USN-4751-1: Linux kernel vulnerabilities

It was discovered that the console keyboard driver in the Linux kernel contained a race condition. A local attacker could use this to expose sensitive information (kernel memory). (CVE-2020-25656)
Minh Yuan discovered that the tty driver in the Linux kernel contained race conditions when handling fonts. A local attacker could possibly use this to expose sensitive information (kernel memory). (CVE-2020-25668)

Bodong Zhao discovered a use-after-free in the Sun keyboard driver implementation in the Linux kernel. A local attacker could use this to cause a denial of service or possibly execute arbitrary code. (CVE-2020-25669)

Kiyin (□□) discovered that the perf subsystem in the Linux kernel did not properly deallocate memory in some situations. A privileged attacker could use this to cause a denial of service (kernel memory exhaustion). (CVE-2020-25704)

Julien Grall discovered that the Xen dom0 event handler in the Linux kernel did not properly limit the number of events queued. An attacker in a guest VM could use this to cause a denial of service in the host OS. (CVE-2020-27673)

Jinoh Kang discovered that the Xen event channel infrastructure in the Linux kernel contained a race condition. An attacker in guest could possibly use this to cause a denial of service (dom0 crash). (CVE-2020-27675)

Daniel Axtens discovered that PowerPC RTAS implementation in the Linux kernel did not properly restrict memory accesses in some situations. A privileged local attacker could use this to arbitrarily modify kernel memory, potentially bypassing kernel lockdown restrictions. (CVE-2020-27777)

It was discovered that the jfs file system implementation in the Linux kernel contained an out-of-bounds read vulnerability. A local attacker could use this to possibly cause a denial of service (system crash). (CVE-2020-27815)

Shisong Qin and Bodong Zhao discovered that Speakup screen reader driver in the Linux kernel did not correctly handle setting line discipline in some situations. A local attacker could use this to cause a denial of service (system crash). (CVE-2020-27830, CVE-2020-28941)

It was discovered that a use-after-free vulnerability existed in the infiniband hfi1 device driver in the Linux kernel. A local attacker could possibly use this to cause a denial of service (system crash). (CVE-2020-27835)

It was discovered that an information leak existed in the syscall implementation in the Linux kernel on 32 bit systems. A local attacker could use this to expose sensitive information (kernel memory). (CVE-2020-28588)

Minh Yuan discovered that the framebuffer console driver in the Linux kernel did not properly handle fonts in some conditions. A local attacker could use this to cause a denial of service (system crash) or possibly expose sensitive information (kernel memory). (CVE-2020-28974)

Michael Kurth and Pawel Wieczorkiewicz discovered that the Xen event processing backend in the Linux kernel did not properly limit the number of events queued. An attacker in a guest VM could use this to cause a denial of service in the host OS. (CVE-2020-29568)

Olivier Benjamin and Pawel Wieczorkiewicz discovered a race condition the Xen paravirt block backend in the Linux kernel, leading to a use-after-free vulnerability. An attacker in a guest VM could use this to cause a denial of service in the host OS. (CVE-2020-29569)

Jann Horn discovered that the tty subsystem of the Linux kernel did not use consistent locking in some situations, leading to a read-after-free vulnerability. A local attacker could use this to cause a denial of service

(system crash) or possibly expose sensitive information (kernel memory).
(CVE-2020-29660)

Jann Horn discovered a race condition in the tty subsystem of the Linux
kernel in the locking for the TIOCSPGRP ioctl(), leading to a use-after-
free vulnerability. A local attacker could use this to cause a denial of
service (system crash) or possibly execute arbitrary code.
(CVE-2020-29661)

It was discovered that a race condition existed that caused the Linux
kernel to not properly restrict exit signal delivery. A local attacker
could possibly use this to send signals to arbitrary processes.
(CVE-2020-35508)

USN-4747-2: GNU Screen vulnerability

USN-4747-1 fixed a vulnerability in screen. This update provides
the corresponding update for Ubuntu 14.04 ESM.
Original advisory details:

Felix Weinmann discovered that GNU Screen incorrectly handled certain
character sequences. A remote attacker could use this issue to cause GNU

Screen to crash, resulting in a denial of service, or possibly execute
arbitrary code.

Linux Foundation Announces DizmeID Foundation to Develop and Enable a Self-Sovereign Identity Credential Network

New DizmeID Foundation and technical project to advance the development of identity credentialing

SAN FRANCISCO, Calif., February 24, 2021 – The Linux Foundation, the nonprofit organization enabling mass innovation through open source, today announced the DizmeID Foundation and technical project with the intent to support digital identity credentialing. The effort will combine the benefits of self-sovereign identity with necessary compliance and regulation, with the aim to enable wallet holders with ownership and control over their digital identity and data access and distribution.

Founding Premier Members of the DizmeID Foundation include: Algorand, Fabrick and InfoCert.

A.P.S.P. is an Associate Member. Participation also includes a Start-up Supporter program for small organizations that want to advance the development of digital identity. Initial startups include eTuitus, Faberbee, Mopso/Amlet and Nym.

The DizmeID technical project leverages the Trust Over IP metamodel and builds upon three areas of existing infrastructure to focus its work on layer 4 that defines and implements the DizmeID features and business model.

“I’m proud to see our InfoCert research project becoming today the DizmeID Foundation cornerstone. We are ready to work with DizmeID Foundation members and all the community contributors in a joint effort to push the adoption of decentralized identity vision and bridge the gap between SSI and eIDAS,” said Daniele Citterio, Chief Technology Officer of InfoCert.

The DizmeID Foundation and technical project will define and allow for implementation of Dizme features on top of Sovrin public identity utility. The Dizme ecosystem is expected to include various technological components leveraging Hyperledger stack and adding a monetization layer based on Algorand blockchain protocol, which will enable the exchange of verifiable credentials and the development of new vertical applications. The identity credentials are managed with three levels of assurance: low, self-declared information; medium, automatic checks; and substantial, trusted identification. These levels of assurance would enable industry to have safer, innovative and cost-effective onboarding processes.

“We are thrilled that the DizmeID Foundation and Linux Foundation have chosen Algorand as the efficient transactional layer for their innovative self-sovereign identity solutions. With a shared vision of decentralized digital identity as a key primitive of the new way of exchanging value, we are honored that Algorand is a Founding Member of this important initiative,” said Pietro Grassano, Business Solutions Director Europe for Algorand.

“We at Fabrick are happy to be one of the Founding Member of DizmeID Foundation. We are pleased to share the vision of building an innovative open and decentralized identity framework with top-notch partners such as InfoCert and Algorand. We strongly believe Dizme ecosystem will sooner be one of the key innovation pillars enabling our Open Finance Ecosystem growth," said Paolo Zaccardi, CEO and cofounder of Fabrick.

"As part of the Linux Foundation, DizmeID Foundation will take advantage of existing innovations in open governance and blockchain technology communities," said Mike Dolan, senior vice president and general manager of Projects at the Linux Foundation. "DizmeID Foundation will take us one step closer to a self-sovereign identity future."

DizmeID Foundation is calling for members and contributors to help build the Dizme ecosystem. For more information and to contribute to this work, please visit: https://www.dizme.io/foundation

About the Linux Foundation

Founded in 2000, the Linux Foundation is supported by more than 1,000 members and is the world's leading home for collaboration on open source software, open standards, open data, and open hardware. Linux Foundation's projects are critical to the world's infrastructure including Linux, Kubernetes, Node.js, and more. The Linux Foundation's methodology focuses on leveraging best practices and addressing the needs of contributors, users and solution providers to create sustainable models for open collaboration. For more information, please visit us at linuxfoundation.org.

###

The Linux Foundation has registered trademarks and uses trademarks. For a list of trademarks of The Linux Foundation, please see its trademark usage page: *www.linuxfoundation.org/trademark-usage*. Linux is a registered trademark of Linus Torvalds.

Media Contact

pr@linuxfoundation.org

The post Linux Foundation Announces DizmeID Foundation to Develop and Enable a Self-Sovereign Identity Credential Network appeared first on Linux Foundation.

Google Funds Linux Kernel Developers to Focus Exclusively on Security

Long-time Linux kernel maintainers Gustavo Silva and Nathan Chancellor to dedicate their focus to maintaining and improving Linux security for the long-term

SAN FRANCISCO, February 24, 2021 – Today, Google and the Linux Foundation announced they are prioritizing funds to underwrite two full-time maintainers for Linux kernel security development, Gustavo Silva and Nathan Chancellor.

Silva and Chancellor’s exclusive focus is to maintain and improve kernel security and associated initiatives in order to ensure the world’s most pervasive open source software project is sustainable for decades to come.

The Linux Foundation’s Open Source Security Foundation (OpenSSF) and the Laboratory for Innovation Science at Harvard (LISH) recently published an open source contributor survey report that identified a need for additional work on security in open source software, which includes the massively pervasive Linux operating system. Linux is fueled by more than 20,000 contributors and as of August 2020, one million commits. While there are thousands of Linux kernel developers, all of whom take security into consideration as the due course of their work, this contribution from Google to underwrite two full-time Linux security maintainers signals the importance of security in the ongoing sustainability of open source software.

“At Google, security is always top of mind and we understand the critical role it plays to the sustainability of open source software,” said Dan Lorenc, Staff Software Engineer, Google. “We’re honored to support the efforts of both Gustavo Silva and Nathan Chancellor as they work to enhance the security of the Linux kernel.”

Chancellor’s work will be focused on triaging and fixing all bugs found with Clang/LLVM compilers while working on establishing continuous integration systems to support this work ongoing. Once those aims are well-established, he plans to begin adding features and polish to the kernel using these compiler technologies. Chancellor has been working on the Linux kernel for four and a half years. Two years ago, Chancellor started contributing to mainline Linux under the ClangBuiltLinux project, which is a collaborative effort to get the Linux kernel building with Clang and LLVM compiler tools.

“I hope that more and more people will start to use the LLVM compiler infrastructure project and contribute fixes to it and the kernel – it will go a long way towards improving Linux security for everyone,” said Chancellor, Linux maintainer.

Gustavo Silva’s full-time Linux security work is currently dedicated to eliminating several classes of buffer overflows by transforming all instances of zero-length and one-element arrays into flexible-array members, which is the preferred and least error-prone mechanism to declare such variable-length types. Additionally, he is actively focusing on fixing bugs before they hit the mainline, while also proactively developing defense mechanisms that cut off whole classes of vulnerabilities. Silva sent his first kernel patch in 2010 and today is an active member of the Kernel Self Protection Project (KSPP). He is consistently one of the top five most active kernel developers since 2017 with more than 2,000 commits in mainline. Silva’s work has impacted 27 different stable trees, going all the way down to Linux v3.16.

“We are working towards building a high-quality kernel that is reliable, robust and more resistant to attack every time,” said Silva, Linux maintainer. “Through these efforts, we hope people, maintainers in particular, will recognize the importance of adopting changes that will make their code less prone to common errors.”

“Ensuring the security of the Linux kernel is extremely important as it’s a critical part of modern computing and infrastructure. It requires us all to assist in any way we can to ensure that it is sustainably secure,” said David A. Wheeler, the Linux Foundation. “We extend a special thanks to Google for underwriting Gustavo and Nathan’s Linux kernel security development work along with a thank you to all the maintainers, developers and organizations who have made the Linux kernel a collaborative global success.”

Funding Linux kernel security and development is a collaborative effort, supported by the world’s largest companies that depend on the Linux operating system. To support work like this, discussions are taking place in the Securing Critical Projects Working Group inside the OpenSSF.

###

Media Contact

Jennifer Cloer
Story Changes Culture
503-867-2304
jennifer@storychangesculture.com

The post Google Funds Linux Kernel Developers to Focus Exclusively on Security appeared first on Linux Foundation.

USN-4747-1: GNU Screen vulnerability

Felix Weinmann discovered that GNU Screen incorrectly handled certain
character sequences. A remote attacker could use this issue to cause GNU
Screen to crash, resulting in a denial of service, or possibly execute
arbitrary code.