# USN-4910-1: Linux kernel vulnerabilities

Ryota Shiga discovered that the sockopt BPF hooks in the Linux kernel could
allow a user space program to probe for valid kernel addresses. A local
attacker could use this to ease exploitation of another kernel vulnerability. (CVE-2021-20239)
It was discovered that the BPF verifier in the Linux kernel did not
properly handle signed add32 and sub integer overflows. A local attacker
could use this to cause a denial of service (system crash) or possibly
execute arbitrary code. (CVE-2021-20268)

It was discovered that the priority inheritance futex implementation in the
Linux kernel contained a race condition, 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-2021-3347)

It was discovered that the network block device (nbd) driver in the Linux
kernel contained a use-after-free vulnerability during device setup. A
local attacker with access to the nbd device could use this to cause a
denial of service (system crash) or possibly execute arbitrary code.
(CVE-2021-3348)

□□ discovered that the NFS implementation in the Linux kernel

did not
properly prevent access outside of an NFS export that is a subdirectory of
a file system. An attacker could possibly use this to bypass NFS access
restrictions. (CVE-2021-3178)

---

# USN-4907-1: Linux kernel vulnerabilities

Wen Xu discovered that the xfs file system implementation in the Linux
kernel did not properly validate the number of extents in an inode. An
attacker could use this to construct a malicious xfs image that, when
mounted, could cause a denial of service (system crash). (CVE-2018-13095)
It was discovered that the priority inheritance futex implementation in the
Linux kernel contained a race condition, 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-2021-3347)

It was discovered that the network block device (nbd) driver in the Linux
kernel contained a use-after-free vulnerability during device setup. A
local attacker with access to the nbd device could use this to cause a

denial of service (system crash) or possibly execute arbitrary code.
(CVE-2021-3348)

---

# How To: Enable x11 Forwarding with SSH

In the last article, I explained how to enable SSH. In today’s article, we’re going to learn how to forward GUI application windows with SSH. x11 forwarding is easy and beneficial.

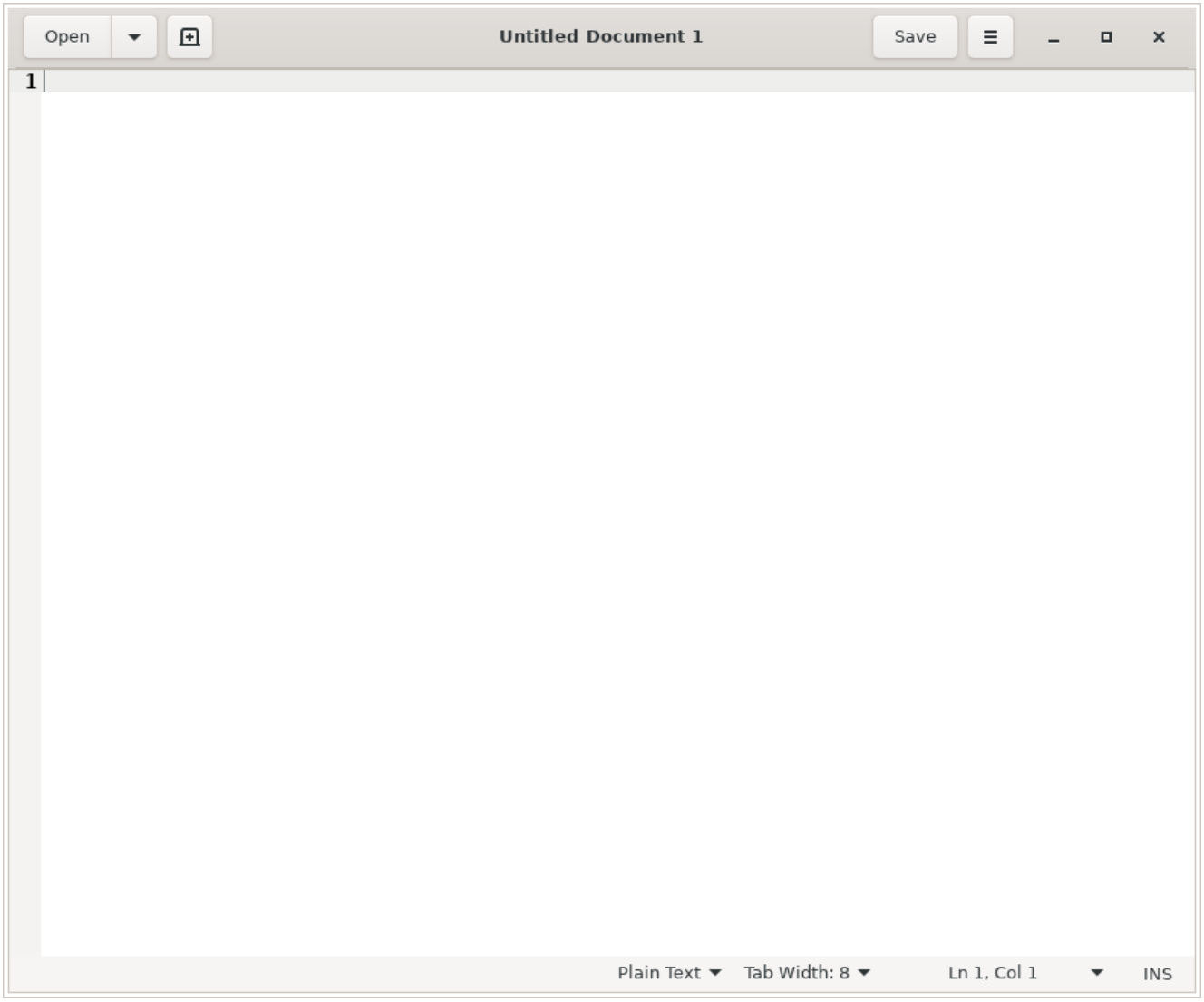
Just to quickly clear up a misconception, x11 forwarding works just fine with Wayland. Way back in the earliest days, it was agreed that it should retain backwards compatibility with x11 forwarding.

## Enable x11 Forwarding With SSH

What is this strange thing, this x11 forwarding?

Well, when you’re connected to another computer via SSH you can use the terminal to control the computer. That’s great, but what if you want to use a GUI application? Sure, you could set up some sort of remote desktop application, such as VNC. Or, alternatively, you can just forward graphic applications over SSH. It’s remarkably easy!

Perhaps a picture is in order. Check this:




This GEdit is actually running on my laptop, forwarded to this desktop.

That’s right. That’s running on my laptop. I’ve just forwarded the GUI application to this computer. If I write something and save it, it’d be saved on the computer that I’m connected to and not the computer that I’m using.

Amusingly, I used this just earlier today. I had to move a complex password to my laptop and I was being lazy. See? It comes in handier than you might think. Okay, I could have easily used nano, but I wanted to make sure that I’d configured x11 forwarding properly and get a screenshot.

So, how do you do it?

Well, first you need to crack open your terminal. To do that,

you just press CTRL + ALT + T on your keyboard and your default terminal emulator will open.

Now, in said terminal, I want you to run the following command:

[code]sudo nano /etc/ssh/sshd_config[/code]

Once you have that open, you just need to remove the appropriate asterisk (uncomment it out) for the right line. Look for the line that says:

[code]#X11Forwarding yes[/code]

And change it to:

[code]X11Forwarding yes[/code]

Then save the file by pressing CTRL + X, then Y, and then ENTER.

Next up, you'll need to restart the SSH service and that's done with:

[code]systemctl restart sshd[/code]

And that's it. You can now use x11 forwarding over SSH. To do so, you just need to add the -X switch.

[code]ssh user@host_name.local -X[/code]

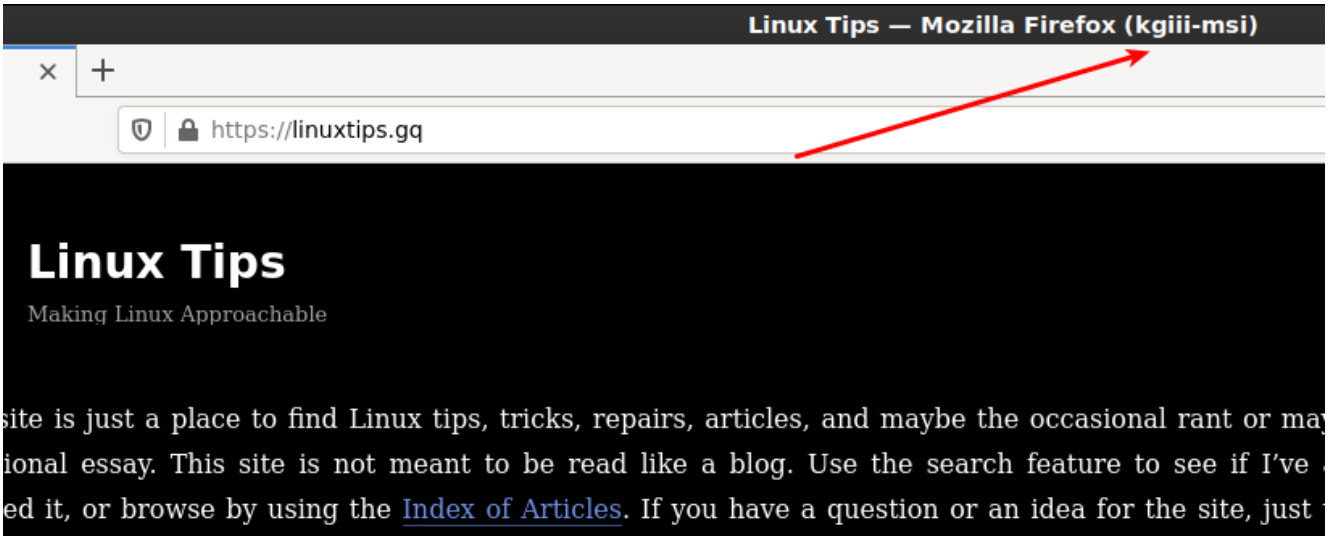
To try to make sense of that, if I were to do this connecting to the new MSI laptop, then my command would look just like:

[code]ssh kgiii@kgiii-msi.local -X[/code]

You can also use the IP address, instead of the hostname, just like we discussed in the previous article about SSH. To do that, it looks like this:

[code]ssh user@ip_address -X[/code]

Once you're there, just go ahead and start an application. For example, open gedit by typing just that. You may find some applications won't work, often due to ownership and permissions, but you'll find many that do. If you find one that doesn't work, you can always check any errors thrown and work to resolve the issue.




See? Note the carefully drawn arrow that shows where it was forwarded from. Tada!

That's an example of Firefox forwarded over SSH using x11 forwarding and you may notice the washed out look. I haven't really dug into it, but I am reasonably confident that it's because of compression. I've never needed to dig into that and, amazingly enough, I don't know everything and don't see any reason to invest time learning about it any further. You get what I know, not what I am able to learn! (You're welcome!)

Anyhow, there you have it. One more article in the books and one more bit of knowledge plastered across the internet. If you found the article useful, you could use that rating button. I kinda use those ratings to decide what to write. You can also sign up for the newsletter. I had to remove some @gmx.com email address because they simply don't let my emails through. (I've never sent a single unsolicited message, it's just a horrible ccTLD and it gets filtered often.) Sign up

again with a different email address. Thanks for reading!

---

# Linux Foundation Hosts Collaboration Among World's Largest Insurance Companies

*openIDL platform provides a standardized data repository streamlining regulatory reporting and enabling the delivery of next-gen risk and insurance applications*

San Francisco, Calif., April 12, 2021 – The Linux Foundation, the nonprofit organization enabling mass innovation through open source, and the American Association of Insurance Services (AAIS), today are announcing the launch of OpenIDL, the Open Insurance Data Link platform and project. The platform will reduce the cost of regulatory reporting for insurance carriers, provide a standardized data repository for analytics and a connection point for third parties to deliver new applications to members.

openIDL brings together some of the world's largest insurance companies, including The Hanover and Selective Insurance Group, along with technology and service providers Chainyard, KatRisk and MOBI to advance a common distributed ledger platform for sharing information and business processes across the insurance ecosystem.

The first use case for the openIDL network is regulatory reporting in the Property and Casualty (P&C) insurance industry. Initially built with guidance from AAIS, a leading insurance advisory organization and statistical reporting agent, openIDL leverages the trust and integrity inherent in

distributed ledger networks. The secure platform guarantees to regulators and other insurance industry participants that data is accurate and complete, implemented by a "P&C Reporting Working Group" within the openIDL network.

"From the very beginning, we recognized the enormous transformative potential for openIDL and distributed ledger technology," said AAIS CEO Ed Kelly. "We are happy to work with the Linux Foundation to help affect meaningful, positive change for the insurance ecosystem."

Insurance sectors beyond P&C are expected to be supported by openIDL in the coming months, and use cases will expand beyond regulatory. A "Flood Working Group" has already been assembled to develop use case catastrophe modeling in support of insurers and regulators. openIDL is also collaborating on joint software development activities, building upon Hyperledger Fabric, Hadoop, Node.js, MongoDB and other open technologies to implement a "harmonized data store," enabling data privacy and accountable operations.

The combined packaging of this software is called an "openIDL Node," approved and certified by developers working on this project, and every member of the network will be running that software in order to participate in the openIDL network. Additional joint software development for analytics and reporting are also included in the openIDL Linux Foundation network.

"We're delighted to join openIDL with AAIS and the Linux Foundation. It is strategically important for Selective to be part of industry efforts to innovate our regulatory reporting and use distributed ledgers," said Michael H. Lanza, executive vice president, general counsel & chief compliance officer of Selective Insurance Group, Inc.

openIDL is a Linux Foundation "Open Governance Network." These networks comprise nodes run by many different organizations,

bound by a shared distributed ledger that provides an industry utility platform for recording transactions and automating business processes. It leverages open source code and community governance for objective transparency and accountability among participants. The network and the node software are built using open source development practices and principles managed by the Linux Foundation in a manner that enterprises can trust.

“AAIS, and the insurance industry in general, are trailblazers in their contribution and collaboration to these technologies,” said Mike Dolan, senior vice president and general manager of Projects at the Linux Foundation. “Open governance networks like openIDL can now accelerate innovation and development of new product and service offerings for insurance providers and their customers. We’re excited to host this work.”

As an open source project, all software source code developed will be licensed under an OSI-approved open source license, and all interface specifications developed will be published under an open specification license. And all technical discussions between participants will take place publicly, further enhancing the ability to expand the network to include other participants. As with an openly accessible network, organizations can develop their own proprietary applications and infrastructure integrations.

**Additional Members & Partner Statements**

**Chainyard**

“Chainyard is pleased to join the OpenIDL initiative as an infrastructure member,” said Isaac Kunkel, Chainyard SVP Consulting Services. “Blockchain is a team sport and with the openIDL platform, companies, regulators and vendors are forming an ecosystem to collaborate on common issues for the betterment of the insurance industry. The entire industry will

benefit through more accurate data and better decision making.”

**KatRisk**

“The openIDL platform will serve to increase access to state of the art catastrophe modelling data from KatRisk and others, serving to reduce the friction required to house and run said models. KatRisk expects all parties, from direct insurance entities to regulators, to see an increase in data quality, reliability and ease of access as catastrophe modelling output is effectively streamed across OpenIDL nodes to generate automated reports and add to or create internal business intelligence databases. If catastrophe models are about owning your own risk, then the OpenIDL platform is an effective tool to better understand and manage that risk,” said Brandon Katz, executive vice president, member, KatRisk.

**MOBI**

“The Mobility Open Blockchain Initiative (MOBI) is delighted to join with the Linux Foundation, AAIS, and insurance industry leaders in founding OpenIDL.  Data sharing and digital collaboration in business ecosystems via industry consortium ledgers like OpenIDL will drive competitive advantage for many years to come,” said Chris Ballinger, founder and CEO, MOBI.

For more information, please visit www.openidl.org

**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.

**ABOUT AAIS**

Established in 1936, AAIS serves the property casualty insurance industry as the modern, Member-based advisory organization. AAIS delivers custom advisory solutions, including best-in-class forms, rating information and data management capabilities for commercial lines, inland marine, farm & agriculture, commercial auto, personal auto, and homeowners insurers. Its consultative approach, unrivaled customer service and modern technical capabilities underscore a focused commitment to the success of its Members. For more information about AAIS, please visit www.AAISonline.com.

###



Media Contact
Jennifer Cloer for Linux Foundation
503-867-2304
jennifer@storychangesculture.com

The post Linux Foundation Hosts Collaboration Among World’s Largest Insurance Companies appeared first on Linux Foundation.

# USN-4899-2: SpamAssassin vulnerability

USN-4899-1 fixed a vulnerability in SpamAssassin. This update provides
the corresponding update for Ubuntu 14.04 ESM.
Original advisory details:

Damian Lukowski discovered that SpamAssassin incorrectly handled certain CF
files. If a user or automated system were tricked into using a specially-
crafted CF file, a remote attacker could possibly run arbitrary code.