



How Supply Chain Attacks Work — and How to Secure Against Them

Software supply chains are increasingly facing attacks that are notoriously challenging to identify and protect against. Here's what security teams need to know.

INSIDE:

How Software Supply Chain Attacks Work – and How to Secure Against Them >>

Cloudflare Falls Victim to Okta Breach, Atlassian Systems Cracked >>

Lock Down the Software Supply Chain With 'Secure by Design' >>

Newly ID'ed Chinese APT Hides Backdoor in Software Updates >>

JFrog Perspectives: Defending the Enterprise from Software Supply Chain Attacks >>

How Software Supply Chain Attacks Work — and How to Secure Against Them

Software supply chains are increasingly facing attacks that are notoriously challenging to identify and protect against. Here's what security teams need to know.

By George V. Hulme, Contributing Writer, Dark Reading

In May 2023, Progress Software was alerted to unusual activity involving its MOVEit secure file transfer service. An investigation showed that attackers were hijacking targeted systems by exploiting a zero-day flaw (CVE-2023-34362) involving a SQL injection vulnerability in MOVEit.

Progress Software notified customers about the attack and quickly released a patch. The US Cybersecurity and Infrastructure Security Agency (CISA) also issued a warning to organizations using the secure file transfer service and urged them to install the security update as soon as possible.

Still, the resulting damage was massive, involving the data of millions of people and affecting more than 1,050 organizations, including those in the federal government, healthcare, education, finance, and insurance. In one incident, the personally identifiable information (PII) of



as many as 11 million people was exposed, including 600,000 Medicare beneficiaries.

Software supply chain attacks pose a growing risk for enterprises because they are widespread and can last for months before being identified. An [Anchore report](#) found that 62% of those surveyed faced at least one supply chain attack in 2021. Similarly, online marketplace Capterra found in its [May 2023 report](#) that 61% of US businesses have been hit directly by software supply chain threats during the past year.

“Software supply chain attacks are at the top of all CISOs’ minds,” says Jeff Music, CISO at ReliaQuest. These attacks are popular, Music adds, because the barrier to success is low for the attacker — and the potential payoff is enormous. “This is especially the case if the vulnerable hardware or software has high adoption across enterprise organizations,” he adds.

The MOVEit attack is one of many software supply chain attacks that have occurred in recent years. One of the most well-known is the SolarWinds incident, which occurred in 2020. Widely believed to have been conducted by a hacking group associated with the Russian Foreign Intelligence Service, the attack involved hijacking SolarWinds' Orion IT monitoring and management software to access the data, networks, and systems of more than 30,000 organizations using the software.

that had already been compromised; and code coverage tool Codecov was used to gain unauthorized access to hundreds of networks.

Common Software Supply Chain Attack Methods

Supply chain attacks are so plentiful and successful because they're stealthy and provide attackers with avenues of entry into many systems. That's true whether attackers target an internal software development pipeline,

“Software supply chain attacks are at the top of all CISOs' minds.” —Jeff Music, ReliaQuest

Other, more recent attacks on organizations running widely used applications and hardware systems include the attack against the PyTorch machine learning framework (December 2022); Applied Materials, which provides equipment, services, and software for the manufacture of semiconductor chips (February 2023); software development tools maker JetBrains (September/October 2023); and identity and access management provider Okta (October 2023).

The list, unfortunately, goes on and on. For example, Apple's development tool Xcode was used to compromise developer workstations; attackers gained unauthorized access to IT services provider Wipro to find accounts

taint an open source application or library with malware, or infiltrate a third-party software provider. Once attackers have penetrated a target, they can subvert applications used by hundreds to thousands of organizations. In many cases, attackers gain persistence in critical applications after compromising continuous development pipelines and stay hidden for a long time.

Common types of software supply chain attacks include:

- **Contaminating third-party software:** Attackers inject malware and attack the tools used in open source software and libraries, and by software service providers, or they modify the source code of third-party applications.



“An increasingly common type that we expect to see more of in 2024 is attacks on third-party enterprise tools, such as software services, especially managed file transfer solutions,” says John Fokker, head of threat intelligence at security firm Trellix. The attacks on SolarWinds and MOVEit are examples of this type of software supply chain attack.

developers use, and even leverage insider threats.

Mike Kiser, director of standards and strategy at SailPoint Technologies, advises organizations not to underestimate social engineering attacks targeting these environments. “Social engineering will always be popular, as humans can be induced to introduce patches and updates that propagate downstream

“Humans are not the weakest link, per se, but they are a link in the security of supply chains.”

—Mike Kiser, SailPoint Technologies

- **Infiltrating development pipelines:** Threat actors often plant malware within enterprise and software service providers’ continuous development pipelines. For example, by “Trojanizing” software update binaries or libraries within software repositories, or the tools and other dependencies within these pipelines, attackers can gain a foothold within the enterprise to tamper with applications as they are being developed. If third-party software is compromised in this manner, customers who use the application will be affected.

Threat actors look for gaps in security defenses and policies within the victim’s development environment. Likewise, they weaponize open source components within public repositories, target the endpoints

and do harm. Humans are not the weakest link, per se, but they are a link in the security of supply chains,” Kiser says.

- **Package typosquatting:** With [package typosquatting](#), attackers mimic the name of a popular package on a public registry. The hope is that users or developers will accidentally download the package including malware instead of the legitimate one.
- **Dependency confusion attacks:** Dependency confusion attacks involve registering the name of a private package on public package repositories such as PyPI or npm. Dependency confusion occurs when a project relies on private packages from an internal registry. “This means that when software is built, it



may be sourced in the compromised public library rather than the local version, especially if the public version appears to be more up to date because of release number and similar,” Kiser says.

“Dependency confusion attacks target the maintainer or the method in which open source packages are consumed,” says Chris Hughes, co-founder and president at Aquia and cyber innovation fellow for supply chain security at CISA.

Hughes notes that dependency confusion attacks, which usually target internal packages hosted in private repositories rather than public open source libraries, were among the most prominently used attack vectors in 2022. “Crafting and automating such

attacks is relatively cheap because attackers don't need to interfere with any existing, legitimate project resources," Hughes says.

Defending the Third-Party Software Supply Chain

Protecting the organization from supply chain attacks involving third-party software is challenging because security teams don't control the application environment. It's therefore crucial to build a strategy that considers the risks associated with third-party applications, components, and libraries, as well as software services. The strategy must consider how supply chain attacks work and incorporate strategies mitigating these threats, including:

- 1. Patch and update software — carefully.** Good security policy and hygiene still apply, including patching and updating software. However, patching itself must be carefully considered because software patches can be used as an attack vector. Applying updates that have been tampered with means installing malware on the system. Additional precautions — such as putting all updates through a testing and approval process before applying them and vetting dependencies and updates with software composition analysis tools — has become a necessary part of the process.
- 2. Review your vendor's risk management program.** With the growing number of attacks targeting



third-party applications and services, it's no longer enough for vendor risk management programs to ask questions and check off boxes. "You need to assess vendors' security practices thoroughly," says Damir J. Brescic, CISO at Inversion6. "This includes their software development life cycle, security controls, and incident response capabilities," Brescic says.

If you are just beginning a vendor risk management program, experts advise, rely on periodic assessments, contractual reviews, and continuous monitoring. A threat intelligence program can also make a difference.

And with the use of third-party vendors increasing,

many organizations are stuck managing vendor risk management programs that don't scale, says Jake Olcott, vice president of government affairs at Bitsight. "As organizations onboard more vendors, vendor risk management becomes weighed down by manual, inefficient, and time-consuming processes. Automating the assessment process can alleviate pressure and provide time savings," Olcott says.

For such automation, consider turning to vendor risk management service providers, services that evaluate the security posture of third parties, and outsourced threat intelligence services. Experts say it's a balance of need and internal resources.

- 3. Embrace security frameworks.** Applying risk security frameworks will help organizations maintain visibility into their software supply chains, including understanding the software components, activities, and practices involved in creating and deploying software. "This includes assessing adherence to industry frameworks like [NIST 800-218](#), certifications, and ISO reports," says John Hernandez, president and general manager at Quest Software.

Hernandez adds that when working with highly regulated sectors, additional checks should be considered, such as applying air gapping, auditing code externally, monitoring, and leveraging vulnerability management and response systems.

“SaaS-based businesses should also prioritize data privacy, access controls, and operational procedures. Basic identity security hygiene is crucial, as limiting access and monitoring for unusual activity can mitigate risks regardless of the thoroughness of third-party vetting,” Hernandez says.

4. Perform software composition analysis. Software composition analyzers help manage the security, quality, and license compliance risks of using open source and third-party code. Software composition analysis provides insight into the open source dependencies applications are built on, providing a software bill of materials (SBOM). “An SBOM lists the components of a software package so that organizations can identify if their supply chain applications might be vulnerable when new exploits are disclosed. The reality is that no vendor is entirely immune to an attack, but taking the time to evaluate the security postures of your suppliers and requiring an SBOM can greatly reduce the chance of exploitation,” says Mike Horn, senior vice president and general manager of security at Splunk.

5. Deploy application behavioral monitoring tools. User and entity behavior analytics (UEBA) tools continuously monitor and understand typical application behaviors so they can identify potentially malicious behaviors. UEBA tools look for any



deviations from the norm, such as spikes in data requests, access attempts at odd times or from unknown devices, and behavior that may indicate malware is present, such as replication.

Protecting Continuous Development Pipelines

Supply chain attacks don't target just popular software hardware suppliers and technology service providers; they can also target internal enterprise development pipelines.

Keeping the development pipeline secure from supply chain attacks involves good DevSecOps practices with an adequate focus on third-party components. This typically entails embedding security checks throughout the software development process. Security testing should be automated, with continuous vulnerability scanning

and auditing.

For a sizable portion of businesses, “the question is not if they will be victims of a supply chain attack, it's when,” Splunk's Horn says. “Securing development pipelines is vital to reducing the likelihood of a supply chain attack, and implementing observability practices is the key to keeping development pipelines secure.”

Organizations can use effective observability practices and tools to collect and examine data from various development pipelines to identify suspicious behavior. This practice, notes Horn, allows organizations to seamlessly integrate their security and developer operations, enhancing visibility across their networks.

“Like any environment, examining the development pipeline and identifying potential weaknesses is important.

Much of this is knowing what sources and packages are being pulled from,” SailPoint’s Kiser says.

And not only should organizations stay current on security advisories from maintainers of both proprietary and open source software libraries, but regular two-way communication between security teams and maintainers also may be warranted. The bottom line is that vigilance is essential — casually accepting a GitHub pull request, for example, can compromise an “otherwise resilient pipeline,” Kiser adds.

It’s essential to ensure development tools are secured, access is tightly managed, and all components are verified. Organizations should plan to conduct thorough code control and review, as well as adopt a trust-but-verify approach to all code sources. Rigorous auditing processes by internal and external parties, as well as applying an air gap between development teams and cycles, are essential measures employed to safeguard the integrity of the development pipeline at every stage, Quest’s Hernandez says.

Careful attention must be paid to protecting upstream components. Enterprise development teams can employ several strategies to ensure trust in the upstream components of their code. One of the most important steps is to understand and track the heritage of all components. “Development teams establish trust in upstream components in their code by adhering to



a comprehensive framework that ensures security, vulnerability, and data privacy controls are met — both for external and internal processes. Code control and review,” Hernandez says.

That means vetting software suppliers and ensuring they develop components that meet all security and functionality requirements. Further, as changes are made and upstream components are modified, these changes should be documented to maintain the heritage of the element.

“Most supply chain attacks happen upstream,” Bitsight’s Olcott says. Typically, it’s an attack against the components or the development environment. One way to ensure trust

in upstream components is to assess each product. While this will incur additional costs at the beginning, it will wind up saving time, money, and legal headaches in the long run, Olcott adds.

To secure internal continuous development pipelines and DevOps teams, organizations should:

- **Understand the threat landscape.** Be aware of current vulnerabilities in the tool sets being used by DevOps teams. This infrastructure supports the development pipeline and stays current on applicable threat intelligence to defend against potential attacks, such as when a development tool or service provider announces that a zero-day vulnerability or a compromise is underway.
- **Adopt continuous testing and continuous monitoring.** Organizations must build continuous testing into the pipeline and perform continuous monitoring of the pipeline. This means ensuring that application and code security tests occur early in — and throughout — the development process. This includes, at minimum, implementing static application security testing, dynamic application security testing, and software composition analysis. While these practices help protect internally developed applications, enterprises must also continuously monitor their toolsets and infrastructure for vulnerabilities and potential compromise to

protect against supply chain attacks.

- **Ensure builds are cryptographically signed.** The code being compiled and built must have 100% integrity during the build process. Ofir Asulin Ichin, cyber intelligence analyst at Cyberint, explains that cryptographic techniques can help here. “By generating a cryptographic digest of both the source code in the repository and the source code that is being compiled in the build pipeline, you can verify that the hashes are identical to ensure that the code has not been tampered with,” Ichin says.
- **Implement DevSecOps principles.** Building cooperation among development, security, and operations teams is vital to effectively fending off attacks and mitigating vulnerabilities.
- **Implement policy as code.** Wherever possible, define and enforce security and internal policies as code so teams can automate the enforcement of their policies in a scalable way. This reduces the chances of human and machine error, and of threat actors gaining access, and helps enforce standards.
- **Instill a zero-trust strategy.** Zero-trust principles aren’t just for business software and resources; they also should apply — maybe even primarily — to developers and the entire development pipeline and infrastructure. This will help reduce the risk of unauthorized access within the development

environment and supporting infrastructure.

- **Use software bills of materials.** An SBOM is a manifest of all the components comprising a product. SBOMs aid in the verification and security vetting of software and the identification of rogue components. Regularly analyze code and version history to identify and mitigate irregularities.

By implementing these practices, organizations can strengthen the security of their continuous development pipelines and DevOps teams.

Trust-but-Verify Security

Trellix’s Fokker stresses the importance of ferreting out vulnerabilities throughout the supply chain, focusing closely on the code base and using caution when referencing open source code. “Vulnerabilities buried in frequently used code can become widespread supply chain issues down the line if they aren’t patched in time,” Fokker says. As an example, he points to a vulnerability in Python’s tarfile module that was left unchecked for 15 years until it was discovered in 2022, which resulted in urgent patching at a massive scale.

“This requires continuous vigilance since new vulnerabilities are discovered frequently, which means you can’t rely on last week’s testing to ensure that today’s source code is safe,” Splunk’s Horn adds.

The stakes are high, and CISOs and security teams grappling with these risks must adopt comprehensive



security strategies — including vendor risk management, security frameworks, software composition analysis, and DevSecOps best practices — to build a strong defense. By understanding the threat landscape, continuously monitoring and testing, and embracing a zero-trust approach, organizations can fortify their defenses against the supply chain attack threat. The collective effort of the cybersecurity community in patching vulnerabilities and sharing knowledge is also critical to mitigating risks and ensuring the integrity of the global software supply chain.

About the Author: For more than 20 years, George V. Hulme has written about business, technology, and cybersecurity topics. He currently focuses on cybersecurity and digital innovation. Previously, he was senior editor at InformationWeek magazine.

Cloudflare Falls Victim to Okta Breach, Atlassian Systems Cracked

The cyberattackers, believed to be state sponsored, didn't get far into Cloudflare's global network, but not for lack of trying.

By Tara Seals, Managing Editor, News, Dark Reading

Cloudflare was a victim of the wide-ranging Okta supply chain campaign in fall 2023, with a data breach impacting its Atlassian Bitbucket, Confluence, and Jira platforms beginning on Thanksgiving day.

“Based on our collaboration with colleagues in the industry and government, we believe that this attack was performed by a nation-state attacker with the goal of obtaining persistent and widespread access to Cloudflare's global network,” the Internet security and DDoS protection company said in a [blog post on the Okta-related cyber incident](#).

Cyberattackers Looked for Lateral Movement Options

Cloudflare worked with CrowdStrike and was able to determine that, after initial reconnaissance work, cyberattackers accessed its internal wiki (Confluence) and bug database (Jira) before establishing persistence on its Atlassian server. From there, the perpetrators poked around for places to pivot into, successfully puddle-hopping into the Cloudflare source code management system (Bitbucket) and an AWS instance.

The analysis showed that the cyberattackers were “looking for information about the configuration and management of our global network, and accessed various Jira tickets ... relating to vulnerability management, secret rotation, MFA bypass, network access, and even our response to the Okta incident itself.”

But they were largely shut out of other systems they tried, like a console server that had



access to a dormant data center in São Paulo, Brazil.

In all, the unknown assailants “accessed some documentation and a limited amount of source code,” but no customer data or systems, according to Cloudflare, thanks to network segmentation and the implementation of a zero-trust authentication approach that limited lateral movement.

Nonetheless, the firm erred on the side of caution: “We undertook a comprehensive effort to rotate every production credential (more than 5,000 individual credentials), physically segment test and staging systems, performed forensic triages on 4,893 systems, reimaged and rebooted every machine in our global network including all the systems the threat actor accessed and [all Atlassian products](#) (Jira, Confluence, and Bitbucket).”

“This ... attack on one of the largest [software-as-a-service] companies ... severely highlights the risks of supply chain attacks,” says Tal Skverer, research team lead for Astrix Security. “In this breach, we again see how non-human access is abused by attackers to achieve high privilege access to internal systems which goes unmonitored. We also see how attackers are targeting both cloud, SaaS, and also on-prem solutions to expand their access.”

Yet Another Okta Breach Victim

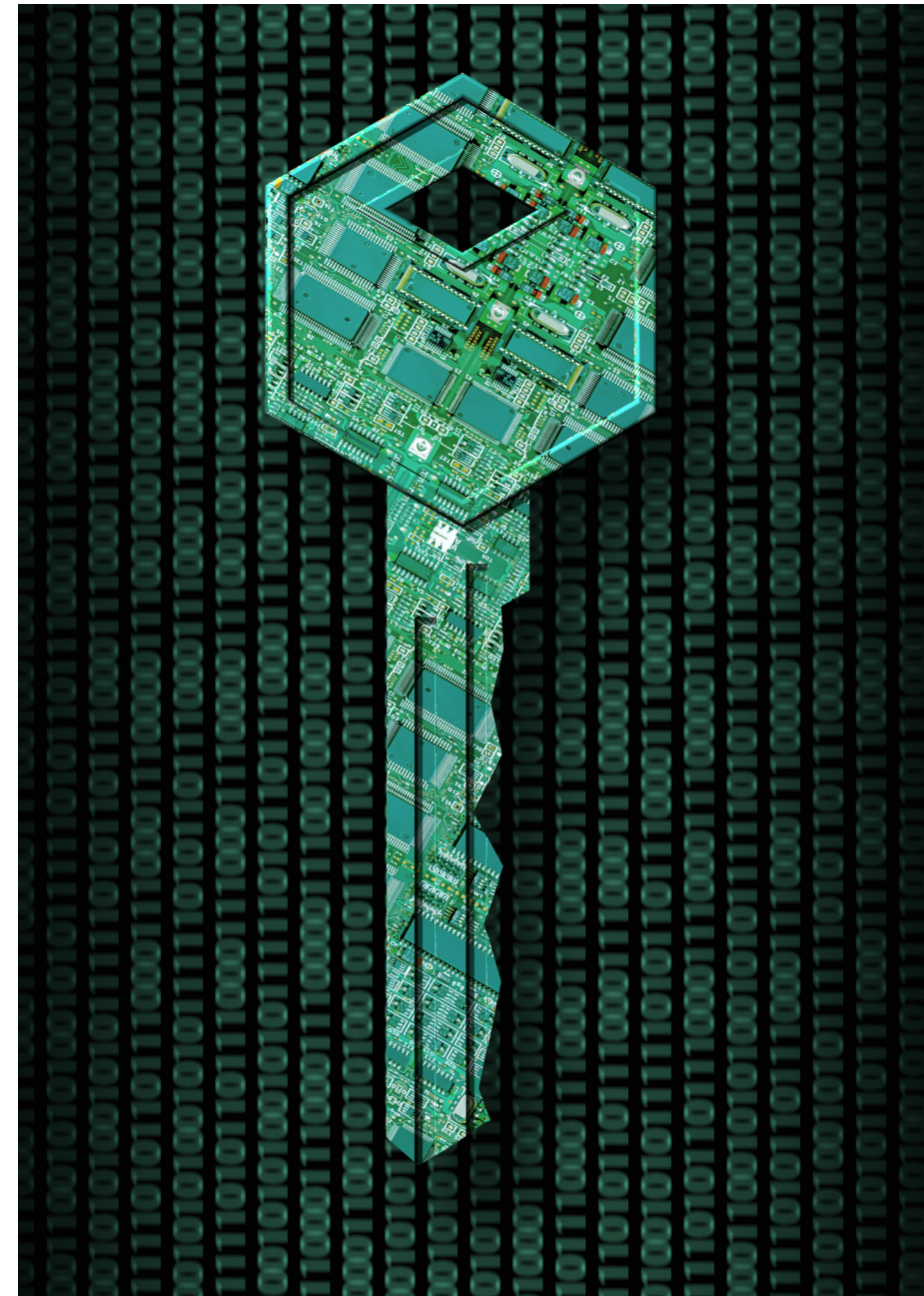
In October 2023, Okta, the identity and access management services provider, disclosed that its [customer support case management system was compromised](#), exposing

sensitive customer data including cookies and session tokens, usernames, emails, company names, and more. Initially the company said that [less than 1% of its customers](#) were affected (134 in all), but in late November the company [widened the number to a staggering 100%](#).

“They [achieved compromise] by using one access token and three service account credentials that had been taken, and that we failed to rotate, after the Okta compromise of October 2023,” according to Cloudflare. “All threat actor access and connections were terminated on November 24 and CrowdStrike has confirmed that the last evidence of threat activity was on November 24 at 10:44.”

An Okta spokesperson tells Dark Reading: “This is not a new incident or disclosure on the part of Okta. On Oct. 19, we notified customers, shared guidance to rotate credentials, and provided indicators of compromise (IoCs) related to the October security incident. We can't comment on our customers' security remediations.”

About the Author: Tara Seals has more than 20 years of experience as a journalist, analyst, and editor in the cybersecurity, communications, and technology space. Prior to Dark Reading, Tara was editor-in-chief at Threatpost, and, prior to that, the North American news lead for Infosecurity Magazine. She also spent 13 years working for Informa (formerly Virgo Publishing) as executive editor and editor-in-chief at publications focused on both the service provider and the enterprise arenas.



Lock Down the Software Supply Chain With ‘Secure by Design’

As zero-days and complex networks create gaps for cyberattacks, software developers and agencies, such as CISA, look to the concept for building in defenses.

By Nathan Eddy, Contributing Writer, Dark Reading

Software that prioritizes security at its most foundational level means designing it with customer security as a key goal rather than a tacked-on feature. And that concept — secure by design — is becoming increasingly crucial as attackers begin targeting supply chains more frequently.

“They understand that they can have a bigger impact by successfully exploiting the supply chain,” says Thomas Pace, CEO of NetRise.

Because traditional security solutions, such as endpoint detection and response (EDR), firewalls, and spam filters have become good at preventing head-on attacks, he says, attackers have to look for openings further up the chain. And pasted-together systems provide just that kind of opening.

“Cyberattacks are easier when businesses and vendors try to 'bolt on' security after the fact,” says David Brumley, CEO of ForAllSecure. “It's like putting an after-market stereo in your car — it just doesn't work exactly right.”

To enhance software security globally, the Cybersecurity and Infrastructure Security Agency (CISA) has proposed an initiative aimed at revolutionizing development practices by embracing secure-by-design principles in the software development life cycle. It reflects a significant shift toward proactive security measures.



The [request for information](#) focuses on addressing recurrent software vulnerabilities, fortifying operational technology, and assessing the impact of secure practices on costs. The call for comment emphasizes the collective responsibility of technology manufacturers and consumers in fostering a future where technology is inherently safe and secure.

“Secure by design means security is part of how you build the software from the ground up,” Brumley explains. “That means it is much more robust from attacks.”

A Foundational Level of Security

Secure by design starts with architecture and risk management principles in operations before an organization migrates to or begins using the cloud, says Ken Dunham, cyber threat director at Qualys Threat Research Unit.

“This is a critical element of modern, complex hybrid infrastructure,” he explains. “In a world of shared responsibility, organizations must decide what risk is acceptable to be shared — and potentially at higher risk — with third parties versus that which is fully owned and managed in-house.”

The life cycle of software manufacturing is increasingly complex, Dunham notes, with many stakeholders who must be secured to reduce risk.

“Are your developers, who care about functionality and

user experiences, adept at secure coding principles, modern-day attacks, security countermeasures, and SecOps?” he asks.

Organizational security expectations put pressure on onboarding teams to properly roll out, configure, and monitor software within the business architecture.

“How mature are your incident response and cyber threat intelligence services?” Dunham says. “Do you trust them in a hybrid cloud world where you may have a complex intrusion attack at blazing speed?”

Adds Brumley: “Once you have the right people, the process is well understood. You architect the product with defense in depth, make sure your dependencies and third-party software are up to date, and use a modern technique like fuzzing to find unknown vulnerabilities.”

For Brumley, secure by default means designing in security that works with how people use the software.

“There are design principles that span multiple principles — just like when building a skyscraper, you need to think about everything from structural support to air conditioning,” he explains.

Paradigm Shift Required in IT Security

2023 was [full of examples](#) where [race conditions](#) existed for zero days — vulnerabilities were reversed and weaponized by bad actors faster than [organizations could patch them](#), Dunham notes.



“There are still some organizations struggling to patch Log4j vulnerabilities after all this time,” he says.

Organizations must identify their attack surface, internal and external, and prioritize assets and risk management accordingly if they're to get out in front when exploitation and attack risk related to a vulnerability increases, Dunham adds.

From Pace's perspective, the IT security industry must undergo a paradigm shift in how it considers risk and how to best prioritize it — and this can only happen with visibility into the supply chain. He shares an example in which a “very large organization” did not know what dependencies its security system had when it dutifully updated that system.

“After the update, it was scanned by a vulnerability scanner and it was determined that the recent [critical Apache Struts vulnerability](#) was present,” Pace says. “Now this organization has introduced a severe risk to their organization.”

Secure Design in the IoT Era

One key challenge for organizations is designing security into long-lived devices, like those part of the Internet of Things (IoT), that may not have had security as a design consideration initially, says John Gallagher, vice president of Viakoo Labs at Viakoo.

“This requires more extensive testing and may require



new engineering resources,” he says. “Likewise, building in new security features is a way to introduce new security vulnerabilities.”

Software manufacturers should embrace the use of software bills of materials (SBOMs) to find and remediate vulnerabilities more quickly, says Gallagher, noting that companies are incorporating secure-by-design practices into new products, which will ultimately be a competitive factor in the marketplace.

“In addition to [multifactor authentication] and restricted access privileges, other measures, like eliminating default passwords and providing mechanisms to more easily and quickly update firmware, are being designed into products,” he says.

Avoiding “security through obscurity” is another tenet of

secure by design, Gallagher points out. SBOMs and open source software, for example, provide security by offering transparency around the software code.

One of the areas Pace says he is most excited about as it relates to secure by default and secure by design is significantly better visibility into the software supply chain.

“Once this visibility can be achieved, we can begin to truly understand where our problems are from a foundational level and then begin to prioritize them in a way that makes sense,” he says.

About the Author: Nathan Eddy is a freelance journalist and award-winning documentary filmmaker specializing in IT security, autonomous vehicle technology, customer experience technology, and architecture and urban planning.

Newly ID'd Chinese APT Hides Backdoor in Software Updates

The threat actor went more than half a decade before being discovered — thanks to a remarkable backdoor delivered in invisible adversary-in-the-middle attacks.

By Nate Nelson, Contributing Writer, Dark Reading

Since 2018, a previously unknown Chinese threat actor has been using a novel backdoor in adversary-in-the-middle (AitM) cyber-espionage attacks against Chinese and Japanese targets.

Specific victims of [the group that ESET has named “Blackwood”](#) include a large Chinese manufacturing and trading company, the Chinese office of a Japanese engineering and manufacturing company, individuals in China and Japan, and a Chinese-speaking person connected with a high-profile research university in the UK.

That Blackwood is only being outed now, more than half a decade since its earliest known activity, can be attributed primarily to two things: its ability to effortlessly [conceal malware in updates for popular software products](#) like WPS Office and the malware itself, a highly sophisticated espionage tool called “NSPX30.”



Blackwood and NSPX30

The sophistication of NSPX30, meanwhile, can be attributed to nearly two whole decades of research and development.

According to ESET analysts, NSPX30 follows from a long lineage of backdoors dating back to what they've posthumously named “Project Wood,” seemingly first compiled back on Jan. 9, 2005.

From Project Wood — which, at various points, was used to target a Hong Kong politician, and then targets in Taiwan, Hong Kong, and southeast China — came further variants, including 2008's DCM (aka “Dark Specter”), which survived in malicious campaigns until 2018.

NSPX30, developed that same year, is the apogee of all cyber espionage that came before it.

The multistaged, multifunctional tool comprises a dropper, a DLL installer, loaders, orchestrator, and

backdoor, with the latter two coming with their own sets of additional, swappable plug-ins.

The name of the game is information theft, whether that be data about the system or network, files and directories, credentials, keystrokes, screengrabs, audio, chats, contact lists from popular messaging apps — WeChat, Telegram, Skype, Tencent QQ, etc. — and more.

Among other talents, NSPX30 can establish a reverse shell, add itself to allowlists in Chinese antivirus tools, and intercept network traffic. This latter capability allows Blackwood to effectively conceal its command-and-control infrastructure, which may have contributed to its long run without detection.

A Backdoor Hidden in Software Updates

Blackwood's greatest trick of all, though, also doubles as its greatest mystery.

To infect machines with NSPX30, it doesn't use any of the typical tricks, including phishing or infected webpages. Instead, when certain perfectly legitimate programs attempt to download updates from equally legitimate corporate servers via unencrypted HTTP, Blackwood somehow also injects its backdoor into the mix.

In other words, this isn't a SolarWinds-style supply chain breach of a vendor. Instead, ESET speculates that Blackwood may be using network implants. Such implants might be stored in vulnerable edge devices in targeted

networks, as is [common among other Chinese APTs](#).

The software products being used to spread NSPX30 include WPS Office (a popular free alternative to Microsoft's and Google's suites of office software), the QQ instant messaging service (developed by multimedia giant Tencent), and the Sogou Pinyin input method editor (China's market-leading pinyin tool with hundreds of millions of users).

So how can organizations defend against this threat? Ensure that your endpoint protection tool blocks NSPX30, and pay attention to malware detections related to legitimate software systems, advises Mathieu Tartare, senior malware researcher at ESET. "Also, properly monitor and block AitM

attacks such as ARP poisoning — modern switches have features designed to mitigate such attack," he says.

Disabling IPv6 can help thwart an IPv6 SLAAC attack, he adds.

"A well-segmented network will help as well, as the AitM will affect only the subnet where it is performed," Tartare says.

About the Author: Nate Nelson is a freelance writer based in New York City. Formerly a reporter at Threatpost, he contributes to a number of cybersecurity blogs and podcasts. He writes the tech podcast "Malicious Life" and is the co-host of "The Industrial Security Podcast."



Defending the Enterprise From Software Supply Chain Attacks

Safeguarding the software supply chain is of paramount importance. To prevent and defend against attacks, security teams must understand the risks of using open source components and remedies.

By Sharan Hiremath, Senior Product Manager, JFrog

Software supply chain attacks are on the rise. There have been many incidents during the past few years, including well-known attacks affecting [SolarWinds](#), [Log4j](#), and [Urgent/11](#).

With [80% of IT leaders](#) expected to increase their use of enterprise open source software for emerging technologies, preventing, detecting, and remediating software supply chain threats has become a top priority for organizations.

One of the benefits of open source software is the weeks of development time it can save enterprise development teams, but it also brings potential risks to the organization. Instead of just prohibiting the use of open source software outright, organizations should try to understand potential vulnerabilities in context and realistically assess the potential for damage in a runtime environment.



Supply Chain Risks

Security teams need to recognize that there are different types of risks associated with open source software components and that they need to take different steps to manage them.

Operational risks. From an operations perspective, it is important to balance the benefits of using a package in the short term against the potential damage that could be incurred in the long term. Even in situations where the effects of a particular vulnerability may be tolerable to make a critical release date, DevOps teams must consider the potential buildup of technical debt.

Code injection risks. In addition to operational risks, more immediate threats stem from the injection of malicious code — including cryptominers, password stealers, and other data exfiltrators — into what look like trusted packages.

Supply Chain Remedies

Security teams need end-to-end visibility into their software supply chains to understand the risk of a potential threat becoming an actual attack. Organizations need to manage and secure software releases while tracking software assets throughout the entire development life cycle.

Operational remedies. There are a number of precautions that security teams can take when it comes to open source packages, such as warning developers of outdated or end-of-life open source packages. One clue that open source software is not being properly supported is lack of response to queries about the package. For example, [JFrog security research](#) has found that authors of packages that are being actively supported typically respond to reported vulnerabilities with an updated version within two weeks of notification.

Code injection remedies. Security teams can identify if malicious code has been injected into software components by scanning open source packages and their dependencies, including binaries being used within the organization. Once malicious code is detected, security teams must ensure that any attempts to use infected packages are blocked during the coding and build phases of development operations.

Contextual analysis. One of the keys to optimizing development operations while maintaining high levels of security is understanding vulnerabilities in context.



Relevant CVEs should always be taken seriously, but, according to JFrog Security Research's latest findings, many vulnerabilities are not exploitable in the vast majority of operating environments. To get a realistic picture of how likely it is that a potential vulnerability will result in an actual attack, it is essential to take into account:

- Symbols used in binary executables
- Daemon configurations
- OS/distro/architecture information
- Applied vulnerability mitigations

Understanding CVEs in their proper context makes software development more efficient in the short term while providing a realistic assessment of potential damage in real-world scenarios.

Takeaways

Vulnerabilities abound, and safeguarding the software supply chain is paramount. To prevent and defend against potential attacks, the following precautions should be taken:

- **Holistic security strategies:** Take comprehensive actions to bolster security across code, configurations, and binaries.
- **Operational risk management:** Gain insights into managing operational risks such as package maintenance issues and technical debt.
- **Protection against malicious activities:** Learn how to identify and mitigate security issues that emerge after code generation.
- **Accelerated remediation techniques:** Equip yourself with the knowledge required to prioritize and address vulnerabilities by leveraging enhanced CVE data.

About the Company: JFrog is on a mission to create a world of software delivered without friction from developer to device. The JFrog Software Supply Chain Platform is a single system of record that empowers organizations to build, manage, and distribute software quickly and securely, ensuring it is available, traceable, and tamper-proof. The integrated security features also help identify, protect, and remediate against threats and vulnerabilities. Learn more at jfrog.com and follow us on X: [@jfrog](https://twitter.com/jfrog).