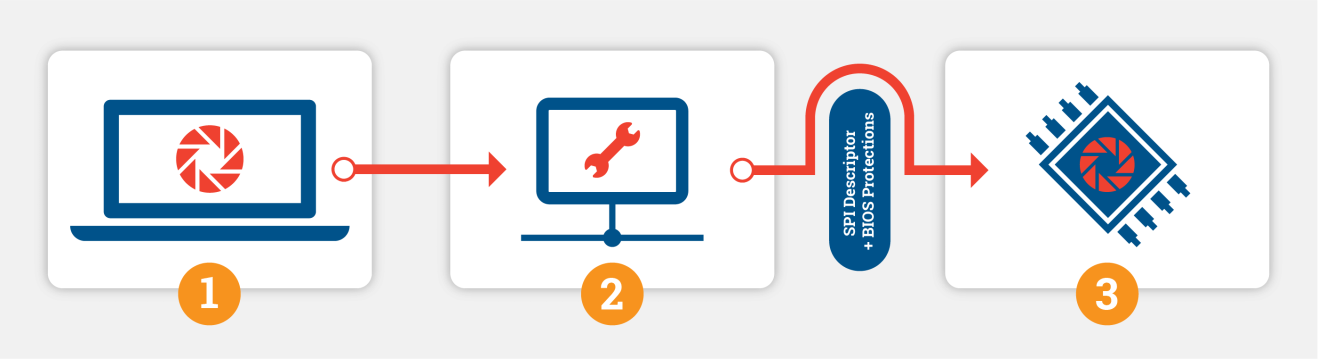
```
IgnoreCase : False
LineNumber : 1643
Line       :   "body": "возможно ты слышал как раз про ME, для  SMM есть работающие POC"
Filename   : 185.25.51.173-20201029.json
Path       : K:\OP_Ru\Downloads\Conti\2\2\185.25.51.173-20201029.json
Pattern    : SMM
Context    : Microsoft.PowerShell.Commands.MatchInfoContext
Matches    : {0}
```

Chats confirming completion of POC for SMM.

Deep Analysis and Threat Modeling

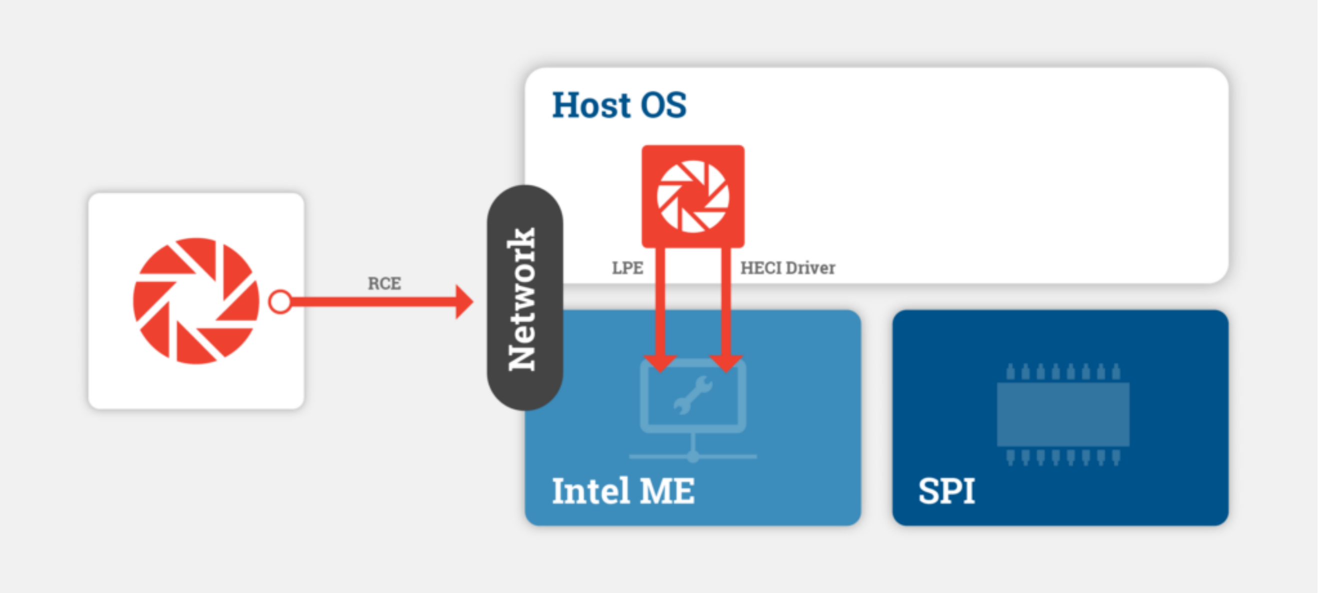
Based on the available insights into attacker goals and methods, the Eclypsium team aimed to dig deeper to understand how a real-world attack would take place. There are several aspects to consider, but at a very high level we can look at an attack as three important phases.

1. **Attacker gains access to a target host** – This can be done via any number of common methods (either local or remote), whether via spear-phishing, exploiting a common OS or application vulnerability, an insider, or during any phase of the distribution, warehousing, or delivery phase of the device's supply chain lifecycle.
2. **Attacker gains control over ME** – This would be done either by a new 0-day vulnerability or via known vulnerabilities that provide remote code execution (RCE) and privilege escalation (PE). Because some of the ME vulnerabilities can be exploited remotely, this can make the previous step optional and an attacker can directly gain control of the ME without first exploiting a vulnerability in the host side of the platform.
3. **Use ME to rewrite UEFI/BIOS or gain SMM Execution** – The ME firmware is inside the UEFI Trusted Computing Base (TCB), which opens the potential for an attacker to infect the UEFI from the ME. Attackers just need to bypass SPI Descriptor and BIOS Control register protections. We will look at multiple scenarios showing how this would work in a real attack.



Mapping the Path to Exploiting ME

First, we need to map out the various ways that an attacker could gain control over the ME firmware. It is important to note that ME has its own dedicated access to the network adapter of the host, which is independent of the host operating system. Attackers such as the PLATINUM group have used this capability in the past to hide command-and-control from OS-level security controls.



Attackers also could exploit the chipset after gaining initial access to the system via virtually any traditional vector such as phishing, malware, or supply chain compromise. Attackers would once again be able target vulnerabilities enabling code execution, but with the advantage of not having to rely solely on vulns that can be exploited over the network. It is important to note that the leaks indicate that Conti engineers are seeking out new ME firmware vulnerabilities. However, we have compiled a list of the known ME and AMT vulnerabilities most likely to be targeted based on their potential to enable remote code execution (RCE) and/or privilege escalation (PE). We have included a table of these vulnerabilities as an appendix here, which includes the relevant Intel security advisories, affected versions, and CVEs.

Attackers could also target ME in a more direct way via phishing by luring a user into executing a malicious ISO as observed in recent Quantum ransomware attacks. This method takes advantage of the HECI driver that provides access to the ME. The HECI driver is installed on any system that has ME and enables privileged users to communicate with ME. Thus an attacker could embed a malicious ISO within an email, gain application execution by tricking the user into opening the file, then run the attack by using HECI to communicate with ME.

Known High-Impact Vulnerabilities in Intel ME

Intel Advisory	Affected Versions	Related CVEs	Notes
Intel-SA-00086	ME – 6.x/7.x/8.x/9.x/10.x//11.0/11.5/11.6/11.7/11.10/11.20	CVE-2017-5705CVE-2017-5711CVE-2017-5712	CVE-2017-5712 is exploitable remotely over the network in conjunction with a valid administrative Intel® Management Engine credential.
Intel-SA-00112	Manageability Engine Firmware version 3.x,4.x,5.x,6.x,7.x,8.x,9.x, 10.x,11.x	CVE-2018-3628	RCE on same subnet
Intel-SA-00118	Management Engine version 11.x	CVE-2018-3627	Execute arbitrary code
Intel-SA-00125	CSME before version 11.21.55	CVE-2018-12147	Local privilege escalation
Intel-SA-00131	CSME before version 11.8.55, 11.11.55, 11.21.55, 12.0.6	CVE-2018-3643	Local execute arbitrary code
Intel-SA-00141	CSME versions before version 12.0.5	CVE-2018-3657CVE-2018-3616	Local execute arbitrary code. Side channel attack to get session key via network.

Intel-SA-00185	CSME before version 11.8.60, 11.11.60, 11.22.60 or 12.0.20	CVE-2018-12196CVE-2018-12200CVE-2018-12190	Local execution of arbitrary code and local privilege escalation.
Intel-SA-00213	CSME before versions 11.8.65, 11.11.65, 11.22.65, 12.0.35	CVE-2019-0091CVE-2019-0086 CVE-2019-0153CVE-2019-0096	Unprivileged user privilege escalation and network privilege escalation.
Intel-SA-00241	AMT versions 11.0 thru 11.8.65, 11.10 thru 11.11.65, 11.20 thru 11.22.65, 12.0 thru 12.0.35, 13.0, 14.0.0	LPE: CVE-2019-11147CVE-2019-11105CVE-2019-11104 CVE-2019-11097CVE-2019-11103CVE-2019-11087CVE-2019-11106CVE-2019-11110CVE-2019-11108 APE: CVE-2019-0169CVE-2019-11088 NPE:CVE-2019-11132CVE-2019-11131CVE-2019-11107	Local privilege escalation. Adjacent privilege escalation, network privilege escalation of unauthenticated user.

Intel-SA-00295	CSME Versions 11.0 through 11.8.76, 11.10 through 11.12.76, 11.20 through 11.22.76, 12.0 through 12.0.63, 13.0 through 13.0.31, 14.0 through 14.0.32, 14.5.11. CSME, AMT, I ISM, DAL and DAL Software before versions 11.8.77, 11.12.77, 11.22.77, 12.0.64, 13.0.32, 14.0.33, 14.5.12	NPE: CVE-2020-0594CVE-2020-0595 LPE: CVE-2020-0586CVE-2020-0542CVE-2020-0533CVE-2020-0541	Network privilege escalation of unauthenticated user, local privilege escalation.
Intel-SA-00307	CSME versions before 12.0.49 (IOT only: 12.0.56), 13.0.21, 14.0.11.	CVE-2019-14598	Local privilege escalation
Intel-SA-00391	CSME and AMT versions before 11.8.82, 11.12.82, 11.22.82, 12.0.70, 13.0.40, 13.30.10, 14.0.45 and 14.5.25	NPE: CVE-2020-8752 LPE: CVE-2020-12297CVE-2020-12303CVE-2020-12354CVE-2020-8744CVE-2020-8757 CVE-2020-8756CVE-2020-8760	Unauthenticated network privilege escalation. Local privilege escalations.
Intel-SA-00404	AMT and ISM versions before 11.8.79, 11.12.79, 11.22.79, 12.0.68 and 14.0.39.	CVE-2020-8758	Local privilege escalation
Intel-SA-00459	CSME versions before 11.8.86, 11.12.86, 11.22.86, 12.0.81, 13.0.47, 13.30.17, 14.1.53, 14.5.32 and 15.0.22	CVE-2020-8703	Local privilege escalation

It is important to note that many systems are vulnerable to CVEs covered in these Intel advisories. For example, a recent analysis of a production network found that 72.3% of devices were vulnerable to CVEs in Intel SA00391, which contains the potential for network privilege escalation. Likewise, 61.45% of devices were vulnerable to issues covered in SA00295, which also enables privilege escalation over a network. These two security advisories include vulnerabilities from the Ripple20 disclosure and additional remotely-exploitable vulnerabilities in the Treck TCP/IP stack found by Intel as a followup to the initial Ripple20 disclosure.

Moving From ME to UEFI/SMM

ME is a highly privileged component within the system, and has its own flash memory within the SPI. An attacker with control over the ME can then use that access to overwrite the UEFI system firmware and gain SMM code execution. The details of how this is done will vary depending on the types of protections and settings of the target system. Two of the most important settings in this regard is if BIOS write protection (BIOS_WP) is properly set on the device, and if ME firmware has the privileges to modify different SPI regions in the access control table within the SPI Descriptor.

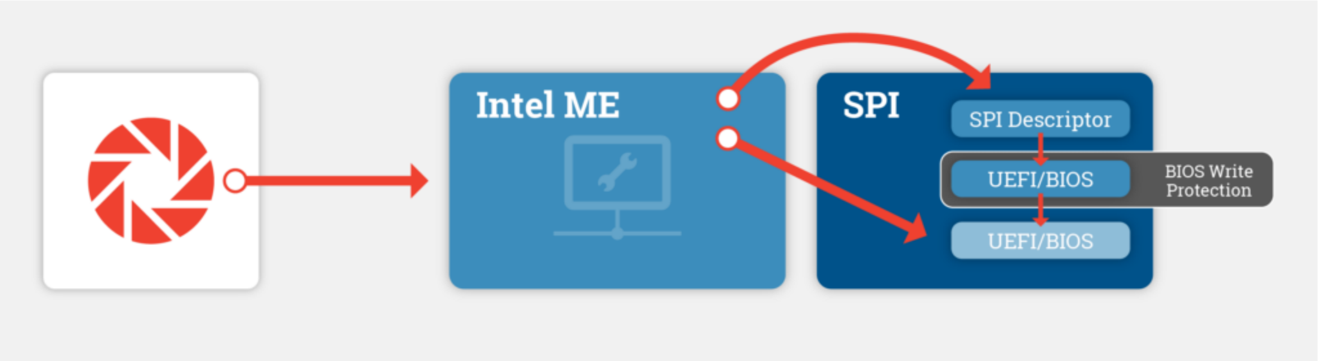
We will look at three scenarios in more detail.

Scenario #1

System State:

- ME has access to overwrite the SPI Descriptor in the SPI Descriptor access control table
- SPI Controller BIOS Write Protections are properly configured
- 51% of firmware update images with SPI descriptor had this level of access

By using ME to modify the SPI Descriptor, an attacker could change the BIOS Region layout, thus moving the BIOS outside the area protected by BIOS_WP. The attacker could then modify the firmware to install their own malicious implant code.



Attacker uses ME to move BIOS region outside of Write Protection region and inserts implant into firmware

Scenario #2

System State:

- ME has access to BIOS Region in SPI Descriptor access control table
- SPI Controller BIOS Write Protections are properly configured
- 52% of firmware update images with SPI descriptor had this level of access

Many systems have the SPI descriptor configured by the OEM to grant the ME write permission to the BIOS region. This could be used to allow the ME to write to the BIOS region in the SPI flash to modify UEFI firmware. In the case that the SPI controller BIOS Write Protections are configured and prevent writes to the BIOS region even by the ME, the ME can trigger the CPU to enter SMM mode and issue SPI transactions while the CPU is running in SMM. The SPI controller is designed to allow the BIOS region to be updated even when the BIOS Control register is configured properly when the CPU is in SMM.

Scenario #3

System State:

- ME does not have access to BIOS Region in SPI Descriptor access control table
- SPI Controller BIOS Write Protections are properly configured

Another useful capability of the ME is the ability to reboot the host CPU and force it to boot from virtual media. This is intended to be used to (re)install operating systems and other maintenance tasks. The way that this is implemented is that the ME can trigger a Host Partition Reset which causes the host CPU to be reset and PLTRST# to be asserted. When this happens, several PCH protection mechanisms are unlocked, such as the BIOS Control register in the SPI controller.

This could be used by a compromised ME to use its DMA engine to inject code into the host processor DRAM to get arbitrary code execution at a point when SMRAM and the SPI protections have not been locked yet. This would allow the ME to write to SPI flash even when the SPI descriptor and BIOS Control register protections are implemented correctly. This could also be used to install malicious SMI handlers before SMRAM is locked to leave an undetectable rootkit running while the operating system loads and runs, though we did not develop proof of concept code to confirm this.

Attacker Objectives and Impact Analysis

It is important to understand why attackers are investing resources in developing these new paths to a system's firmware. Control over firmware gives attackers virtually unmatched powers both to directly cause damage and to enable other long-term strategic goals.

Destruction of Assets

In terms of damage, an attacker can effectively "brick" a system permanently by overwriting the system firmware. Similarly, an attacker could use this level of access to wipe the Master Boot Record or other high-value files on a system. Wipers such as WhisperGate and HermeticWiper have played a major and ongoing role in the Russian invasion of Ukraine and provide a stark reminder of the damaging potential of low-level attacks on devices. While such low level wiper attacks have averaged about one major event per year, in the first quarter of 2022, there have been 6 or more wipes discovered in the wild. Attacker motives are shifting to destructive objectives, whether APTs, ransomware, or high profile criminal espionage actors such as LAPSUS$.

Persistence

However, firmware also provides long-term persistence capabilities that would be particularly valuable to a group such as Conti. Attackers are able to use the unique privileges of firmware to evade a wide variety of security features, and security products in order to establish ongoing persistence on a device. Groups like Conti directly monetize such persistence by reselling access to other threat actors, or even dropping additional ransomware payloads at a later date.

Evasion of AV/EDR/XDR Products

This ability to maintain persistence is tied to the ability for an attacker's code to reside off of the traditional system storage drives, and most importantly, the ability to evade security controls. By sitting below the operating system, attackers can hide in areas where traditional security tools lack visibility. Antivirus and EDR have very limited coverage of firmware threats and lack the ability to proactively verify the integrity of a device's firmware. This makes it relatively easy for attackers to evade detection. Additionally, traditional products lack visibility into components outside of UEFI such as ME firmware. This ability to shift the attack to areas that security tools fail to protect gives attackers incredible advantages.

Even when antivirus, EDR, and other endpoint security tools do query a system's firmware, they typically rely on the operating system in order to do so. However, if the firmware is already compromised, a malicious firmware implant can easily report false information to the operating system in order to evade those security controls. Such techniques were recently observed in the wild. Likewise, the use of SMM allows attackers to execute code that is completely invisible to the operating system, and likewise, the security products running within the OS.

The focus on evading security tools was confirmed in the chats, which showed the group was actively acquiring endpoint security tools for developing evasion techniques and testing. Additional chats confirmed that the actors were able to bypass up to eight well-known EDR products, while struggling to evade at least three of them. Indeed, the driving motive behind exploring ways to attack and access the UEFI, is to be able to disable any of the overlying security controls and third party security stack that resides at the OS layer.

Evasion of Device Protections

In addition to evading security products, attackers can use compromised firmware in order to evade many of the protections built into modern systems. This includes features such as BitLocker, Windows Virtual Secure Mode (VSM), Credential Guard, and Early Launch AntiMalware (ELAM).

Mitigations and Best Practices

Based on the information disclosed in the Conti leaks, organizations should take action to reduce their exposure and develop capabilities to detect and respond to these new techniques. There are several key steps that we recommend:

1. **Scan Devices for Exploitable Versions of ME** – We have provided a list of the known CVEs most likely to be used in an attack. Device scans should include CVEs as well as detecting versions of firmware that gives ME access to the BIOS Region in the SPI Descriptor access control table. Organizations should use a tool that specializes in firmware vulnerabilities (e.g. CHIPSEC, Eclypsium), as many traditional vulnerability scanners lack the necessary drivers and access to scan down to the level of ME.
2. **Monitor ME for Any Configuration Changes** – Organizations should check to verify that the ME firmware on their devices matches a valid, known version of firmware from Intel. Any changes would indicate that the firmware may have been compromised or tampered with. Ideally, organizations should verify this information using mechanisms that are independent of the operating system.
3. **Verify the Integrity of the SPI Flash and Monitor for Any Configuration Changes** – The same integrity checking and monitoring should be applied to the SPI flash firmware as well as the UEFI/BIOS of the device. Once again, firmware should match known good versions, and teams should be alerted to any changes particularly tied to anomalous or unreleased code.

Conclusions

The recent Conti leaks mark a critical phase in the rapidly evolving role of firmware in modern attacks. Threats such as TrickBoot, MosaicRegressor, and dozens of new forms of wiper malware have continued to drive attacks below the level of the operating system. However, the Conti leaks exposed a strategic shift that moves firmware attacks even further away from the prying eyes of traditional security tools. The shift to ME firmware gives attackers a far larger pool of potential victims to attack, and a new avenue to reaching the most privileged code and execution modes available on modern systems.

As the realities of the threat landscape continue to evolve, it is critical that organizations continue to inform their defenses based on the latest intelligence available. The Eclypsium team will continue to strive to provide updated insight and guidance into these and other firmware threats as they become available. For any questions related to this research, please contact the Eclypsium team at info@eclypsium.com.