Mitigate the GRUB2 BootHole Vulnerability

Summary

Eclypsium® has disclosed a vulnerability, CVE-2020-10713 also known as BootHole [1], in the Grand Unified Bootloader (GRUB2) that is widely used to boot Linux®-based operating systems. The vulnerability is triggered by modifying a GRUB2 configuration file to force a buffer overflow allowing arbitrary code execution. The vulnerability allows malicious actors to subvert the boot process and introduce untrusted code, commonly referred to as bootkits, enabling highly effective persistence on an endpoint. The vulnerability can also be used to bypass Unified Extensible Firmware Interface (UEFI) Secure Boot validation mechanisms allowing malicious code to execute prior to loading the operating system.

Exploitation of the BootHole vulnerability requires modifying a configuration file that is parsed by a vulnerable version of GRUB2. The configuration file is not commonly signed, measured at boot time, or encrypted. Physical access or administrator privilege is required to exploit the vulnerability and subvert the boot process. Use of hardware-based full disk encryption may mitigate some threats from physical access.

A boot component known as Shim is used to load GRUB2 when Secure Boot is enabled. Linux distributions embed their distribution specific key in Shim which allows loading of bootloaders, such as GRUB2, signed by the distribution's specific key. Many versions of Shim have been signed by the Microsoft Third Party UEFI Certificate Authority (CA) and the Microsoft CA is trusted by many modern business and consumer computers. The BootHole vulnerability is not in Shim, but the trust of the Shim binaries must be revoked in order to prevent loading of vulnerable versions of GRUB2 that are allowed by existing signed versions of Shim. Eclypsium and industry members have identified many vulnerable GRUB2 binaries as well as many Shim binaries that will load a vulnerable version of GRUB2.

If an endpoint does not have Secure Boot enabled, then the endpoint is already vulnerable to boot malware regardless of the GRUB2 vulnerability being present. Linux-based endpoints may not commonly have Secure Boot enabled. Software that is not maintained by a distribution, such as commercial security software, may have unsigned kernel modules that prevent adoption of Secure Boot unless local administrators sign the kernel modules. While all affected endpoints should be mitigated, prioritization should be given to endpoints that have Secure Boot enabled, use software-based full disk encryption, or whose owners cannot maintain exclusive physical control due to being mobile devices (e.g. laptops and tablets).

Additional GRUB2 vulnerabilities (CVE-2020-14308, CVE-2020-14309, CVE-2020-14310, CVE-2020-14311, CVE-2020-15705, CVE-2020-15706, CVE-2020-15707, CVE-2020-7205) and Linux kernel vulnerabilities (CVE-2019-20908, CVE-2020-15780) have been discovered by industry members while investigating the BootHole vulnerability. Exploitation of the vulnerabilities may also allow bypass of Secure Boot as well as compromise of boot components and must be patched.

Mitigation Actions

Administrators may choose from two different options to mitigate the GRUB2 vulnerability. The standard mitigation option involves updating an endpoint's vulnerable boot components and revoking the trust of existing boot components. The standard mitigation is best suited for typical consumer, business, and enterprise environments.

A more advanced mitigation involves implementing a Secure Boot trust infrastructure and customizing the endpoint's Secure Boot implementation to use that trust infrastructure. The advanced mitigation option is best suited for endpoints that have higher security and integrity requirements.

Standard Mitigation: Update and Revoke Trust of Vulnerable Boot Components

Linux-based endpoints require updating their boot components and revoking the trust in older boot components. Windows® endpoints only require revoking trust if the endpoint's firmware trusts the specific CA identified in Microsoft's advisory [2]. Fully mitigating the BootHole vulnerability requires multiple steps that must be performed in a specific order

to update and revoke the trust for existing signed boot components. **Failure to ensure each step is completed before proceeding to the next step may result in an endpoint no longer being able to boot while Secure Boot is enabled**.

The steps in this section provide a general outline of the mitigation process. Refer to operating system, virtualization software, and Original Equipment Manufacturer (OEM) vendor advisories linked from Eclypsium's advisory to ensure that all vendor specific steps are taken into account before implementing the mitigations.

Step 1: Update Boot Components

Linux-based endpoints must have their boot components updated with patches issued by distribution maintainers. Updating the boot components must be performed before any other mitigation steps, otherwise later steps could render the endpoint unbootable while Secure Boot is enabled. Windows endpoints do not need their boot components updated since Windows uses different boot components than Linux endpoints and Windows boot components are signed by a different certificate authority. Updating the endpoint's firmware is also recommended as some OEMs require that a firmware update be applied before revoking trust in older boot components

In addition to traditional desktop, server, laptop, and tablet form factors, updates may need to be applied to network devices and appliances (e.g. firewalls, web proxies, email gateways which commonly use Linux), installation and recovery media, dual/multi boot endpoints, virtualization software, and cloud vendor provided boot images. Note that dual/multi boot endpoints may need to be updated multiple times to ensure each operating system instance has updated boot components.

Updating the endpoint's boot components patches the vulnerable code, but endpoints are still vulnerable to exploitation without taking additional action to revoke the trust of existing signed boot components that load vulnerable versions of GRUB2. Malicious actors could roll back patches or leverage existing boot components to compromise the endpoint until trust is revoked.

Step 2: Test Boot Component Trust Revocation

The UEFI Forbidden Signatures Database (DBX) contains hashes and certificates used to revoke trust of boot components. The firmware will not execute the boot component when a hash in the DBX matches the hash of a boot component or when a certificate in the DBX matches the certificate that validates the signature of a boot component. For the BootHole vulnerability, the DBX needs to be updated to block all the existing signed versions of Shim that can load a vulnerable version of GRUB2. The DBX can be updated by installing vendor issued patches that update the DBX or by running operating system specific tools or commands that apply the UEFI Forum's UEFI Revocation List File [3] to the DBX. Refer to operating system vendor documentation, linked from Eclypsium's advisory, for information on how to apply the DBX updates using operating system specific tools.

Windows endpoints only require revoking trust if the endpoint's firmware trusts the specific CA identified in Microsoft's advisory [2]. Microsoft's knowledge base article [4] indicates that patches that apply the DBX update may not be available via Windows Update until sometime in 2021.

Some OEM's models have experienced problems when adding a large amount of entries to the DBX. Some endpoints may become unbootable if too many DBX entries are added. When the problem exists, the problem typically exists in all firmware revisions up to a specific firmware revision. Testing all different firmware revisions per model used in an enterprise is crucial to ensuring that endpoints do not become unbootable when Secure Boot is enabled.

Administrators may find that a more straightforward testing approach is to update all endpoints to the latest available firmware revision before applying the UEFI Revocation List File, or applying a vendor issued patch, to update the DBX. Some OEMs require a firmware update before applying the DBX update. Refer to OEM advisories, linked from Eclypsium's advisory, for additional information and instructions.

Unforeseen problems may arise from applying a DBX update, as happened earlier this year [5], due to the diversity of the UEFI ecosystem. Windows endpoints that use BitLocker® for full disk encryption, but have not had BitLocker suspended before applying a firmware update or a Secure Boot configuration change such as a DBX update, will need the BitLocker

Recovery Key to successfully boot. Administrators must ensure a selection of endpoints are tested that fully represent the diversity of endpoints in the enterprise that includes all OEMs, models, and firmware revisions.

Step 3: Apply Boot Component Trust Revocation

Apply the DBX updates across the enterprise once all endpoint boot components have been updated and the DBX updates have been tested on a representative population of endpoints. Remember, if the DBX updates are applied before updating the endpoint's boot components, then the boot components will no longer be trusted to execute by the firmware and could render the endpoint unable to boot while Secure Boot is enabled.

Advanced Mitigation: Implement Custom UEFI Secure Boot Trust

Business and enterprise class models may allow Secure Boot to be customized so that Microsoft and system vendor Secure Boot certificates can be minimized or removed. Secure Boot customization requires that an enterprise operate their own certificate infrastructure. UEFI Option ROMs, boot binaries, kernels, kernel modules, and drivers may all need to be signed using the enterprise's certificate infrastructure. Customization involves the largest administrative overhead but provides immediate and complete mitigation of the GRUB2 vulnerability. Secure Boot customization is best suited for endpoints that have higher security and integrity requirements. See the upcoming NSA cybersecurity technical report "UEFI Secure Boot Customization" for more information and instructions [6].

Detection Guidance

Monitoring for changes to firmware, firmware configuration, and boot components is recommended due to the amount of time that may be required to perform effective testing before all mitigations can be applied to all endpoints. The Extensible Firmware Interface System Partition (ESP) is a location that UEFI firmware uses to load operating system boot components. The contents of the ESP should not change often. Security operations personnel may want to ensure metadata of ESP contents such as names, paths, signatures, and hashes of binaries as well as names, paths, hashes, and file sizes of non-executable content (e.g. configuration files) are continuously collected to aid in identifying known vulnerable boot components and to enable the ability to perform anomaly detection.

Additional Guidance

Additional guidance related to hardware and firmware security is published at https://github.com/nsacyber/Hardware-and-Firmware-Security-Guidance on an ongoing basis.

Works Cited

[1] Eclypsium (2020). There's a hole in the boot. [Online] Available at: https://www.eclypsium.com/2020/07/29/theres-a-hole-in-the-boot/

[2] Microsoft (2020). ADV200011 | Microsoft Guidance for Addressing Security Feature Bypass in GRUB. [Online] Available at: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV200011

[3] UEFI Forum (2020). UEFI Revocation List File. [Online] Available at: https://www.uefi.org/revocationlistfile

[4] Microsoft (2020). Microsoft guidance for applying Secure Boot DBX update. [Online] Available at: https://support.microsoft.com/en-us/help/4575994/microsoft-guidance-for-applying-secure-boot-dbx-update

[5] ZDNet (2020). Microsoft pulls security update after reports of issues affecting some PCs. [Online] Available at: https://www.zdnet.com/article/microsoft-pulls-security-update-after-reports-of-issues-affecting-some-pcs/

[6] National Security Agency (2020). UEFI Secure Boot Customization. [Online] Available at: https://www.nsa.gov/What-We-Do/Cybersecurity/Advisories-Technical-Guidance/

Trademarks

Eclypsium is a registered trademark of Eclypsium, Inc in the United States and/or other countries.
Linux is the registered trademark of Linus Torvalds in the U.S. and other counties.
Windows and BitLocker are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Disclaimer of Endorsement

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

Purpose

This document was developed in furtherance of NSA's cybersecurity missions, including its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense information systems, and the Defense Industrial Base, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Contact

Client Requirements / General Cybersecurity Inquiries: Cybersecurity Requirements Center, 410-854-4200, Cybersecurity_Requests@nsa.gov
Media inquiries / Press Desk: Media Relations, 443-634-0721, MediaRelations@nsa.gov