The largest Data Breach in US - SolarWinds Cyber Attack

The SolarWinds attack, one of the largest cyber attacks, took the attention of the world infosec community. FireEye, which is known for its incredible capabilities in preventing APT/ cyber attacks, recognized a breach on its server on 8th Dec 2020. That time, the CEO of FireEye announced it was a sophisticated nation-state attack that highly targeted their red team assessment/penetration testing tools, and stole them. Further investigation discovered the source of the attack, which was caused by a backdoor in SolarWinds software Orion. This software was trojanized while it was updated by inserting almost 3500 lines of malicious code. Four malware strains: Sunburst, Sunspot, Raindrop, and Teardrop, were used in this attack though every other was part of Sunburst malware.

At first, in September 2019, the attackers accessed the SolarWinds network, and using Gold SAML techniques they compromised the SAML signing certificate and successfully inserted a tiny code with the SolarWinds update, which gave them the hope to launch the supply chain attack. Then in February 2020, the attackers launched a similar attack and added the Sunburst backdoor into the Orion platform update by using the Sunspot malware. Sunspot malware was used to capture the process of the msbuild.exe file, which is used to build and compile the source code automatically. The attacker compromised this process and created a temporary update file while Orion code was compiling. The very last time, when the code was last audited, the malicious code inside the temporary file told the machine to swap the developer's code into the malicious code. It was then released as the updated version of the Orion software from the SolarWinds website.

Attackers used another technique to avoid suspicion, and it was keeping off the activity of the malware for 12-14 days. It then started working by disabling the antivirus and forensic tools and sending the IP addresses, and operating system information to the threat actors using the Command-and-Control (C2) server. Attackers assigned another unique C2 server to those IP addresses where they wanted to proceed. And then they used the hands-on-keyboard attack to escalate privilege and steal information. A few months later, TearDrop and Raindrop malware were used to install the Cobalt Strike, a penetration testing tool that deploys the beacon which functions for privilege escalation, keylogging, port scanning, and many others. While TearDrop was deployed directly to the Sunburst backdoor, RainDrop was delivered to the victim's network and used power shell commands which led to the infection of other computers consequently. Finally, FireEye, Microsoft, and GoDaddy discovered a kill switch to stop the execution of this malware.

The primary focus of this attack was to target the US highly valued organizations since they have a maximum amount of the nation's leading information. White House later issued a statement and condemned the Russian Intelligence service for this master class attack though Russia denied

this. Ramakrishna, the CEO of SolarWinds, said the threat actors possibly hacked 18 thousands of customers with just one sophisticated attack. However, whoever attacked could have stolen enough sensitive data since the attack was uncovered for one year.

My recommendation

SolarWinds attack was a nation-state attack, and that's for sure not easy to detect and prevent because of its sophistication. But I would say, whatever the attack is successful at its primary stage, it could be prevented from moving forward by detecting its nature at every step though the best is to STOP at the threat level.

The initial breach into the SolarWinds network was unnoticed by the security team. They missed the intruder's unusual activity though they could prevent that by using SIEM or IDS tools, ensuring the robust security of their network, and continuously monitoring them. The attacker somehow compromised the SAML signing certificate and created fake SAML tokens to gain unauthorized access to the user accounts. To thwart unauthorized access to the SAML signing certificate, it is recommended to implement robust security that could be restricted to only authorized users. Also, rotating the certificate means generating new certificates after a timeframe by changing the cryptographic keys could prevent it as well.

The attacker then escalated within the system pretending to be a legitimate user, and discovered the vulnerable path - the supply chain. They altered the code that was prepared to release. I think if the SolarWinds security team could ensure robust security in the software building environment by using IDS/ IPS, SIEM, and using MFA authentication method, they could ensure the code integrity.

Trust is the word that we love so much. But there’s no reason to accept this word when it’s associated with sensitive matters. Though in the SolarWinds attack, the attacker used so many attacking methods, the attack spread by breaking the trust, which was one of the weakest parts of the security. Every company that updated the software wasn’t serious about that. Before deploying the software, they could assess the security including scanning the updates, source code reviews, and binary analysis to mitigate the risk of the third-party software.

SolarWinds made a mistake by publishing their buyer's list which could have attracted the attention of the attacker since so many US high-valued organizations' names were listed there. The security team should have asked them before to remove it for the security purpose.

Sadly, the DHS / US National Security Agency couldn't catch this year-long attack. Christopher Krebs, who was in charge of the DHS, said DHS had a threat detection system Einstein which could only detect known threats, and they only depended on this tool since 90 to 95 percent of threats were known in previous attacks. But this attack used unidentified techniques and Einstein didn't scan the software updates. I will say that before proceeding with every step, they should analyze the steps in detail.

The first sign of the malicious activity was identified in Volexity, a cybersecurity company. However their investigation team couldn't gather much information to report that issue to the US government or SolarWinds, so they ignored it. But they could at least share the doubt with SolarWinds or collaborate with the cybersecurity community.

I believe it was a highly sophisticated attack since attackers left minimal indicators of compromise that made it harder to identify. Often, attackers used legitimate tools like cobalt strike, command-control servers, power shells, communicated using encryption, and didn't interfere with the high-profile account activity. So, they remained unsuspicious for longer. But I think a more comprehensive analysis could potentially prevent or discover this attack early.

Most vulnerable device

Since the task is to discover the device which is easiest to hack, I would say, I have the best option to choose those devices which have the uses of the Android Operating System. In this case, why not say smartphone? First of all, it's highly challenging to hack the device because of the lack of ability to control the security without the operating system. For example, a smartphone has input/output ports but it won't respond to any commands without the software function. Therefore, the remaining discussion will be on using the operating system.

An operating system has the dedicated functionality to operate and secure a device. Android OS also has released so many versions while updating the features and security though according to the CVE statistics, the most, 6900 vulnerabilities are discovered in Android OS versions. So, I think adding the Android OS to the smartphone could make it easier to hack.

To make it most vulnerable, I would detach some significant security features from that OS. First of all, I would cut the network security features like VPN, TLS/SSL, WPA3/ WPA2 so that an attacker can easily do ARP poisoning, MAC flooding, DDoS attacks, and many more. Then I would remove the full disk encryption protocol which secures the stored data on the device even if the device is compromised. Then the sandbox feature isolates applications from each other. Again, without an antivirus, a device can't detect and prevent malware. Using outdated systems could also cause a potential threat. Sometimes it's easier to hack a device using a malicious cable which could cause the device to respond to the command, controlled by another device.

Another way to make this device vulnerable is by enabling Bluetooth and the Near-Field-Communication feature. The intruder may use this feature for bluebugging which is taking control of the device for making calls and sending/ reading messages. Using default / weak configuration settings helps the attacker exploit the system easily. Also, I would add an embedded device which has the least security feature. By exploiting the embedded device an attacker can gain access to my invented smartphone.

It doesn't need to say how much simpler it is to hack a device if it uses a weak password. Devices that don't use the application locks have a great risk of stealing sensitive credentials and using those not only hacking but also the PII (Personal Identifiable Information)/ sensitive information.

Finally, I want to address my speculation: while I hypothetically invented a vulnerable device, my aspiration remains resolute to secure the world from any kind of data breach. Everyone has data including me, my relatives, my country as well as the entire world. So, no more data breach. Never! If God wishes I would improve the security more and more in the coming days.