USN-4734-1: wpa_supplicant and hostapd vulnerabilities

It was discovered that wpa_supplicant did not properly handle P2P
(Wi-Fi Direct) group information in some situations, leading to a
heap overflow. A physically proximate attacker could use this to cause a
denial of service or possibly execute arbitrary code. (CVE-2021-0326)
It was discovered that hostapd did not properly handle UPnP subscribe
messages in some circumstances. An attacker could use this to cause a
denial of service. (CVE-2020-12695)

How To: Check Your Logs For Errors

When you notice a problem with your Linux box, you're possibly only noticing the symptoms and not actually seeing the reason. The reasons are often made clear in your error logs. This is how to read them.

Back in the early 1900s (when horses, trains, and your feet were the primary mode of travel), it required the terminal to really deal with your error logs. You'd ask for help on a forum and they'd want you to run something like:

[code]grep -i "Feb 10" /var/log/*.log | sort | uniq -c | sort -n[/code]

It worked and it worked well, especially if you knew what you were looking for. If you weren't, then you'd be scrolling through tons of text in the terminal with few visual cues or no easy way to sort through the data.

Well, enter KSystemLog. KSystemLog's man page describes it as thus:

Description: system log viewer
KSystemLog show all logs of your system, grouped by General (Default system
log, Authentication, Kernel, X.org…), and optional Services (Apache, Cups,
etc, …). It includes many features to read nicely your log files:
* Colorize log lines depending on their severities
* Tabbed view to allow displaying several logs at the same time
* Auto display new lines logged
* Detailed information for each log lines
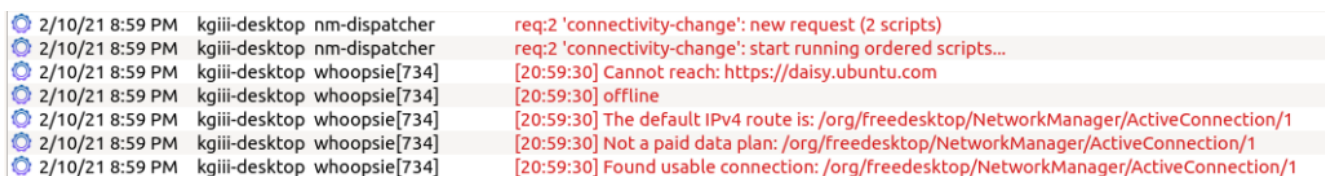
And it looks like this:
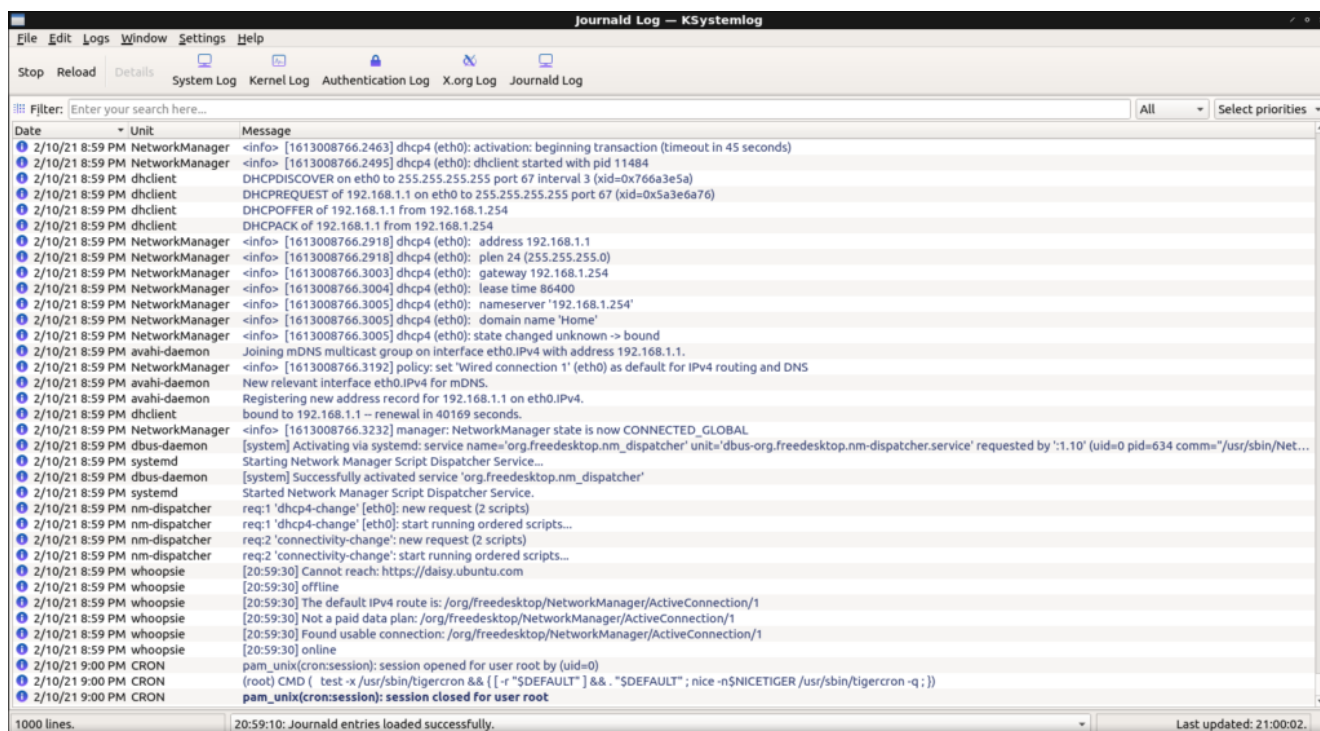
KSystemlog UI — not pictured, fancy colors

As you can see from the image, this is a handy GUI method to view your various system logs. If you look at the image above, you'll see that one of the lines is a darker blue than the previous lines. This is because it was a new log entry, an event that happened after I'd opened the application.

So, in other words, you can open KSystemLog and then open an application that's giving you trouble and see what errors are thrown. You can also examine the entire list of errors and events.

Not only that, but the events in the logs are color-coded. This provides easy visual cues as the events range in color from a light gray (normal) to a bright red for emergency-level events. You can also apply filters and search for specific events.

Note the color change for errors.

KSystemLog is a KDE application but pulls in very few dependencies. It loads rapidly and will pick up all installed log files by default. In my case, it finds these:
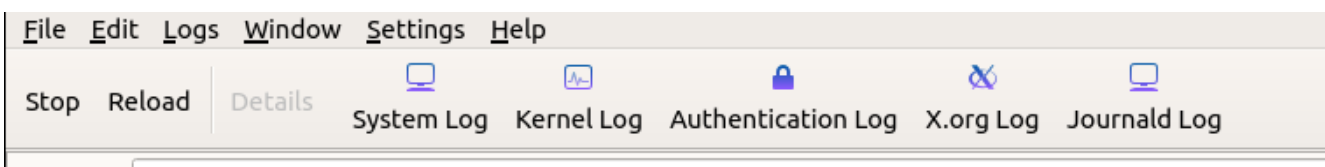
Currently available logs on this particular system.

Chances are pretty good that it is in your default repositories. If you're using a Debian derivative then you can install it by opening the terminal and entering:

[code]sudo apt install ksystemlog -y[/code]

Anyhow, there you have it. Adjust the above command for your distro's package manager (or go dig it out if you're manually installing packages). You can take the information from the logged errors and use them as search terms to help.

It's absolutely amazing how many errors can be resolved by just taking the exact error message and tossing it verbatim into a search engine of your choice. I've used this method to troubleshoot so many times that I think it's an asset we should all have in our toolbox.

As always, go ahead and subscribe to get notifications when new articles are published. If you want, you can go ahead and register and start helping. If you don't want to maintain your own site and want to write articles, that can be arranged. Thanks for reading!

Understanding Open Governance Networks

Throughout the modern business era, industries and commercial operations have shifted substantially to digital processes. Whether you look at EDI as a means to exchange invoices or cloud-based billing and payment solutions today, businesses have steadily been moving towards increasing digital operations. In the last few years, we've seen the promises of digital transformation come alive, particularly in industries that have shifted to software-defined models. The next step of this journey will involve enabling digital transactions through decentralized networks.

A fundamental adoption issue will be figuring out who controls and decides how a decentralized network is governed. It may

seem oxymoronic at first, but decentralized networks still need governance. A future may hold autonomously self-governing decentralized networks, but this model is not accepted in industries today. The governance challenge with a decentralized network technology lies in who and how participants in a network will establish and maintain policies, network operations, on/offboarding of participants, setting fees, configurations, and software changes and are among the issues that will have to be decided to achieve a successful network. No company wants to participate or take a dependency on a network that is controlled or run by a competitor, potential competitor, or any single stakeholder at all for that matter.

Earlier this year, we presented a solution for Open Governance Networks that enable an industry or ecosystem to govern itself in an open, inclusive, neutral, and participatory model. You may be surprised to learn that it's based on best practices in open governance we've developed over decades of facilitating the world's most successful and competitive open source projects.

The Challenge

For the last few years, a running technology joke has been "describe your problem, and someone will tell you blockchain is the solution." There have been many other concerns raised and confusion created, as overnight headlines hyped cryptocurrency schemes. Despite all this, behind the scenes, and all along, sophisticated companies understood a distributed ledger technology would be a powerful enabler for tackling complex challenges in an industry, or even a section of an industry.

At the Linux Foundation, we focused on enabling those organizations to collaborate on open source enterprise blockchain technologies within our Hyperledger community. That

community has driven collaboration on every aspect of enterprise blockchain technology, including identity, security, and transparency. Like other Linux Foundation projects, these enterprise blockchain communities are open, collaborative efforts. We have had many vertical industry participants engage, from retail, automotive, aerospace, banking, and others participate with real industry challenges they needed to solve. And in this subset of cases, enterprise blockchain is the answer.

The technology is ready. Enterprise blockchain has been through many proof-of-concept implementations, and we've already seen that many organizations have shifted to production deployments. A few notable examples are:

- Trust Your Supplier Network 25 major corporate members from Anheuser-Busch InBev to UPS In production since September 2019.
- Foodtrust Launched Aug 2017 with ten members, now being used by all major retailers.
- Honeywell 50 vendors with storefronts in the new marketplace. In its first year, GoDirect Trade processed more than $5 million in online transactions.

However, just because we have the technology doesn't mean we have the appropriate conditions to solve adoption challenges. A certain set of challenges about networks' governance have become a "last mile" problem for industry adoption. While there are many examples of successful production deployments and multi-stakeholder engagements for commercial enterprise blockchains already, specific adoption scenarios have been halted over uncertainty, or mistrust, over who and how a blockchain network will be governed.

To precisely state the issue, in many situations, company A does not want to be dependent on, or trust, company B to control a network. For specific solutions that require broad industry participation to succeed, you can name any industry,

and there will be company A and company B.

We think the solution to this challenge will be Open Governance Networks.

The Linux Foundation vision of the Open Governance Network

An Open Governance Network is a distributed ledger service, composed of nodes, operated under the policies and directions of an inclusive set of industry stakeholders.

Open Governance Networks will set the policies and rules for participation in a decentralized ledger network that acts as an industry utility for transactions and data sharing among participants that have permissions on the network. The Open Governance Network model allows any organization to participate. Those organizations that want to be active in sharing the operational costs will benefit from having a representative say in the policies and rules for the network itself. The software underlying the Open Governance Network will be open source software, including the configurations and build tools so that anyone can validate whether a network node complies with the appropriate policies.

Many who have worked with the Linux Foundation will realize an open, neutral, and participatory governance model under a nonprofit structure that has already been thriving for decades in successful open source software communities. All we're doing here is taking the same core principles of what makes open governance work for open source software, open standards, and open collaboration and applying those principles to managing a distributed ledger. This is a model that the Linux Foundation has used successfully in other communities, such as the Let's Encrypt certificate authority.

Our ecosystem members trust the Linux Foundation to help solve

this last mile problem using open governance under a neutral nonprofit entity. This is one solution to the concerns about neutrality and distributed control. In pan-industry use cases, it is generally not acceptable for one participant in the network to have power in any way that could be used as an advantage over someone else in the industry. The control of a ledger is a valuable asset, and competitive organizations generally have concerns in allowing one entity to control this asset. If not hosted in a neutral environment for the community's benefit, network control can become a leverage point over network users.

We see this neutrality of control challenge as the primary reason why some privately held networks have struggled to gain widespread adoption. In order to encourage participation, industry leaders are looking for a neutral governance structure, and the Linux Foundation has proven the open governance models accomplish that exceptionally well.

This neutrality of control issue is very similar to the rationale for public utilities. Because the economic model mirrors a public utility, we debated calling these "industry utility networks." In our conversations, we have learned industry participants are open to sharing the cost burden to stand up and maintain a utility. Still, they want a low-cost, not profit-maximizing model. That is why our nonprofit model makes the most sense.

It's also not a public utility in that each network we foresee today would be restricted in participation to those who have a stake in the network, not any random person in the world. There's a layer of human trust that our communities have been enabling on top of distributed networks, which started with the Trust over IP Foundation.

Unlike public cryptocurrency networks where anyone can view the ledger or submit proposed transactions, industries have a natural need to limit access to legitimate parties in their

industry. With minor adjustments to address the need for policies for transactions on the network, we believe a similar governance model applied to distributed ledger ecosystems can resolve concerns about the neutrality of control.

Understanding LF Open Governance Networks

Open Governance Networks can be reduced to the following building block components:

- Business Governance: Networks need a decision-making body to establish core policies (e.g., network policies), make funding and budget decisions, contracting with a network manager, and other business matters necessary for the network's success. The Linux Foundation establishes a governing board to manage the business governance.
- Technical Governance: Networks will require software. A technical open source community will openly maintain the software, specifications, or configuration decisions implemented by the network nodes. The Linux Foundation establishes a technical steering committee to oversee technical projects, configurations, working groups, etc.
- Transaction Entity: Networks will require a transaction entity that will a) act as counterparty to agreements with parties transacting on the network, b) collect fees from participants, and c) execute contracts for operational support (e.g., hiring a network manager).

Of these building blocks, the Linux Foundation already offers its communities the Business and Technical Governance needed for Open Governance Networks. The final component is the new, LF Open Governance Networks.

LF Open Governance Networks will enable our communities to establish their own Open Governance Network and have an entity

to process agreements and collect transaction fees. This new entity is a Delaware nonprofit, a nonstock corporation that will maximize utility and not profit. Through agreements with the Linux Foundation, LF Governance Networks will be available to Open Governance Networks hosted at the Linux Foundation.

If you're interested in learning more about hosting an Open Governance Network at the Linux Foundation, please contact us at **governancenetworks@linuxfoundation.org**

The post Understanding Open Governance Networks appeared first on Linux Foundation.

USN-4733-1: GNOME Autoar vulnerability

Yiğit Can Yılmaz discovered that GNOME Autoar could extract files outside
of the intended directory. If a user were tricked into extracting a
specially crafted archive, a remote attacker could create files in
arbitrary locations, possibly leading to code execution.

USN-4732-1: SQLite vulnerability

It was discovered that SQLite incorrectly handled certain sub-queries. An

attacker could use this issue to cause SQLite to crash,
resulting in a
denial of service, or possibly execute arbitrary code.