How Supply Chain Attacks Work — and How to Stop Them

Security leaders are scrambling to reexamine the security of their supply chains. This report details how to protect software components and applications from attack.

Brought to you by informa tech

Sponsored by

mend.io

INSIDE:

Next

FEATURE

How Supply Chain Attacks Work — and How to Stop Them

Software bills of materials are in the spotlight, but SBOMs are merely the first step in achieving software supply chain security.

By Jeffrey Schwartz, Contributing Writer, Dark Reading

With attackers increasingly targeting different parts of the supply chain, securing third-party components, partners, and software is becoming an important part of enterprise defense. Attacks targeting vulnerable software components such as the Apache Log4j library, vulnerabilities in widely used software applications such as Microsoft Exchange, or service providers such as CircleCI are on the rise, prompting organizations to scrutinize their software supply chain.

Software supply chain security has only elevated into a widespread critical issue during the past two years, and many security professionals are still trying to get their heads around it. “Everybody’s concerned about it, but they just don’t know how to approach it,” says ReversingLabs field chief information security officer (CISO) Matt Rose. “It’s still so nebulous in terms of its definition. I could ask 15 people what software supply chain security is, and I would probably get 15 different answers. The industry needs to determine exactly what it is.”

Rose’s definition encompasses the entire software development life cycle. “Software supply chain security is looking at all aspects of building software,” he says. “But truly, it’s that final examination of the deployed artifacts. All the work you’re doing to build the

software needs to be looked at before it's deployed. It's a bigger story than just one piece of technology."

Software supply chain incidents include vulnerabilities in dependencies, which encompass libraries, artifacts, and software frameworks, as well as the use of malicious packages in code repositories. Threat actors create use tactics such as typosquatting and dependency confusion to create these malicious packages.

During the year's first two months, research firm Comparitech logged 17,500 individual malicious packages, with more than 15,000 of them the result of a mass spam upload to the npm repository.

For now, much of the attention is on software bills of materials (SBOMs), because they present a critical first step in preventing software compromises by ensuring everyone knows what's in the applications. Since modern application development relies heavily on open source dependencies and libraries, many developers wind up pulling unvetted dependencies and libraries into their code. When that happens, the developer puts the organization and everyone else who later pulls the code at risk.

Starting With SBOMs

There is a growing movement to have developers generate SBOMs for their applications — to the point that SBOMs appear as a major provision in the Biden administration's Executive Order 14028, Improving the Nation's Cybersecurity. Suppliers who provide "critical systems" to federal agencies face a June 11 deadline to submit software bills of materials, marking a key milestone in the US government's promise to address the steep rise in software supply chain security incidents. The government's requirement applies to hardware and software. The provisions of the 2021 executive order also underscore the need for SBOMs to integrate with vulnerability exploitation exchanges (VEXs).

The White House issued the executive order following the attack on the widely deployed SolarWinds Orion network management system and the Apache Foundation's revelation of the Log4Shell vulnerability in the Log4j library. Both have famously wreaked havoc on DevSecOps organizations worldwide and are considered among the most extensive software supply chain incidents to date. While they weren't the first, SolarWinds and Log4j set off a cascade of software supply chain security alerts and have brought the risks to the forefront.

As the deadline for submitting the first SBOMs approaches, security teams face challenges in thwarting supply chain attacks by securing dependencies and protecting source-code repositories from malicious packages.

Limitations of SBOMs and VEXs

Kayla Underkoffler, lead security technologist at HackerOne, says assembling an SBOM is more complex than scanning and compiling software across an organization's environment. "We all maintain very complex environments, so whether you're producing a product or you're a vendor or a consumer in this space, you're purchasing other people's products and putting them into your environment," Underkoffler says.

> "Assembling an SBOM is more complex than scanning and compiling software across an organization's environment," says HackerOne's Kayla Underkoffler.

Increasingly, software composition and analysis (SCA) tools are gaining the ability to generate and export SBOMs. However, those that don't have that capability need to add it, Forrester senior analyst Janet Worthington noted in the researcher's Q1 2023 "Software Composition Analysis Landscape" report. "As the government and

Previous Next

software and security communities refine requirements and best practices for securing the software supply chain and SBOMs, SCA vendors and adjacent software supply chain vendors must quickly pivot to new requirements and regulations," she wrote.

However, one reason not to expect the SBOM mandate in its current form to have any significant impact from the outset is the current maturity of SBOMs. There are multiple SBOM standards that the respective groups backing them are still fleshing out. Also, experts emphasize that for SBOMs to be truly effective, they must integrate with machine-readable VEXs.

"There may be a mandate, but people aren't going to process SBOMs because they don't have the processing tools for them," says Walt Szablowski, founder and executive chairman of Eracent. "When you receive an SBOM, your SBOM process needs to be able to read the SBOMs, which can be quite complex since the whole SBOM is a file structure. And you have to be able to read it, identify the libraries, go onto the Internet, find out where those libraries are managed, which vulnerabilities are published for those libraries, and then associate that with your SBOM."

Adding to the complexity, Szablowski argues that organizations must associate the vulnerabilities rendered in the SBOM with specific software, infrastructure, and endpoints. Few organizations have a precise inventory of those components, he says. And the agencies need the systems to operate; hence he doesn't see widespread enforcement. "They're going to do absolutely nothing," he says. "They can't. They absolutely cannot. Because they're not equipped."

> SBOMs alone won't stop supply chain software incidents.

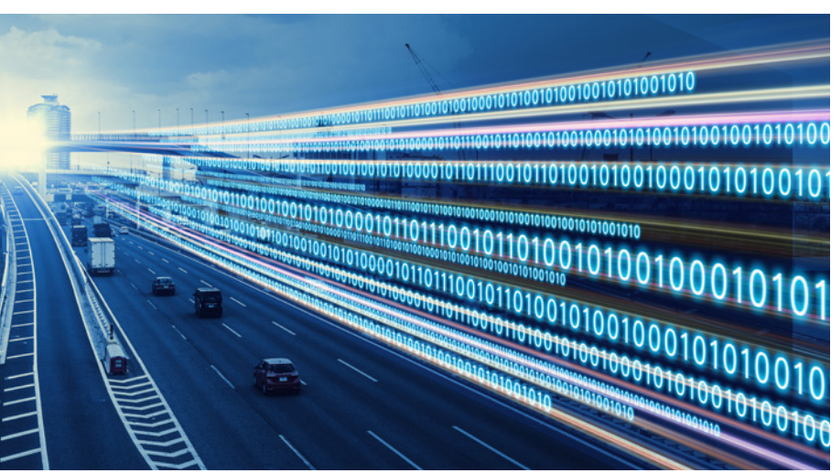
While most experts agree that SBOMs alone won't stop supply chain software incidents, not everyone shares Szablowski's expectations. Some even attest that is not the case. "I can very confidently say that the US government is already receiving SBOMs," says Tim Mackey, head of software supply chain strategy at Synopsys. "Now, whether there are any processes, other than saving the SBOM to some file server or printing it out and saving it into a vault someplace, that's still a TBD. But some agencies have explicitly requested SBOMs as part of their contracts."

Emerging SBOM Standards

HackerOne's Underkoffler adds that assembling SBOMs and operationalizing them is a significant challenge. "SBOMs are not going to fix the problem on their own,"

Underkoffler says. A significant reason SBOMs are challenging to create and operationalize is that there are three standards for generating, sharing, and automating them. The three standards are:

- CycloneDX, an open source project sponsored by the Open Worldwide Application Security Project (OWASP) Foundation, is favored by many experts because it's a modern, lightweight specification explicitly designed for security.
- Software Product Data Exchange (SPDX), a grassroots Linux Foundation project, is a standard (ISO/IEC 5962:2021) designed to communicate the components, licenses, and copyrights associated with a software package.
- Standard for software identification (SWID) tags "defines a structured metadata format for describing a software product," according to NIST. SWID tag documents have a structured set of data components that identify a software product, its version, and the organizations and individuals that created and distributed a product. The documents also describe the software's artifacts and their relationships with other software.



SWID tags have gained some momentum in the past year, according to experts. The first two currently are more popular, and while there is some base interoperability now and more in the pipeline, the different specifications are largely incompatible. As sponsors of the respective projects continue to flesh out the standards, security professionals emphasized the need for SBOMs to integrate with VEXs. Experts say that integrating with VEXs is critical, because it adds context to an SBOM by assessing and classifying the vulnerabilities in software components. In January 2023, CISA's VEX working group voted to publish its "Minimum Recommended Data Elements for VEX" document, building on last year's draft.

Also in January, Anchore, Chainguard, Google, HPE, TestifySec, VMware, and the Linux Foundation introduced the OpenVEX specification. While a VEX is a companion tool to SBOMs, it can be used independently of an SBOM, Chainguard founder and CEO Dan Lorenc wrote in January, when he announced OpenVEX.

"Until today, VEX has been a concept the industry has invested time debating and building minimum requirements around," Lorenc explained. "With the release of OpenVEX, organizations can now put VEX into practice." Lorenc added, "OpenVEX is complementary to SBOMs, allowing suppliers to communicate precise metadata about the vulnerability status of products directly to consumers and end users."

The group developed OpenVEX in collaboration with CISA's VEX Working Group. According to Lorenc, it is the first format to meet the new VEX minimum requirements. Chainguard said it has already moved OpenVEX into production in its Wolfi Linux distribution and Chainguard Images product.

Underkoffler says assembling an SBOM is more complex than scanning and compiling software across an organization's environment. "We all maintain very complex environments, so whether you're producing a product or you're a vendor or a consumer in this space, you're purchasing other people's products and putting them into your environment," she says.

Attention to the SBOM requirements is rippling beyond those who must immediately comply with them because of growing concern over the rise in supply chain attacks. C-level business leaders are increasingly seeking advice on how to respond. "The executive order is being much more scrutinized in board-level conversations, where they want to further understand the security ramifications," says Avanade CSO Justin Haney.

The pending requirements in the executive order aimed at protecting the software supply chain have also showcased the need among organizations to redress their secure development life-cycle processes. "Security should be by design and embedded in all things that we do at any level, including safety and privacy," Haney says.

Risks From Dependencies and Vulnerable Sources

How do software developers unknowingly end up with these malicious packages that they acquired in the software supply chain? The rise in software supply chain attacks is the outgrowth of programmers copying dependencies, reusable libraries, or packages for repeatable functions, rather than writing them. Programmers typically search for the dependencies they need from open source repositories.

The advent of dependency managers has made it possible to automate the download and installation of dependency packages at a much grander scale than once possible. "Before dependency managers, publishing an eight-line code library would have been unthinkable: too much overhead for too little benefit," explained Russ Cox, a Google distinguished engineer, in an Association for Computing Machinery (ACM) post. "NPM, however, has driven the overhead approximately to zero, with the result that nearly trivial functionality can be packaged and reused."

Now there are dependency managers for every major programming language, including Maven Central (Java), NuGet (.NET), Packagist (PHP), PyPI (Python), and RubyGems (Ruby), with each hosting more than 100,000 packages. "The arrival of this kind of fine-grained, widespread software reuse is one of the most consequential shifts in software development over the past two decades," Cox warned. "And if we're not more careful, it will lead to serious problems."

If a developer lacks adequate defenses and their account gets compromised, an attacker can gain supply chain influence over the code that the developer maintains, which then flows into the software that requires it.

Dependency managers have changed the time and cost of open source reuse. Shared code could be as small as individual functions that are just handfuls of lines long, or comprehensive libraries. Either way, it's faster to grab reusable code than to write it from scratch.

Peter Morgan, co-founder and CSO of Phylum, explains a common scenario. "In the typical case, you'll write something, put it on GitHub, and then maybe you'll make a Python package out of it, so that goes on PiPI, and then any developer that wants to use it just installs that through Python Package Manager," Morgan says. "Now you're using the open source code that that developer wrote, and most of the time, it's legitimate open source developers trying to write legitimate software and improve the open source."

But speaking to the "serious problems" foreshadowed by Google's Cox, Morgan warns that a malicious developer can do the same thing, because there are no restrictions on who can join that effort. And if a developer lacks adequate defenses and their account gets compromised, an attacker can gain supply chain influence over the code that the developer maintains, which then flows into the software that requires it.

Types of Software Supply Chain Attacks

Morgan adds that networks of developers aren't using just one package from a single source. "Each of those packages has their own dependencies, which have their own dependencies," he says. "We've seen dependency graphs with 20,000 packages for even a small application. And the supply chain attack surface that extends to all of the

authors in all those repositories, an attacker wins against any of those things. They can inject code into that that will then flow down to anything that relies on it. And here we go with the supply chain."

A typical software supply chain attack method is dependency confusion, where an attacker exploits a vulnerability in a package manager and injects malicious code into a dependency, code that adds to a software component that a programmer uses to avoid repetitive tasks such as writing, testing, and debugging a specific library.

Software developer and security consultant Alex Birsan documented how he uploaded malicious Node packages to the npm Registry under unclaimed names. "From one-off mistakes made by developers on their own machines to misconfigured internal or cloud-based build servers, to systemically vulnerable development pipelines, one thing was clear: squatting valid internal package names was a nearly sure-fire method to get into the networks of some of the biggest tech companies out there, gaining remote code execution, and possibly allowing attackers to add backdoors during builds," Birsan wrote.

The dependency confusion was detected inside more than 35 organizations with over 1,000 employees at the time, across three programming languages. According to open source management platform provider FOSSA, an effective way to defend against dependency confusion attacks is to reserve the package name or namespace on the default public registry, which prevents accidentally exposing a project to a vulnerability when making configuration changes.

Typosquatting is a similar and common form of dependency confusion that many are familiar with. ReversingLabs' Rose notes that it's similar to ransomware attacks, where a user unknowingly clicks on a link in an email or SMS that directs them to a rogue site. The difference with typosquatting is that it uses URL manipulation, where the threat actor creates a URL like the intended site. The goal is to trick developers into accessing or uploading a dependency when they inadvertently mistyped the repository's Internet address.

"If you're under intense pressure to get the next release out in a set period, typosquatting is very simplistic, but it works," Rose says. "People can leverage the open source repositories and find ways to surreptitiously insert compromised packages through processes like typosquatting."

Attacking Organizations' IT Assets

The targets of some of the worst software supply chain attacks have hit the crown jewels of their potential victims' operations: their IT organizations, where they can spread the fastest and have the broadest impact. The SolarWinds and Log4j incidents certainly had that effect. Just days into the new year, CircleCI reported it was the victim of a

software supply chain incident. Software development organizations using CircleCI rely on it for their continuous integration/development processes. CI/CD pipelines enable automated software development incorporating the build, test, and deploy processes. It allows developers to commit their code in small increments continuously.

CircleCI was alerted in late December 2022 that there was suspicious GitHub OAuth activity by one of its customers. An unauthorized third party compromised the customer's GitHub OAuth token. The customer quickly resolved the issue, but "out of an abundance of caution, on December 31, 2022, we proactively initiated the process of rotating all GitHub OAuth tokens on behalf of our customers."

An internal investigation found that an unauthorized third party leveraged malware it had deployed onto a CircleCI engineer's laptop, letting them exfiltrate a valid, 2FA-backed SSO session, CircleCI CTO Rob Zuber wrote in the explanation of the incident. CircleCI discovered that the malware executed the theft of session cookies, allowing the hacker to impersonate the engineer remotely, which let them escalate access to some of its production systems. Zuber noted that CircleCI's antivirus software did not detect the malware on the engineer's computer.

"We have reason to believe that the unauthorized third party engaged in reconnaissance activity," Zuber added. "Though all the data exfiltrated was encrypted at rest, the third-party extracted encryption keys from a running process, enabling them to potentially access the encrypted data."

Zuber advised those who use CircleCI to immediately rotate any secrets stored in their platform. Noting that doing so is known to be disruptive, CircleCI urged its customers to create inventories of the environmental variables in their projects and pipelines. It was also emphasized that secrets shouldn't be deleted but rather revoked to ensure malicious actors couldn't access them. Customers were also advised to rotate other environmental variables, including their SSH keys.

The fact that CircleCI is a CI orchestration platform is significant, ReversingLabs' Rose says, because many organizations use it for their CI orchestration. "If that is compromised, then the thing that's actually compiling your code potentially could be compromised, and the secrets associated with it," he says. "That's a huge problem. But until it's identified, someone could be actively compromising that CI orchestration platform, no matter what it is."

One of the most recent software supply chain incidents in 2023 was reported in late March, when PBX provider 3CX CISO Pierre Jourdan issued an alert to customers and partners that the desktop communications client for Windows and Mac had been compromised. "The issue appears to be one of the bundled libraries that we compiled into the Windows Electron App via GIT," Jourdan wrote. According to Jourdan, the attack appears to be a potentially state-sponsored advanced persistent threat (APT).

3CX says it has 12 million daily users among 600,000 customers, which include Avis, Best Western, Chevron, Honda, Pepsi, PwC, and Wilson. At the time of this report, it was unclear whether the attackers compromised 3CX's build system by injecting malicious source code into its build or if a 3CX developer pulled down a malicious dependency.

ReversingLabs founder and chief architect Tomislav Pericin says the 3CX attack is just another case where developers must scan every dependency in their pipelines early in development. "The effect is the same as

Previous Next

whether it was SolarWinds or 3CX," Pericin says. "In this case, you have all of these organizations which use the software, which now are probably compromised."

Rose adds that the accelerated speed of software release cycles is the most alarming concern. "Our software release cycles used to be once every three months," he says. Now, you're releasing 100 times a day. That needs to be taken into account, because one build or code change could potentially introduce risk in terms of software supply chain security."

About the Author: *Jeffrey Schwartz is a journalist who has covered information security and all forms of business and enterprise IT, including client computing, data center and cloud infrastructure, and application development for more than 30 years.*

Window Snyder's Startup Launches Security Platform for IoT Device Makers

Thistle's technology will give device makers a way to easily integrate features for secure updates, memory management, and communications into their products, Snyder says.

By Jai Vijayan, Contributing Writer, Dark Reading

Renowned security expert Window Snyder, whose experience includes helping companies such as Apple, Microsoft, and Mozilla bolster the security of their products, is betting she can do the same thing for Internet of Things (IoT) device manufacturers.

Snyder's company Thistle Technologies is making generally available a new platform that aims to help IoT manufacturers securely deploy updates and implement capabilities for secure communications and memory management into their devices. The Thistle Security Platform will give development teams working for embedded device manufacturers a way to directly incorporate security functionality into their products during the build phase.

Crucial Capabilities

Snyder says the technology is crucial because embedded devices are like fully functional computers that face the same kind of threats that operating systems and applications software do but often don't have basic security mechanisms for protecting against them.

"What we are trying to do is democratize security," says Snyder, who launched

Previous Next

Thistle in early 2021 after a stint as chief security officer at financial technology company Square. The goal is to give IoT and embedded-device makers an infrastructure for quickly adding security functions to their devices without needing to develop it themselves. "These devices have all the same type of threats that general-purpose operating systems have but with a lot less security," she says.

Thistle's set of security tools and services include an update component, a memory allocator, and an integrated memory-safe Transport Layer Security (TLS) stack for secure communications.

The update client, for Linux and Windows-based devices, enables IoT manufacturers to securely deliver signed updates to their device fleet from a single, central location. The updates could include new device features, security functions, and vulnerability fixes. It includes a failover feature that allows a device to return to a last known good state — without having to reboot — in case an update creates problems. The update client also supports vulnerability monitoring and access control capabilities. Thistle's memory allocator manages device memory in such a way as to mitigate buffer overflows and other common memory-related issues.

Automated Updates

When implemented, Thistle's technology will enable IoT devices to receive automated updates the same way that general-purpose operating systems and applications receive updates. When a vulnerability surfaces in a product, or new functionality becomes available for it, the device manufacturer then can securely push the update out centrally to all installed devices, thereby eliminating the need for manual intervention.

In her various stints as a senior security executive at some of the world's largest technology companies, Snyder has contributed to advances in areas such as secure software development life cycles, memory management, and attack surface reduction.

She perceives the technology her company is now bringing to the IoT market as giving resource-strapped device manufacturers a way to integrate baseline security features — such as encrypted communications and memory management capabilities — into their devices. Her hope is that device makers will then leverage her company's platform to build on those features going forward.

Thistle's immediate focus will be on IoT players in key markets such as automotive, power, water, networking, and the industrial sector.

Update mechanisms — when they exist — in the IoT space can be buggy and unreliable, Snyder says. She points to multiple incidents when a bad update bricked a device or caused other problems. One example: a 2017 incident where a bad firmware update bricked hundreds of smart locks from Lockstate that Airbnb was using as part of a program for its hosts. There have been other instances where key fobs and even cars have been bricked because of a faulty update, Snyder notes.

"The tolerance for update mechanisms is incredibly low," Snyder says. "When you have really low tolerance for update failures, you need to have an update mechanism that is highly reliable in addition to being supported."

Integration With Build Environments

The new Thistle security platform integrates with build environments and provides developers with tools such as those for integrating Thistle's security features into their devices and for things like signing and processing updates. Thistle's platform integrates with the open source Yocto build system, which allows developers to add features to Linux products relatively quickly. It also integrates with the OpenWrt router operating system and with the U-Boot open source bootloader.

Chris Wysopal, founder and chief technology officer at Veracode and seed investor in Thistle, says many of the capabilities that the company is making available are new to the space — especially among smaller IoT device makers. The technology should help embedded device makers implement a secure-by-design approach where key security features get integrated into the product.

"Thistle is making it easier for people to incorporate this technology at a price point they can afford," Wysopal says. "It is changing the market by making security functionality available where it wasn't before."

Thistle's platform launch comes at a time when interest in technologies for securely updating IoT devices appears to be increasing. In recent years, vendors and security researchers have been reporting a growing number of vulnerabilities in IoT products.

A report from Claroty last year showed that in the first half of 2022, IoT vulnerabilities accounted for 15% of all vulnerabilities in the so-called Extended IoT (XIoT) comprised of all connected cyber-physical systems. In the previous six-month period, IoT vulnerabilities accounted for just 9% of all XIoT vulnerabilities.

Pressure Mounts on Device Makers

The trend is significant because organizations across industries such as transportation, telecommunications, manufacturing, and other sectors are connecting all sorts of embedded devices to their networks to support digital transformation and operational requirements.

"The devices have a unique profile because they are not a general-purpose computer and yet they have a processor, memory, are connected to the network, and a lot of the time are doing something critical," Wysopal says.

He expects that enterprise organizations are going to increasingly demand better security capabilities from their IoT suppliers. The availability of technologies like that from Thistle is going to make it harder for device manufacturers to explain away their failure to implement fundamental security mechanisms in their products, Wysopal says.

Recently, the National Institute of Standards and Technology released a new encryption standard for IoT devices, which means enterprise organizations and consumers could soon begin expecting device makers to implement it in their products.

Measures like the Internet of Things Cybersecurity Improvement Act of 2020 are another factor because they require organizations selling IoT devices to government agencies to ensure minimum security standards for their technologies.

Embedded and IoT device makers are feeling more pressure than before to respond to security threats, Snyder says.

"Customers are also asking better questions and there have been more and more demonstrations over time that these devices are deeply vulnerable," she says.

About the Author: *Jai Vijayan is a seasoned technology reporter with over 20 years of experience in IT trade journalism. He was most recently a Senior Editor at Computerworld, where he covered information security and data privacy issues for the publication.*

Toyota Global Supply Chain Portal Flaw Put Hacker in the Driver's Seat

The automaker closed a hole that allowed a security researcher to gain system administrator access to more than 14,000 corporate and partner accounts and troves of sensitive data.

By Elizabeth Montalbano, Contributing Writer, Dark Reading

An ethical hacker found a backdoor in a Web app used by Toyota employees and suppliers for coordinating tasks related to the automaker's global supply chain, gaining control of the global system merely by knowing the email address of one of its users. Attackers can use DNS as a command-and-control (C2) channel to communicate with these networks through DNS servers connected to the Internet, and thus breach them even when an organization believes the network is successfully isolated, the researchers revealed.

Security researcher Eaton Zveare revealed that in October, he found the backdoor login mechanism in the Toyota Global Supplier Preparation Information Management System (GSPIMS) Web portal, a site used by Toyota employees and their suppliers to coordinate various business activities. The backdoor allowed him to log in as any corporate user or supplier.

From there he found a system administrator email and logged in to their account, thus gaining "full control over the entire global system," he explained in a blog post about the hack.

Once acting as an administrator, Zveare said he had "full access" to internal Toyota projects, documents, and user accounts, including some of those that belonged to

Toyota external partners and suppliers, such as Michelin, Continental, Stanley Black & Decker, and Harman.

All in all, the researcher gained read/write access to Toyota's global user directory of more than 14,000 users. Zveare also could access corporate user account details, confidential documents, projects, supplier rankings/comments, and other sensitive data related to those users, he said.

Significant Supply Chain Threat

The hack demonstrates once again how a simple, overlooked flaw in an enterprise system can inadvertently give an attacker access to sensitive data and corporate accounts of a company's supply chain. This, in turn, paves the way for malicious activity that affects not only that organization but its entire ecosystem of partners, security experts noted.

Indeed, had a threat actor discovered the issue before him, "the consequences could have been severe," Zveare observed.

The issue could have allowed attackers to create their own user account with an elevated role to retain access should the issue ever be discovered and fixed, or download and leak all the data to which they had access, he said.

They also could have deleted or modified data in a way to be disruptive to global Toyota operations, or crafted a highly targeted phishing campaign to attempt to capture "real corporate login details, which could have exposed other Toyota systems to attacks," Zveare wrote.

The researcher reported the issue to Toyota on Nov. 3 and the company reported back 20 days later that it had been fixed — a speedy response with which Zveare was "impressed," he said.

"Out of all the security issues I have reported so far to various vendors, Toyota's response was the fastest and most effective," he said.

Zveare revealed his research nearly a year after Toyota suffered a major supply chain breach that subsequently forced it to halt production of all 28 lines of its 14 plants in Japan. On Feb. 22, 2022, the company reported a cyberattack causing a "system failure" at supplier Kojima Industries that created problems with its just-in-time production control system.

Fortunately for Toyota, the latest breach was an ethical one and, thanks to Zveare's responsible disclosure, the company could fix it before there was any impact on the company or its partners' business, notes one security professional.

"Not all 'breachers' are as responsible as in this case!" observes Henning Horst, CTO of data security specialists at Comforte AG.

How It Was Done

Zveare's journey to finding the backdoor wasn't completely straightforward, he acknowledged in his post. Initially he wasn't even sure if the portal — which he said was created and maintained by Toyota — was a very important entity for the company.

To access the system, first he had to patch JavaScript code of an initial login screen that asks a user to click on a button to identify with which Toyota business they are affiliated. In a previous incident in which he hacked into the Jacuzzi SmartTub app, this action was all that was

needed to achieve full access to the network due to an improperly secured API.

However, the GSPIMS API appeared to be secure, which inspired Zveare to further dig into the application code to see what else might be cooking. What he eventually found was that JSON Web Tokens — or session tokens representing the users' valid authenticated sessions on the website — were being generated based on a user's email without requiring a password.

Zveare Googled for Toyota supply chain users and made an educated guess to formulate the email of someone who he thought would be a user of the GSPIMS portal. "Then I fired off the createJWT HTTP request, and it returned a valid JWT!" he wrote.

His discovery gave him the ability to generate a valid JWT for any Toyota employee or supplier registered in GSPIMS, "completely bypassing the various corporate login flows, which probably also enforce two-factor authentication options," Zveare wrote.

Though the user whose email he accessed the system with did not have system administrator privileges, he eventually searched within the GSPIMS to find the email of someone who did, and using that he gained full control of the system as an administrator.

A Big-Picture Security Approach

Enterprises have work to do to in order to block the issue Zveare found, security experts say. For starters, security administrators must take a more holistic approach to security and realize the wider impact their overall security posture — or lack thereof — can have on all of the partners and customers with whom they do business.

"What are perceived as 'internal systems' to organizations, no longer are," Dror Liwer, co-founder of cybersecurity firm Coro said in an email statement to Dark Reading. "With partners, suppliers, and employees collaborating via the Internet — all systems should be considered external, and as such, protected against malicious intrusion."

Developing this big-picture perspective and security strategy is not so simple, as most enterprises already have their hands full managing their own company's risk, notes Lorri Janssen-Anessi, director of external cyber assessments for BlueVoyant.

However, considering how easy it was for Zveare to gain access to a system that serves Toyota's global supply chain, companies need to get their heads around this risk to maintain security across any third party that touches their network, she says.

"What today's organizations should take from the reported vulnerability in Toyota's supplier management network is a firm reminder to look at their own vendor and supplier cybersecurity," Janssen-Anessi says.

Among the key measures to consider include shoring up access control and user account privileges, ensuring that they only provide employees and third parties with access to the data needed for their particular role, she notes. "This helps to control what data can be accessed in the event of a breach," Janssen-Anessi says.

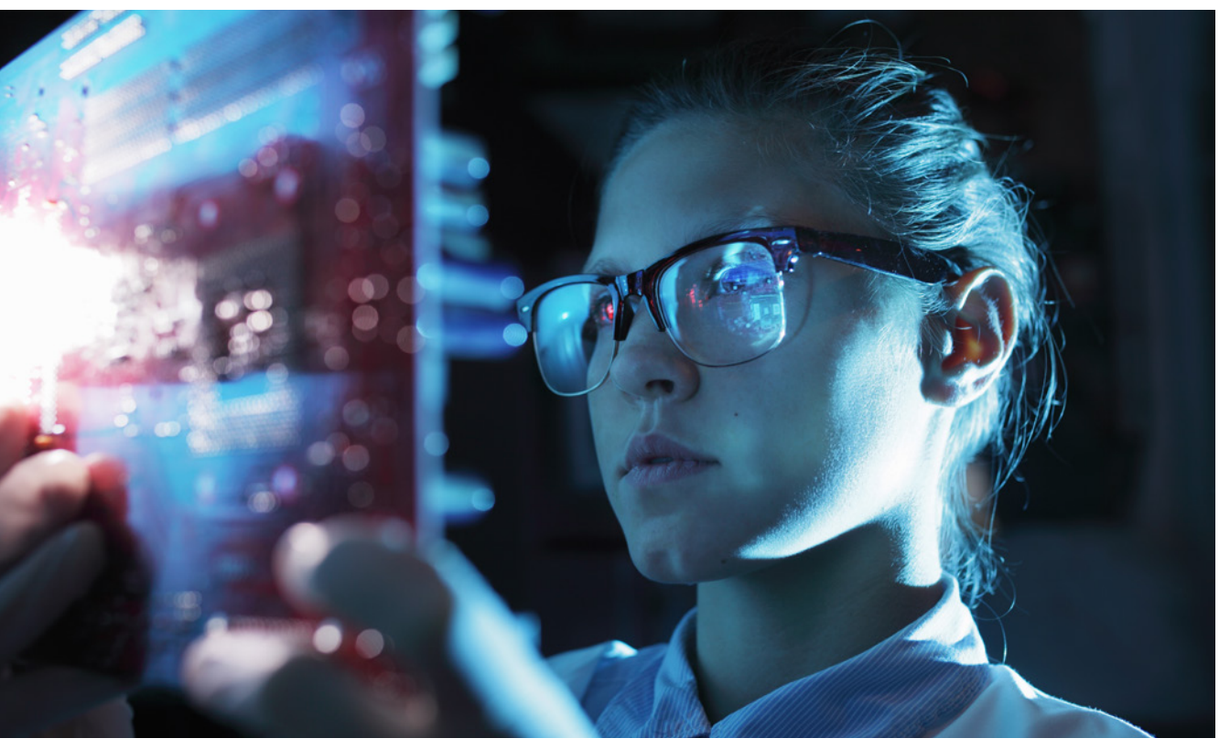
Indeed, a more data-centric approach overall to security could help enterprises avoid or mitigate a scenario that Zveare demonstrated, Comforte AG's Horst observes. He advises that organizations find ways to protect data as soon as it enters their corporate data ecosystem, thus protecting "the data itself rather than perimeters and borders around the data."

About the Author: *Elizabeth Montalbano is a freelance writer, journalist, and therapeutic writing mentor with more than 25 years of professional experience. Her areas of expertise include technology, business, and culture.*

Firmware Flaws Could Spell ‘Lights Out’ for Servers

Five vulnerabilities in the baseboard management controller (BMC) software used by 15 major vendors could allow remote code execution if attackers gain network access.

By Robert Lemos, Contributing Writer, Dark Reading



Five vulnerabilities in the baseboard management controller (BMC) firmware used in servers of 15 major vendors could give attackers the ability to remotely compromise the systems widely used in data centers and for cloud services.

The vulnerabilities, two of which were disclosed in January by hardware security firm Eclypsium, occur in system-on-chip (SoC) computing platforms that use AMI’s MegaRAC BMC software for remote management. The flaws could impact servers produced by at least 15 vendors, including AMD, Asus, ARM, Dell, EMC, Hewlett-Packard Enterprise, Huawei, Lenovo, and Nvidia.

Eclypsium disclosed three of the vulnerabilities in December, but withheld information on two additional flaws until January in order to allow AMI more time to mitigate the issues.

Since the vulnerabilities can only be exploited if the servers are connected directly to the Internet, the extent of the vulnerabilities is hard to measure, says Nate Warfield, director of threat research and intelligence at Eclypsium.

“We really don’t know what the what the blast radius is on this, because while we know some of the platforms, we don’t have any details as to [how] prolific these things

Previous Next

are," he says. "You know, did they sell 100,000 of them? Did they sell 10 million of them? We just don't know."

Baseboard management controllers are typically a single chip — or system-on-chip (SoC) — installed on a motherboard to allow administrators to remotely manage servers with near total control. AMI's MegaRAC is a collection of software based on the Open BMC firmware project, an open source project for developing and maintaining an accessible baseboard management controller firmware.

Many server makers rely on BMC software to allow administrators to take complete control of the server hardware at a low level, giving it access to "lights-out" features, the Eclypsium advisory stated. Because the software is widely used, the footprint of the vulnerable features is quite large.

"[V]ulnerabilities in a component supplier affect many hardware vendors, which in turn can pass on to many cloud services," Eclypsium stated in its advisory. "As such these vulnerabilities can pose a risk to servers and hardware that an organization owns directly as well as the hardware that supports the cloud services that they use."

AMI is the latest BMC software maker to have vulnerabilities found in their code. In 2022, Eclypsium also found vulnerabilities in Quanta Cloud Technology (QCT) servers that have found common use by cloud firms. And previous research by the company in 2020 found that the lack of signed firmware in laptops and servers could allow an attacker to install a Trojan horse to remote control the devices.

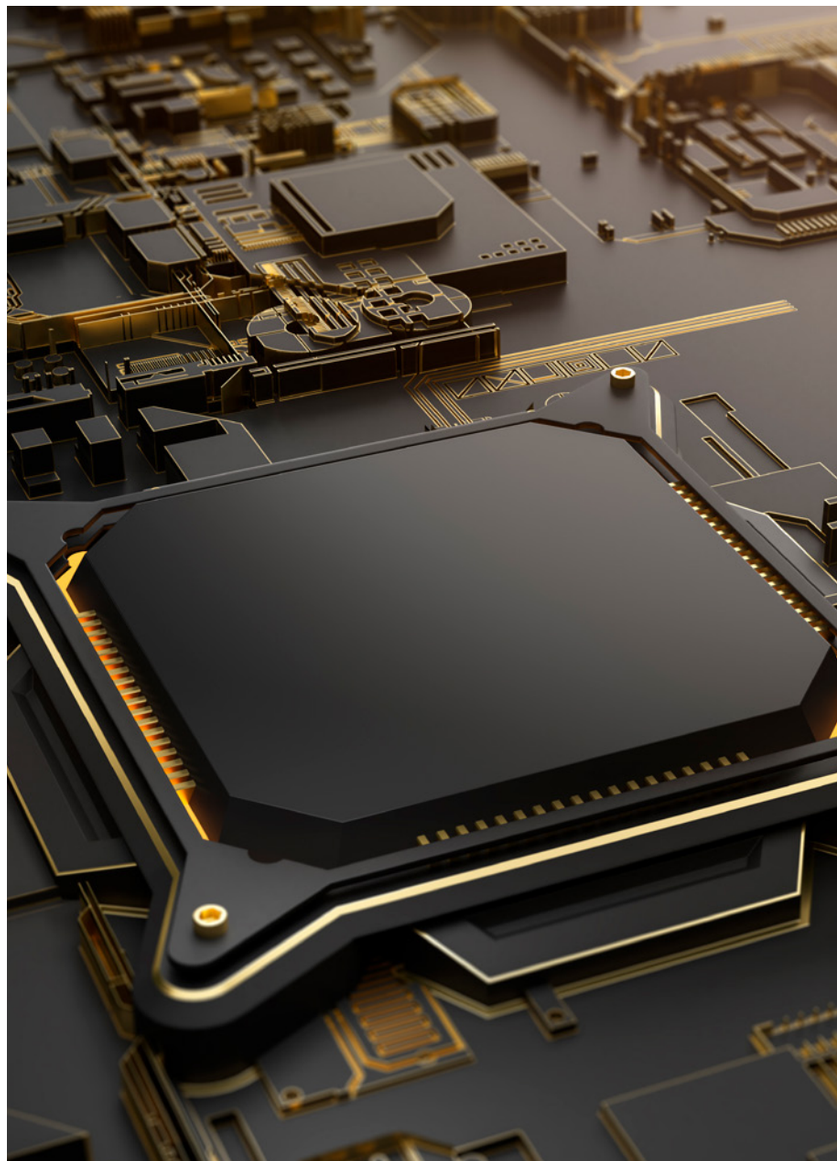
December Flaws Most Serious

The two latest flaws released on Jan. 30 include two lower severity issues. The first vulnerability (CVE-2022-26872) gives an attacker the ability to reset a password if they can time the attack during a narrow window between when a one-time password is validated and when the new password is sent by the user. In the second security issue (CVE-2022-40258), the password file is hashed with a weak algorithm, Eclypsium stated.

Both issues are less severe than the three vulnerabilities disclosed in December, which include two vulnerabilities — a dangerous command in the BMC's API (CVE-2022-40259) and a default credential (CVE-2022-40242) — that could allow simple remote code execution, Eclypsium stated in the advisory. The other vulnerability (CVE-2022-2827) allows an attacker to remotely enumerate usernames via the API.

The Redfish API replaces previous versions of the Intelligent Platform Management Interface (IPMI) in modern data centers, with support from major server vendors and the Open BMC project, according to Eclypsium.

Eclypsium conducted its analysis of the AMI software after the code was leaked to the Internet by a ransomware



group. AMI is not thought to be the source of the leaked software code; rather, the code is a result of a third-party vendor being hit by ransomware, Warfield says.

"What we've discovered back in the summer was that somebody had leaked intellectual property for a bunch of technology companies onto the Internet," he says. "And, as we were digging through it ... trying to figure out what it was and who had it, we came across some of AMI's intellectual property. So we kind of started digging into that to see what we could find."

Patching Rate Unknown

AMI has issued patched software for all five vulnerabilities, and now the mitigation of the vulnerabilities is in the hands of server makers and their customers.

Already, many vendors — such as HPE, Intel, and Lenovo — have issued advisories to their customers. However, patching those servers will be up to the companies who have the servers deployed in their data centers.

Firmware patching tends to happen at a glacial rate, which should be a worry, says Warfield.

"The tricky part is the time between the patches coming out and people actually applying them," he says. "BMC is not something with, sort of, a Windows update mechanism, where you can say, 'Oh, I've got 100,000 servers that are affected. Let me just push this out to all of them.'"

About the Author: *Rob Lemos is a veteran technology journalist of more than 20 years and a former research engineer. He has written for more than two dozen publications, including CNET News.com, Dark Reading, MIT's Technology Review, Popular Science, and Wired News.*

MEND PERSPECTIVES

The Growing Threat of Malicious Package Attacks

Malicious packages in your application can cause a wide range of issues, including data theft, remote code execution, and network intrusion.

By Jeanette Sherman, Senior Product Marketing Manager, Mend

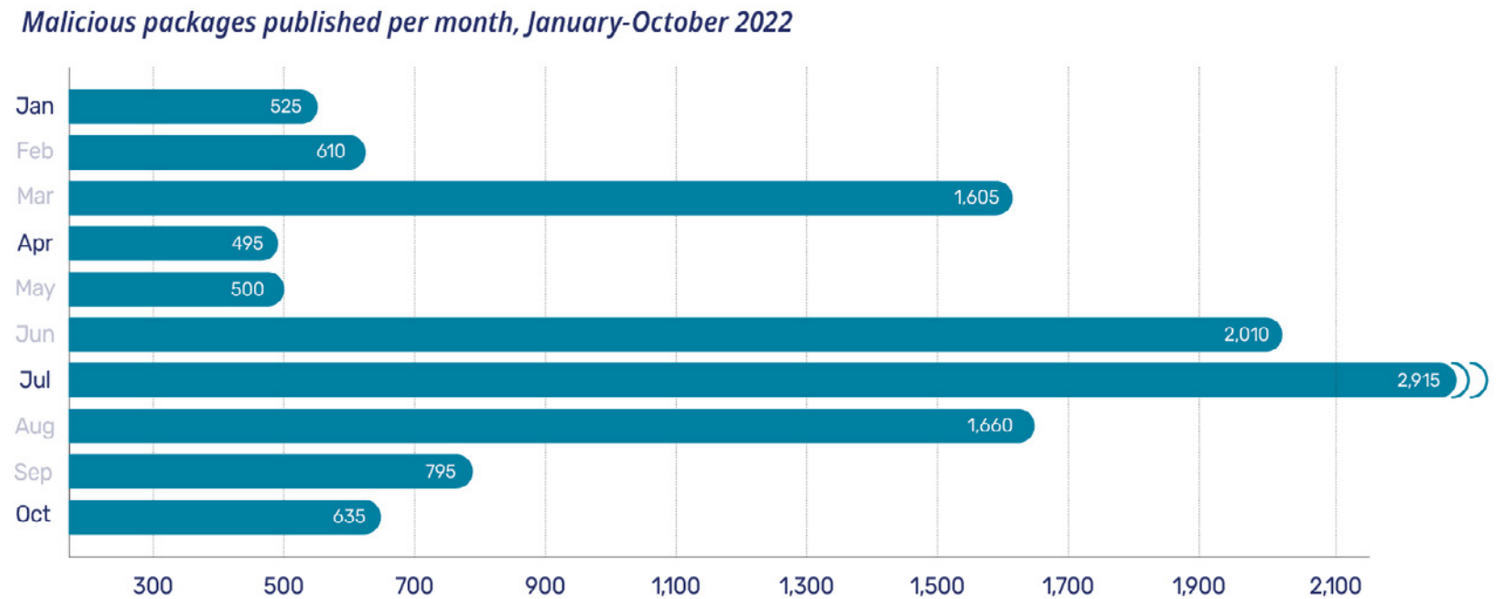
For years, attackers seeking to compromise applications confined themselves largely to exploiting vulnerabilities in custom code. This wasn't ideal for attackers: In order to gain unauthorized access, they needed to find a vulnerability, use it, and do whatever nefarious activity they planned without getting caught. But what if attackers no longer had to look for a vulnerability — because they were invited into your application code by your developers?

That's exactly how malicious package attacks work.

Open source package registries such as npm and RubyGems certainly enable faster application development and improve access to new updates. However, these repositories tend to be maintained and verified by open source communities or consortiums, so there is usually a minimum standard of security associated with package maintenance. That leaves the door wide open for malware, as threat actors publish malicious packages designed to deliver payloads and execute commands via an open source package. Mend's research team has seen a steady growth in malicious package attacks over the past several years.

Common Attack Vectors

The first step is getting someone — or something — to download the malicious package, and attackers generally rely on four basic attack vectors to accomplish that goal: typosquatting, brandjacking, dependency hijacking, and dependency confusion.



Typosquatting: A social engineering attack in which the attacker publishes a malicious package with a name similar to an existing popular package. The idea is that the developer will misspell a package name and unintentionally fetch the malicious version. Alternatively, the developer may find this package, not notice the slightly misspelled name, and fetch this one thinking it is the legitimate package.

Brandjacking: Similar to typosquatting, brandjacking leverages behavior from the victim to gain access to systems or data. With brandjacking, attackers name malicious packages with the goal of making people think the package is associated with a well-known company or package owner. It doesn't necessarily mean the attack stole credentials or otherwise compromised the organization or original project, but that the adversary is taking advantage of an opportunity to seize ownership related to the brand name.

Dependency hijacking: The attacker takes control of the account of a package maintainer of a public repository in order to upload a new and malicious version of the package.

Dependency confusion: Publishing a malicious package in a public repository that has the same name as a package in a private or internal repository. The attacker then uses this so-called feature — having the same package names — to trick dependency management tools into downloading the public malicious package rather than the private, non-malicious package.

Organizational Impact of These Attacks

When attackers gain access to applications via a malicious package, it can impact an organization in multiple ways. How much damage it inflicts will depend on several key factors:

1. **Intent.** When threat actors infiltrate using a malicious package, their intent substantially determines the impact. A threat actor trying to inform people about a war or protesting an action by displaying annoying messages has a lower overall impact than one trying to steal information from the organization or use compromised machines to mine cryptocurrency.
2. **Organization type.** Attacks intended to exfiltrate personal information will have a larger, potentially long-term impact on companies trusted with sensitive data. Attacks that disable systems can have outsized impact in organizations like hospitals, where lives depend on uptime.
3. **Duration.** When malicious packages are discovered quickly and removed completely, the damage they cause can be limited. The greatest damage can be

caused by packages that remain undetected for months or years, while quietly delivering their payload.

4. **Spread.** Some of the most dangerous malicious packages are designed to provide initial access to a network, at which point the threat actor can move laterally through systems to steal passwords or protected information to gain even more access.

Unlike vulnerabilities — which can and do often exist for months or years in application code without being exploited — a malicious package represents an immediate threat to your organization. Think of your applications and organization as your house and attackers as burglars. A vulnerability is a window that's been left unlocked — it could potentially let a burglar in. A malicious package is like accepting a FedEx package that has a burglar inside.

Malicious packages don't enter your code base to do nothing. If they're in your application, your organization already has a problem. Organizations may have different thresholds for acceptable risk from vulnerabilities, but the only acceptable number of malicious packages to have in your application code is zero. The question is: Does your application security strategy have a plan in place to defend against this new threat?

About the Author: *Jeanette Sherman works as a product marketer at Mend to understand the struggles of cybersecurity leaders as they work to secure open source. After a youth spent befriending famous hackers, Jeanette has developed a perspective on cybersecurity that takes into account not only today's business needs, but also the thought patterns of real threat actors.*