Securing Linux Systems in the Internet of Things

Four Essential Steps for Ongoing Threat Mitigation
EXECUTIVE SUMMARY

Open source Linux is a popular operating system choice for developers of embedded systems and devices in the Internet of Things (IoT). But with ever-increasing numbers of interconnected IoT devices being deployed, the threat of vulnerabilities has become more widespread than ever. Taking responsibility for identifying vulnerabilities and making the necessary updates to mitigate threats is often beyond the capacity of equipment builders and manufacturers. This paper outlines a proven four-step process for resolving Linux vulnerabilities: monitoring, assessment, notification, and remediation. It also explains the cost a company might incur in monitoring and fixing vulnerabilities in-house, and why it might make more sense to partner with an experienced Linux security team to ensure that deployed devices and systems continue to be protected.

TABLE OF CONTENTS

Executive Summary .................................................. 2
Our Vulnerable World .................................................. 3
Mind the Gaps .......................................................... 3
The Advantage of Open Source ...................................... 4
The Four Essential Steps .............................................. 4
  Monitoring .......................................................... 4
  Assessment .......................................................... 5
  Notification .......................................................... 5
  Remediation .......................................................... 5
The Price of Protection ................................................. 5
The Wind River Linux Security Response Process .................. 5
Conclusion ............................................................ 7
OUR VULNERABLE WORLD

Open source Linux software has gained favor among IoT system developers for a variety of reasons. Mostly, it is readily available through binary versions prepackaged with embedded hardware and an ever-growing set of community-driven frameworks that are built for Linux. It also offers some practical benefits for IoT applications, notably support for the interoperability that IoT devices often require. Moreover, the cloud systems that run IoT solutions are increasingly built on open source, Linux-based operating systems.

In today’s interconnected world, securing Linux-based systems and devices has become one of the most pressing challenges facing developers and device manufacturers. Gone are the days of “fire and forget” device deployment. Virtually every device made these days is designed for interconnectivity with something, which makes it highly susceptible to security vulnerabilities. Connected devices are targets and are becoming more vulnerable with every reported exploit.

With the rapid growth of IoT, interconnected devices are proliferating at exponential rates. This massive increase in devices, connections, data volume, network traffic, and users has brought a corresponding increase in cyberthreats across a wider attack surface. In response, device manufacturers and developers of IoT applications are employing sophisticated methods to build in powerful security functionality at the earliest stages of design. And that’s a good thing. In fact, it’s essential. But it’s not enough. Threats are constantly evolving. Operators of IoT systems need to have both a strategy and a mechanism to maintain robust security over the entire useful life of deployed devices.

Today, manufacturers need to expand their security strategies beyond system-level reinforcement to include the agile integration of new vulnerability patches. Unless systems are constantly updated, they are vulnerable to newly emerging threats, no matter how strong built-in security is when they deploy.

Consider the example of your own laptop computer. There was a time when all you needed to secure it was a password, and the biggest external threat was an infected floppy disk. Once you connect it to the Internet, however, it becomes a target for attackers, typically via the applications that reside on it. Chances are you receive weekly or monthly update advisories or auto-updates from app providers intended to protect your computer from the latest software vulnerabilities.

Every IoT device running Linux needs that same level of ongoing protection. The question is how to accomplish it in a systematic, scalable, and cost-effective manner.

MIND THE GAPS

Before you can fix vulnerabilities in a system, you have to know what and where they are. That’s becoming increasingly challenging as security vulnerabilities multiply in parallel to the expansion of the IoT.

Common Vulnerabilities and Exposures (CVE) is the government-funded, widely accepted industry standard for identifying, reporting, and repairing vulnerabilities in publicly released software. Using CVE identifiers, information about a vulnerability can be correlated to appropriate security patches or protection technologies, which is especially vital in the open source software world. Disclosure of vulnerabilities can come from a variety of sources, including the software vendor, security vendors, independent researchers, community mailing lists, and government agencies such as the U.S. Computer Emergency Readiness Team (US-CERT). Today, the CVE database is challenged by the growing volume and scale of vulnerabilities resulting from IoT—in 2018, 16,555 new vulnerabilities were reported, ranging in severity from 1 to 10.
Not only has the number of CVEs increased radically over time but the reported vulnerabilities have consistently increased in severity, due in part to the growing sophistication of cybercriminals, and in part to the multiple avenues of attack presented by an increasingly interconnected and AI-enabled world. McAfee Labs’ 2019 security predictions (www.mcafee.com/enterprise/en-us/threat-center/mcafee-labs/threat-predictions.html) listed these top areas of vulnerability:

|THE ADVANTAGE OF OPEN SOURCE|

Using open source software actually presents significant advantages from a security perspective. Ongoing threat mitigation requires the ability to update the software on a device as soon as a vulnerability is identified. Because of the large open source community, information about vulnerabilities surfaces quickly through legions of researchers, government agencies such as US-CERT, and dedicated mailing lists. As a result, users of open source in deployed devices can take fast action to lower a potential risk.

Still, the community will always address the fix in the latest iteration of the affected components and move on, focusing on innovation, while old versions of those components (e.g., kernel, libraries, etc.) will remain exposed.

It is impractical to think that any system can be rendered 100% impervious to outside threats from persistent attackers given sufficient time and resources. But specific measures can be taken to make things extremely difficult for hackers and considerably reduce the odds of breaches.

|THE FOUR ESSENTIAL STEPS|

Ongoing threat mitigation in deployed systems requires a four-step approach: monitoring, assessment, notification, and remediation.

1. Monitoring

Think of monitoring as the “surveillance camera” in your security strategy. Assuming two houses have strong locks, the one that also has the surveillance camera is going to be better prepared against an intrusion. In this case, the cameras are operated by organizations that issue vulnerability alerts and advisories, such as US-CERT, the National Institute of Standards and Technology (NIST), the CVE database, various security vendors, private mailing lists, and communities focused on finding Linux vulnerabilities. The challenge here is that, with dozens of organizations issuing advisories, there is bound to be a certain level of confusion about what vulnerabilities may affect which kinds of deployments. This makes it critical to know which organizations can be relied upon for relevant, accurate, and actionable information. The Wind River® Linux Security site (www.windriver.com/products/linux/security) offers constantly updated alerts about vulnerabilities specific to devices powered by our embedded Linux OS.
2. Assessment

Once an advisory or security report is issued, the system operator or its software partners need to make a determination as to whether its devices are vulnerable, and to what extent. A threat is typically ranked as high, medium, low, or not present, and it is prioritized based on likely severity, difficulty of attack, and likelihood of avoidance.

Assessment requires knowing exactly which packages and which versions are vulnerable, and also the exact configurations of your systems. Knowing this, of course, depends on how well your development team has documented its work. For vulnerable products, the clock for finding a patch starts the minute the vulnerability is exposed.

3. Notification

Once a vulnerability has been assessed, affected users must be notified promptly of the issue, the relative probability of vulnerability based on the nature of the device, and the action plan for remediation. This step requires the right tools and methodologies, so that notifications are sent to all affected parties in a timely and efficient manner. Timely notification depends on and reinforces a trusted relationship between software vendor and device deployer.

4. Remediation

The timing and method of remediation is usually based on priority. Vulnerabilities deemed to be of high severity may require an immediate “hot fix,” while others of lower severity may be covered in periodic software updates. The challenge is having the ability to quickly deliver effective patches and distribute them to end users via a secure channel. No matter what the severity of the vulnerability, addressing it in a timely manner will likely impact the focus of your business.

What would it cost to assemble a dedicated in-house security team? Based on 8,000 to 10,000 CVEs uncovered each year, an organization would require a team of four or five highly skilled engineers to investigate and address each one. At an average annual salary of $120,000 for the appropriate experience and skill set, the organization would need to budget as much as $500,000 a year just for staff.

Most device manufacturers and operators of IoT systems consider such specialized expertise outside of their core competency and beyond their budget. The more cost-effective alternative is to assign this responsibility to an experienced commercial Linux vendor with a dedicated security response team—a proven strategy for providing timely protection within hours of vulnerability publication, often weeks or months ahead of the upstream patching.

The right software partner would have the necessary connections within the Linux community and among advisory organizations—combined with its own monitoring and investigative capabilities—to stay on top of vulnerabilities as they are discovered. And because this provider would be able to scale its security response services across multiple customers, outsourcing this critical responsibility costs far less than trying to manage it in-house.

THE WIND RIVER LINUX SECURITY RESPONSE PROCESS

As a leading provider of commercial-grade Linux software for embedded applications, Wind River has designed in the resources necessary to help device manufacturers and their customers maintain ongoing threat mitigation over the life of their systems. The Wind River Linux Product Security Incident Response Team (PSIRT) identifies, monitors, resolves, and responds to Wind River Linux security vulnerabilities. The team follows the four-step process prescribed earlier and ensures adherence to the Wind River Security Response Policy, which establishes target response times based on the priority of the vulnerability.

The Wind River PSIRT is constantly monitoring the CVE database at cve.mitre.org for potential issues affecting Wind River Linux. This includes specific security notifications from U.S. government agencies and organizations such as NIST, US-CERT, and public and private security mailing lists. Wind River receives email alerts from each of these organizations whenever a new security threat arises. Alerts include both community-confirmed and potential vulnerabilities—the team looks into all of them.

THE PRICE OF PROTECTION

If this four-step process sounds like a lot of work, it is. There’s no denying that it requires a substantial commitment of people, time, and effort. There are no shortcuts. Speed of response is of the essence. The ideal solution is a dedicated security response team to address every potential vulnerability. If you try to do it all in-house, your team will need to change focus from development to remediation every time a vulnerability threatens.
Through its membership and participation in the appropriate forums, the security team is often privy to Linux vulnerabilities that have not yet been made public, allowing Wind River and the community to collectively close vulnerabilities and issue patches at a mutually agreed time that coincides with public announcement of the vulnerability. This results in a steady stream of security updates as well as same-day closure of some of the most severe vulnerabilities.

Our Wind River Linux Security website gives users an up-to-the-minute report on known vulnerabilities as well as tools that enable them to assess the threat to their own Linux deployments. Not only does the PSIRT roll all patches into future service packs and major releases of Wind River Linux but it ensures that all releases contain no known security vulnerabilities. If you build your devices using Wind River Linux, you have a dedicated security team at your disposal throughout each device’s deployed lifecycle.

In addition, all Wind River Linux development is contributed upstream into the Yocto Project. Your most important vulnerabilities are addressed in a timely manner while, simultaneously, the patches also become part of the mainstream Yocto Project and get the full weight of the community behind them.

**CONCLUSION**

Security vulnerabilities are a fact of life in today’s interconnected world, and they are multiplying with the proliferation of embedded IoT applications. Managing them and mitigating related threats are essential steps for the protection of end users, but such action requires a level of engagement that is beyond the scope of most IoT solution developers, device manufacturers, and system operators. Fortunately, the open source community is extremely vigilant in finding vulnerabilities that affect Linux software. By working with a software partner that is a leading contributor in key communities, with a proven process for monitoring and assessing threats, notifying customers, and fixing vulnerabilities, manufacturers and developers can help protect their customers against cyberthreats over the life of deployed IoT systems.