OpenBMC Security in Practice

Vulnerabilities, Variants, and AMI MegaRAC OneTree™

EXECUTIVE SUMMARY_

Baseboard Management Controllers (BMCs) are critical components in server platforms, enabling out-of-band management and presenting a significant attack surface for threat actors. This paper examines the security landscape of the Linux Foundation's OpenBMC, the most widely adopted open-source BMC firmware, highlighting its flexibility, widespread use, and the complexity this brings to vulnerability management. The analysis underscores the challenges of tracking vulnerabilities across diverse implementations by reviewing recent and historical CVEs affecting the Linux Foundation's OpenBMC project and major vendor forks, including IBM, Intel, and SuperMicro. The document also evaluates vendor patch practices, highlighting AMI MegaRAC OneTree as the optimal solution. It emphasizes the importance of maintaining a robust Software Bill of Materials (SBOM) to manage supply chain risk. Ultimately, the findings stress that while open-source BMCs offer adaptability, their security depends on proactive vulnerability management, regular updates, and transparency throughout the supply chain.

THE BMC LANDSCAPE_

BMCs are specialized SoCs (System-on-a-Chip) integrated into server platforms to provide out-of-band remote management and control of select functionality. Specifically, BMCs allow system administrators to monitor and control systems independent of the operating system. For example, system administrators can monitor temperature, control fan speeds, and remotely turn systems on or off. The BMC operates independently and draws power so long as the power supply is turned on, to control the system even when power to the rest of the server has been turned off. This enables remote restoration and maintenance even when something goes wrong with the primary operating system.

THREATS AGAINST BMCs_

Given this critical degree of control, a compromise of the BMC is a prize for APTs, ransomware, and other malicious actors. Thus, BMC security must be of paramount importance within any given enterprise. Havoc can be unleashed if vulnerabilities such as Denial-of-Service (or DoS) attacks, Remote Code Execution (or RCE), and general Supply Chain issues (more on this later) are discovered and not patched.

Research, like that presented by Dan Farmer and described further by HD Moore in 2013, started to make clear that this little-known and extremely powerful component can be a serious vector for attacks. Additional research presented in 2018 by Airbus and Synactiv, and the infamous Pantsdown vulnerability, brought this to the forefront again, a few years later. Of course, Eclypsium itself has published numerous demonstrations and vulnerabilities related to BMCs. These vulnerabilities demonstrated an attack surface from remote attacks against externally exposed BMC interfaces and local attacks that elevate privilege from malware running on the host server.

This culminated in evidence of BMC-resident malware in the form of iLoBleed and ransomware around the same time, even leading the government agencies to (repeatedly) warn administrators to harden BMCs against attack. Now, attacks targeting the BMC are a part of our everyday world.

INDUSTRY EFFORTS IN BMC SECURITY_

OpenBMC has emerged as a common codebase organized around the goal of interoperability among heterogeneous OEM servers. The GitHub project has over 900 forks and contributions from 72 companies as of early 2025. The project's flexibility allows for extensive customization, enabling developers and manufacturers to tailor the firmware to specific hardware requirements and integrate new features. This adaptability has led to widespread adoption across various sectors: enterprise data centers, high-performance computing environments (such as AI-focused server pools), telecommunications suppliers, and cloud service providers (ubiquitous with modern applications and scalability).

Due to the number of iterations and customization, it can be challenging to discern the impact of vulnerability across the various OpenBMC flavors. For example, if there is an upstream dependency for one OpenSSL version that is used across multiple OpenBMC builds and there is not a direct mention of the vulnerable library being used in half of the vendors are they truly vulnerable due to that inheritance or is it a false positive vulnerable because that library isn't mentioned?

As a result, OpenBMC vulnerabilities can be problematic and quickly become a bigger mess to manage. This lends to individuals wanting to stick to one of the "primary" versions. Another factor that can be considered is that when someone uses OpenBMC, they typically do not have any support, customizations, or enhancements unless they use a paid version.

Considering the previous statements, let's take a peek at some known vulnerabilities and their impacts on OpenBMC:

Main OpenBMC Project

(Maintained by The Linux Foundation)

1. **CVE-2024-41660** - slpd-lite is a unicast SLP UDP server. Any OpenBMC system that includes the slpd-lite package is impacted. Installing this package is the default when building OpenBMC. Nefarious users can send SLP packets to the BMC using UDP port 427 to cause memory overflow issues within the slpd-lite daemon on the BMC. Patches will be available in the latest OpenBMC/slpd-lite repository.

2. **CVE-2022-3409** - A vulnerability in bmcweb of the OpenBMC Project allows a user to cause a denial of service. This vulnerability was identified during mitigation for CVE-2022-2809. When fuzzing the multipart_parser code using AFL++ with address sanitizer enabled to find the smallest memory

corruptions possible, it detected a problem in how the multipart_parser handles unclosed HTTP headers. If a long enough HTTP header is passed in the multipart form without a colon, there is one byte overwrite on the heap. It can be conducted multiple times in a loop to cause DoS.

3. **CVE-2022-2809** - A vulnerability in bmcweb of the OpenBMC Project allows a user to cause a denial of service. When fuzzing the multipart_parser code using AFL++ with address sanitizer enabled to find the smallest memory corruptions possible, it detected a problem in how the multipart_parser handles unclosed HTTP headers. If a long enough HTTP header is passed in the multipart form without a colon, there is one byte overwrite on the heap. It can be conducted multiple times in a loop to cause DoS.

4. **CVE-2021-39296** - In OpenBMC 2.9, crafted IPMI messages allow an attacker to bypass authentication and gain complete control of the system.

5. **CVE-2021-39295** - In OpenBMC 2.9, crafted IPMI messages allow an attacker to cause a denial of service to the BMC via the netipmid (IPMI LAN+) interface.

6. **CVE-2020-14156** - user_channel/passwd_mgr.cpp in OpenBMC phosphor-host-ipmid before 2020-04-03 does not ensure that /etc/ipmi-pass has strong file permissions.

IBM OpenBMC

1. **CVE-2024-35124** - A vulnerability in the combination of the OpenBMC's FW1050.00 through FW1050.10, FW1030.00 through FW1030.50, and FW1020.00 through FW1020.60 default password and session management allows an attacker to gain administrative access to the BMC. IBM X-Force ID: 290674.

2. **CVE-2024-31916** - IBM OpenBMC FW1050.00 through FW1050.10 BMCWeb HTTPS server component could disclose sensitive URI content to an unauthorized actor that bypasses authentication channels. IBM X-Force ID: 290026.

Intel OpenBMC

1. **CVE-2022-35729** - An out-of-bounds read in firmware for OpenBMC in some Intel® platforms before version 0.72 may allow an unauthenticated user to potentially enable a denial of service via network access.

2. **CVE-2022-29494** - Improper input validation in firmware for OpenBMC in some Intel platforms before versions egs-0.91-179 and bhs-04-45 may allow an authenticated user to potentially enable denial of service via network access.

3. **CVE-2023-35123** - Uncaught exception in OpenBMC Firmware for some Intel Server Platforms before versions egs-1.14-0, bhs-0.27 may allow an authenticated user to potentially enable denial of service via network access.

4. **CVE-2023-49144** - Out of bounds read in OpenBMC Firmware for some Intel Server Platforms before versions egs-1.15-0, bhs-0.27 may allow a privileged user to enable information disclosure via local access.

5. **CVE-2025-20097** - In February 2025, a vulnerability was discovered in Intel's OpenBMC firmware for certain server families. It allows an authenticated user to cause a denial of service through an uncaught exception.

SuperMicro OpenBMC

1. **CVE-2024-10239** - A security issue in the firmware image verification implementation at Supermicro MBD-X12DPG-OA6. An attacker with administrator privileges can upload a specially crafted image, which can cause a stack overflow due to the unchecked fat→fsd.max_fld.

2. **CVE-2024-10238** -A security issue in the firmware image verification implementation at Supermicro MBD-X12DPG-OA6. An attacker can upload a specially crafted image that will cause a stack overflow by not checking fld→used_bytes.

3. **CVE-2024-10237** - There is a vulnerability in the BMC firmware image authentication design at Supermicro MBD-X12DPG-OA6. An attacker can modify the firmware to bypass BMC inspection and bypass the signature verification process.

4. **SuperMicro (CVEDETAILS)** - The CVEs listed above are relatively new and issued after SuperMicro became a CNA in the CVE program. The resource here lists CVEs issued before SuperMicro became a CNA and includes over 20 CVEs issued for BMC and BMC-related products.

Other Forks:

a. Specific CVEs for LibreBMC and Firmware-action haven't shown up in simple public searches, but that doesn't mean that they have not had security issues; it merely means that they could not be documented publicly and were hopefully resolved by internal teams without much fanfare.

Remembering that security issues in the Linux Foundation's OpenBMC project may affect its forks and implementations unless patched, as we stated previously. There are challenges when determining if the current build contains vulnerable software components. The OpenBMC community actively works to address security concerns as issues are submitted or discovered. This is evidenced by establishing security response team guidelines. Progress that has been made can be implemented by private builds. This is partially due to private builds supporting specific customer fields and needs.

ANALYSIS OF IMPLEMENTATIONS_

Working closely with AMI, Eclypsium has reviewed multiple OpenBMC derivatives to help end users understand the security of these implementations. This includes the Linux Foundation's OpenBMC project, Intel's S2600 server, and AMI MegaRAC OneTree-2.0 implementations.

SOFTWARE COMPONENTS_

A strong lifecycle maintenance plan is required when using open source in a product. This includes updating to the latest version to fix bugs and close security issues. One way to examine the software components of a complex system like OpenBMC is a Software Bill of Materials (SBOM), a comprehensive inventory of all software components, dependencies, and metadata within a codebase. It provides critical details such as component versions, licenses, and patch status, enabling organizations using said software to identify security vulnerabilities and compliance risks. SBOMs have become essential in software development to enhance transparency and manage supply chain risks, especially in light of high-profile breaches and widespread issues. OpenBMC's ecosystem is inherently complex (due to its being built via YOCTO and Bitbake from various tools and applications). Like any other complex system, it is susceptible to supply chain vulnerabilities and threats. By maintaining an SBOM, developers can track dependencies. With an SBOM, end users can more rapidly respond to mitigate vulnerabilities when discovered by having an inventory of what is in use (For example, YOCTO supports SBOMs natively).

Every developer involved in the supply chain builds upon some existing blocks, and each block adds another set of potential security issues. The above is not even an exhaustive analysis of software components. Instead, we have focused on the most common and impactful components. While no implementation is perfect, we can see a vast difference between implementations that are expending significant effort to stay up to date and implementations that are not. A simple metric is to review when the most recent official release was made.

Latest Release				
Main GitHub	**Intel S2600WF**	**Dell R670csp**	**Supermicro X14DBG-AP**	**AMI MegaRAC OneTree-2.0**
May 16, 2023	Jan 3, 2024	Dec 11, 2024	Dec 16, 2024	Jan 9, 2025

AMI applies patches to the OpenBMC build and maintains a quarterly release cadence. This includes backporting patches to mitigate risk and vulnerabilities.

For further analysis, we pick specific software components common to OpenBMC implementations and evaluate which are up to date and the corresponding security exposure. The tables below list these specific analyzed components.

Methodology note: Vulnerability counts are derived from CVEs publicly associated with a given software version across all assessed products. This does not account for patches backported to an older version, which likely reduce the overall vulnerability count in practice.

Linux Kernel - Latest stable versions are 6.13.5, 6.12.17, 6.6.80, 6.1.129, 5.15.178, 5.10.234, 5.4.290

	Main GitHub	Intel S2600WF	Dell R670csp	Supermicro X14DBG-AP	AMI MegaRAC OneTree-2.0
Observed Version	**meta-hpe: 5.14.0** *Released Aug 2021* **meta-raspberrypi: 6.1.77** *Released Feb 2024* **meta-ibm: 4.17.18** *Released Aug 2018*	**4.9.37** *Released Jul 2017*	**5.10.161** *Released Dec 2022*	**5.15.0** *Released Oct 2021*	**6.6.47** *Released Aug 2024*
Security Issues	Up to **2844** CVEs potentially allowing: *code execution, bypass, privilege escalation, DoS, and information leak.*	Up to **1961** CVEs potentially allowing: *code execution, bypass, privilege escalation, DoS, and information leak.*	Up to **2192** CVEs potentially allowing: *code execution, bypass, privilege escalation, DoS, and information leak.*	Up to **2920** CVEs potentially allowing: *code execution, bypass, privilege escalation, DoS, and information leak.*	Up to **1320** CVEs potentially allowing: privilege escalation, DoS, and information leak.
Exposure	**Network Stack, Local shell or SSH**				

Busybox - Latest stable version is 1.36.1					
	Main GitHub	**Intel S2600WF**	**Dell R670csp**	**Supermicro X14DBG-AP**	**AMI MegaRAC OneTree-2.0**
Observed Version	**1.37.0** *Released Sep 2024*	**1.36.1** *Released May 2023*	**1.36.0** *Released Jan 2023*	**1.34.0** *Released Aug 2021*	**1.36.1** *Released May 2023*
Security Issues	Unstable development version	CVE-2023-42363, CVE-2023-42364, CVE-2023-42365, CVE-2023-42366	CVE-2023-42363, CVE-2023-42364, CVE-2023-42365, CVE-2023-42366	CVE-2022-28391 (other vulnerable modules not included)	CVE-2023-42363, CVE-2023-42364, CVE-2023-42365, CVE-2023-42366
Exposure	**Local shell or SSH**				

OpenSSL - Current versions are 3.5.0, 3.4.1, 3.3.3, 3.2.4, 3.0.16					
	Main GitHub	**Intel S2600WF**	**Dell R670csp**	**Supermicro X14DBG-AP**	**AMI MegaRAC OneTree-2.0**
Observed Version	**3.4.0** *Released Oct 2024*	**1.1.1.w** *Released Sept 2023 (out of support)*	**3.1.5-r0** *Released Jan 2024*	**1.1.1.g** *Released Apr 2020 (out of support)*	**3.4.0** *Released Oct 2024*
Security Issues	**CVE-2024-12797** (TLS session handling), **CVE-2024-13176** (side channel)	**6 CVEs** related to memory corruption in certificate handling.	**7 CVEs** related to DoS and memory corruption.	**29 CVEs** related including the 6 CVEs in 1.1.1.w.	**CVE-2024-12797** (TLS session handling), **CVE-2024-13176** (side channel)
Exposure	**SSL/TLS handling, Certificate validation**				

SECURE BOOT AND ROOT OF TRUST_

Most implementations based on the ASpeed SoC parts will support (and, for production builds, enable) the hardware-based secure boot mechanism built into these processors. These hardware root-of-trust solutions are based on fuses corresponding to RSA keys, which are used to verify a signature on the firmware image. This makes persistent code changes more difficult for attackers, leaving them to focus mainly on the above runtime software exploitation issues and any other handling of untrusted inputs. The newer AST2700 supports a newer secure boot implementation led by the Open Compute Project (OCP) known as Caliptra.

As of 2019, the Linux Foundation's OpenBMC GitHub did not support ASpeed secure boot. However, ASpeed submitted patches for inclusion into the Linux kernel in 2021. Even with support, the default appears to leave secure boot disabled, which is normal for development and debugging. (You wouldn't want to brick your development board.) However, all implementations must carefully examine their production settings to ensure secure boot is enabled with secure keys. AMI MegaRAC OneTree includes Secure Boot support, strengthening the security posture of the platform and ensuring the integrity of the system firmware during the boot process.

Mg MegaRAC OneTree™

AMI MegaRAC OneTree_

Going above and beyond

Working with AMI, we learned about some unique aspects of their OneTree implementation that merit attention.

2FA based on TOTP

OneTree has two-factor authentication support based on the TOTP protocol, which is used in apps like Google Authenticator. When checking other vendors, either 2FA support is missing entirely, or it is based on other protocols, such as SMTP (found in HPE iLO and Dell iDRAC). Google Authenticator is a step up from different vendors' 2FA mechanisms, which rely on insecure communications such as email or SMS. For example, email might travel unencrypted over the internet, and attackers or nation-states might intercept SMS.

RADIUS authentication support

AMI supports the RADIUS protocol implemented in OneTree. Notably, it does not have a problem revealing the RADIUS secret on API GET requests, which might allow attackers with MITM listener capabilities to intercept later login requests for other users, which is sometimes found in different RADIUS implementations.

Client implementation uses FreeRADIUS underneath, and as such is not affected by Blast-RADIUS (FreeRADIUS Security).

Encrypted backups

OneTree has an implementation for backup/restore, which encrypts the entire backup file as part of the backup process.

SNMP support

By default, OpenBMC only supports sending SNMP traps for error log entries. AMI MegaRAC OneTree contains an implementation that exposes sensor and user data as SNMP tables, reducing the vendor's burden.

Extra non-security features

AMI MegaRAC OneTree contains extra features compared to the Linux Foundation's OpenBMC implementation. These include NVME support, RAID MSCC and BRCM, Intel ASD, and others.

Backported fixes

Fixes from the latest Linux Foundation's OpenBMC branch regularly find their way into OneTree. The OpenBMC bugfixes on the master branch are also included in OneTree and include security fixes.

Tight unauthenticated API surface

Based on OpenBMC, the unauthenticated API surface is reasonably tight, and only a few web APIs are exposed to an unauthenticated user (typically, the endpoints required for user login). OneTree tries to follow this and not introduce unauthenticated APIs into its code, making exploitation difficult.

CONCLUSION_

As we have seen, there are numerous cybersecurity considerations for BMCs in your organization. No matter what role you play, these critical components affect you. OpenBMC-based implementations are becoming more common, and it is helpful to keep the following key considerations in mind:

1. If your organization manages servers internally, you have BMCs. Do you know the version and configuration of these critical devices? Are they appropriately isolated? To maximize the benefits of out-of-band management, ensure you use the latest BMC firmware version. Consider how frequently your supplier keeps up with the latest OpenBMC releases and security patches. Does this meet your security goals?

2. If your role requires you to purchase servers, look for well-known vendors that provide preconfigured options and sustained support. Utilizing a vendor's version of OpenBMC allows for easier update management as it offloads a degree of asset management when it comes to the SBOMs associated with a customer build. Evaluate the total cost of ownership, including development resources for customization and ongoing maintenance, to determine what approach to OpenBMC aligns with your organization's strategy.

3. If you perform a cybersecurity role, ensure your team and processes are aware of the OpenBMC's presence in your organization and its critical role in server management. Regularly audit your systems for vulnerabilities, missing firmware updates, robust access controls, and unauthorized changes.

4. Consider whether SBOMs are available for BMCs and other components of your devices. Incorporate such information into asset management practices, such as tracking BMC deployments and monitoring for unauthorized changes.

The complexity of modern technology supply chains makes it difficult to understand everything that goes into your equipment. Eclypsium is always working with partners and suppliers to give you better visibility and help you translate this into actionable next steps. We are grateful to AMI for the transparency they have offered in this analysis and in constantly working to secure core components like BMCs.