James Shannon

SolarWinds Inc. Data Breach

In my opinion and based on the attached readings, the root cause of the problem that caused the security breach of several SolarWinds Inc. clients and the exposure of sensitive information is the backdoor access that hackers obtained to modify code before it was sent out via patches. SolarWinds should have verified the code was correct before pushing the patch and detected the malicious code before the code affected their customers.

Secured Software Development Framework (SSDF) is “a subset of high-level practices based on established standards, guidance, and secure software development practice documents.” SSDF can help SolarWinds Inc to “reduce the number of vulnerabilities in released software, to mitigate the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and to address the root causes of vulnerabilities to prevent future recurrences” per the NIST Cybersecurity White Paper.

Two SSDF practices that SolarWinds Inc. can benefit from include protecting all forms of code from unauthorized access and tampering and providing a mechanism for verifying software release integrity. This will help to prevent changes in code like what happened in the breach and establish integrity with each patch to verify the source code has not been modified inappropriately to maintain safety and integrity of the systems.

The specific tasks that SolarWinds Inc. can perform to support these practices would be to “store all forms of code, including source code and executable code, based on the principle of least privilege so that only authorized personnel have the necessary forms of access.” and to “make verification information available to software consumers” per the NIST Cybersecurity White Paper.

A specific implementation example to support the practice of protecting code from unauthorized access and tampering would be to use a repository with restricted access, a changelog, having someone verify all changes before pushing code, the usage of code signing and cryptography to protect the integrity of data, and to “create and maintain a software bill of materials (SBOM) for each software package created” per the NIST Cybersecurity White Paper.

A specific implementation example to support the practice of providing a mechanism for verifying software release integrity would be to post the cryptographic hashes for each release file on a well-secured website, the aforementioned code signing, and auditory oversight of the code signing process.

Sources:
https://doi.org/10.6028/NIST.CSWP.04232020