SolarWinds Hack and its Threat to Cloud Services

By Nishant Srivastava

An in-depth look at one of the most devastating hacks of 2020.

Introduction

Leading US info tech firm SolarWinds suffered a cyberattack that affected its clients and went undetected for months. Hackers from outside the US, believed to be from Russia, infiltrated the supply chain to spy on FireEye, an elite cybersecurity company, and representatives of the US government, including the Treasury Department and the Department of Homeland Security [1].

Supply chain attacks like the one on SolarWinds can have dire consequences for thousands of stakeholders. What’s worse is that the vulnerabilities enabling them are often overlooked. This article will communicate the importance of the supply chain, organizations’ dependencies on cloud providers that create weaknesses, and examples of failures. It will conclude with mitigation strategies and prevention approaches on par with the highest security standards.

Supply chain as a whole and its importance

A supply chain attack is a technique in which a malevolent entity slips damaging code or a damaging element into a trusted piece of hardware or software. By compromising a single supplier, the attacker can hijack its distribution systems to turn any equipment they ship, every application they sell, and/or any software update they launch into a virus. Then, they build a springboard to the supplier’s network of customers, who can number as many as tens of thousands.

“Supply chain attacks are scary because they’re really hard to deal with, and because they make it clear you’re trusting a whole ecology,” Nick Weaver, a security researcher at UC Berkeley’s International Computer Science Institute, told Wired Magazine. “You’re trusting every vendor whose code is on your machine, and you’re trusting every vendor’s vendor. [2]”

In the above malware campaign [3], intelligence officers used SolarWinds software to target their customers and users, who used SolarWinds software to view events transpiring on their computer networks. In the hack, cybercriminals inserted malicious code into the software platform Orion, which wreaked havoc during an update.

According to data [4] cited by a security researcher with GitHub, between 85% and 97% of codebases used by enterprises are derived from open source components. On average, a project has just over 200 dependences. This data was available even before the SolarWinds hack. Without a doubt, reusing code makes app development much faster and much simpler. However, it’s not without its downsides. For one, breached off-the-shelf components make innumerable organizations vulnerable to attack. The majority of organizations remain complacent about the dangers of supply chain attacks. Studies show that less than 20% of crucial national infrastructure organizations cite partners and suppliers as the biggest threat to their organization [5].

The SolarWinds attackers [6] subsequently gained access to various cloud-based solutions given that they were able to reach and decrypt API keys from Orion’s compromised databases.
Supply Chain Attack in Detail

In a filing with the SEC on December 14, 2020 [7], SolarWinds reported that hackers were able to access a system they used to make updates to Orion. Then, they inserted malware into the update. To gain access, hackers exploit vulnerabilities in target systems, such as unpatched or outdated software, or used phishing to trick people into downloading viruses, ransomware, or malware. In this supply chain attack, the cybercriminals simply bided their time until the companies and government agencies installed the Orion update upon being prompted by SolarWinds.

SolarWinds’ numerous customers are all potential targets with the release of the compromised update.

When hackers manipulate the code in a software component of a third party, the ‘downstream’ apps using it are compromised. This is a software supply chain attack. Attackers use compromised software to gain access to other parts of the network by means of ancillary movement, to corrupt the systems targeted, or to steal data.

It’s possible to target any part of an organization’s supply chain, including the open-source community, API providers, app developers, and providers of off-the-shelf software. In this type of hack, attackers compromise software development using a tool for remote access to inject a malicious element.

Examples of failures

This year, attackers breached a certificate used by cloud security firm Mimecast to verify its services on Microsoft 365 Exchange Web. Thankfully, just a few clients of the company were affected. Every tenth client used apps relying on the compromised certificate.

Security researcher Alex Birsan used a novel attack approach to breach systems [8] belonging to Tesla, Uber, Apple, and other giants. He used namespace/dependency confusion to send counterfeit (but harmless) data to a large number of well-known targets. This did not entail social engineering.

A memorable supply chain attack occurred on a GitHub repository in 2018. Hackers injected malware into an inoperative stream dependency, which was part of event-stream. To this day, nobody knows how many apps introduced compromised dependency into their code.

Security Risks of API Keys

API keys contain many things that can present a security risk in the wrong hands, which is why it’s important to be capable of identifying what’s connecting to the API server. If the key is a JSON Web Token (JWT), it might contain sensitive information. JWT defines a self-contained and compact way to transmit information as a JSON object securely between parties. RFC 7519, the open standard these tokens are based on, makes it possible to verify and trust this information since it’s digitally signed. You can sign JWTs using a private or public key pair via ECDSA or RSA or a secret one with the HMAC algorithm.

From the time you use an API key to protect access to resources, it becomes a security risk even if it doesn’t contain any sensitive data. This is because of how easy it is to access these keys from client apps and initiate an automated attack by reusing them. In these attacks, the hacker imitates the API server as being the real user and the genuine application connecting to the API server.

As the SolarWinds attack showed, some IT security professionals do not perceive the full extent of risks inherent to software supply chains. Today, ready-made components are used to build software: third-party APIs, open-source components, and proprietary code. Reusing software is customary. This is convenient without a doubt, but it is not without risk.

Other dependencies

Reliance on cloud-based providers increases the risks mentioned in the previous section. The supply chain attack on SolarWinds’ Orion endangered Microsoft Azure API keys and Amazon web services. This made it a game-changing attack on cloud-based infrastructure as well as on-premise organization systems.

Integration of “least-privilege” protocols, rotating credentials, and other counter responses is recommended. Least privilege is the practice and concept of restricting access rights for processes, accounts, and users to just those who are required to perform legitimate and routine activities. The concept of privilege itself refers to the right to bypass specific security restraints.

When applied to human users, least-privilege protocols mean enforcing minimal user rights levels or lowest clearance levels that let the user fulfill their role. Least privilege can also apply to systems, processes, devices, and applications in that each of these requires only the permissions needed to carry out an authorized activity [9].

In this case, an effective measure may involve deployment of Orion on isolated accounts, separately from everything else in or based on the cloud.

Mitigation strategies

The first and most crucial step at the technical level is to increase DevOps professionals’ security awareness. DevOps teams must dispose of a solid system to patch security bugs, be attentive to vulnerability disclosures, have a complete map of the dependencies their apps use, and take other measures to incorporate security into the entire development process. Those responsible for project management could use software bills of materials.
(SBOMs) to track components. They can keep software secure by auditing their own controls.

For example, Github gives access to a dependency graph [10] among other software composition analysis tools. The graph makes it possible to view all upstream dependencies at a glance. Dependency updates are provided by tools like Dependabot. Users get these automatically.

NIST recommends runtime application self-protection (RASP) as a way of mitigating vulnerabilities of software supply chains. Companies would also do well to consider using SAST, DAST, and other application scanning tools [11] to get early warnings of communication between command and control servers and their apps.

Security researchers point out the need to draw attention to common attack vectors like firmware keys, providing access to hundreds of thousands (and sometimes millions) of computers. In the cases of ASUS, SolarWinds, and Mimecast, cybercriminals signed a legitimate key on a malicious version of company software, resulting in the damage that made the organizations infamous.

Security professionals can create mobile apps making use of public API keys and distribute them in a safe way. This solution is called mobile app attestation [12].

Its role is to ensure that no one tampered with your mobile app at runtime, that the app is not being instrumented by Frida, xPosed, or a similar framework, and that it’s not running in a rooted device. To achieve this, you can run an SDK in the background. The app will be challenged by the cloud-based service. The service will attest to its integrity and to that of the device it’s running on based on the responses it receives. The SDK itself doesn’t make any decisions.

Any sensitive data on web apps becomes publicly accessible upon release. Anyone analyzing the app with inbuilt browser developer tools can use it.

Returning to the Orion interface – in essence, it doesn’t demonstrate all the features saved, obfuscating the attempts of the providers affected to track the extent of public accessibility. A large number of malevolent identities and processes remains connected to the cloud despite having been detected. What’s more, any cloud setting component using IAM identification can be perceived as a threat because compromised IAM identities give hackers access to sensitive resources such as Lambda, S3 buckets, KMS, and more.

Microsoft is investigating and acting on the breach inside Azure. Company representatives comment that such an innovative approach would not only target companies that leverage mainly cloud-based services, but would also have the insolence to penetrate the cloud itself. In fact, that may be their ultimate goal.

**Containing Damage**

Assuming a breach has occurred, site visitors, files, and many other elements can compromise a company’s security thereafter. Companies that want to protect their systems and users must look at credentials saved by Orion in the cloud, if any. The next step is carefully inspecting and auditing all cloud access and rotating any passwords and/or keys affected.

Service providers must improve controls on interior access and assess each resource and ID manually to determine the extent of publicity. Security professionals also need to determine the degree of access to the respective data and capabilities of the organization, which integrations and tokens in other clouds have. In the event of another breach, there is a risk that someone might use an integration account to extract data or affect a client database or another service, even a completely unrelated one.

**Ways to improve software supply chain security:**

- Create and continuously update effective software asset inventory
- Eliminate redundant or outdated systems
- Audit unapproved “shadow IT” infrastructure
- Assess suppliers’ reliability by talking to their CISOs during purchase negotiations
- Don’t treat supplier risk validation as a one-time thing
- Map and test key services
- Assess the ability to replace or restore these services if an attack occurs
- Use protection tools like RASP

**Conclusion**

In conclusion, companies need to introduce least-privilege protocols and rotating credentials and modify their software acquisition strategies. Beyond relying on vendors to provide certification and on surveys, IT departments could consider penetration testing, source code reviews, and audits. While these alternatives are expensive, they are also robust. Even a less serious vulnerability can prove damaging to an organization.

**References**

5. Ibid.
Conclusion

Despite and/or due to the current distractions of COVID and recent vulnerabilities, the lack of cybersecurity talent seems to be a top concern. Hiring managers should consider where they have the largest talent gaps and carefully craft roles and descriptions to address teams’ needs through identifying soft skills and problem-solving aptitudes. Posting help-wanted job descriptions to also include non-technical skills of critical and creative thinking with technology may be a better way to attract diverse talent. And overhauling the interview process to focus on these things rather than years of experience is also key.

As we suspected, when cybersecurity staff is stretched thin, the negative consequences get real. Cybersecurity professionals say many issues can be prevented with enough people. The glaring root causes of data breaches and attacks, misconfigured systems, slow patch cycles, rushed deployments, not enough time for proper risk assessment, not enough oversight of processes and procedures, and more, can be mitigated with people, processes, and technology [3]. In the aftermath of the Log4j vulnerability, those with short-staffed teams may suffer consequences. As we begin a new year, old habits need to change around attracting and retaining talent if we are serious about equalizing the supply versus demand.

Coming next month: “Rethink How You Hire – Part 2 of Inspiring and Preparing the Next Generation of Cybersecurity Professionals.”

References


About the Author

Dr. Curtis C Campbell, C|CISO, VP of Atlantic Capital Bank in Atlanta, GA, serves as Director on the ISSA International Board, and is President of the ISSA Chattanooga Chapter. Curtis holds a Ph.D. in Organizational Leadership in Information Systems Technology, and serves on the advisory board of University of TN-Chattanooga, a national Center for Academic Excellence for Cyber-Defense (CAE-CD) studies. She was named ISSA Fellow in 2020. Connect with Curtis via curtis@mprotechnologies.com.

When Will Supply Meet Demand?

continued from page 8

About the Author

Nishant Srivastava is a Cyber Security expert with total 10+ years of work experience in Information Technology. He is currently employed with Cognizant US firm as Cyber Security IAM Architect and worked in Deloitte in past within Cyber risk services.