ShiftLeft

Illuminate: Identify Insider Attacks in Your Source Code

Can a determined attacker plan an insider attack in your source code?

Insider Attack in Source Code

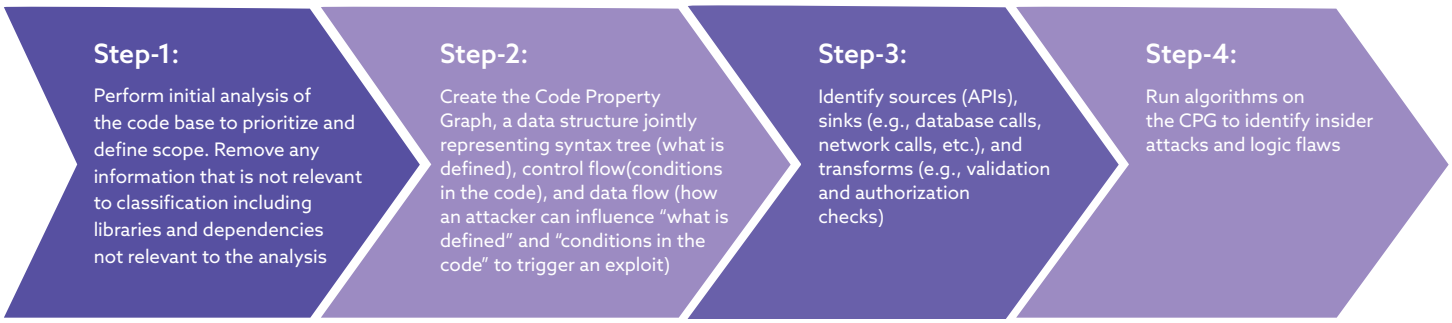
Software supply chain compromise is trending after the public disclosure of the SUNBURST backdoor in early December of 2020. The attack trojanized SolarWinds Orion business software updates. SolarWinds has 300,000 customers which includes every branch of the US military and four-fifths of the Fortune 500. It is believed that external actors infiltrated the source code management system of SolarWinds and added malicious code evading all the checks and balances in the software development process. More details on the attack can be found here.

Organizations that write source code are figuring out how to avoid similar insider attacks. The Solarwinds attack shows the importance of analyzing source code for insider attacks and backdoors in addition to OWASP vulnerabilities.

Why ShiftLeft

Detecting insider attacks and backdoors in source code is non-trivial. In the case of SUNBURST, it is estimated to have happened over multiple months with small incremental changes. ShiftLeft's code analysis solution utilizes complete data flow and advanced control flow analysis to uncover such complex backdoors.

Service Overview



Some potential indicators in algorithms used for analysis:

- **Markers** - Use of dynamic invocation, suspicious control flows, code in dormant state based on conditional scheduling, and obfuscated literals
- **Behavior** - Usage of concealed system commands to gain knowledge of the target (e.g., ping, netstat, etc.)
- **Errors** - sinkhole exceptions for application failures to evade detection

ShiftLeft algorithms use complex techniques to:

- Overcome attacker's obfuscation measures
- Uncover suspicious commands/functions
- Detect and track complex control flows used by attackers to hide their intention

Benefits

- Definitively know whether an insider attack has occurred in your source code. Receive remediation steps
- Identify the potentially exploitable areas for "insider attack". Receive recommendations on reducing future risk
- What to look for and where on an ongoing basis depending on the unique architecture of your application (e.g., coding practices, library usage, etc.)

What you get

- Summary report for your executive and senior-level management, identifying any insider attacks, remediation advice, and overall risk of the code base.
- Technical report of insider attacks found and remediation advice
- Set of rules for ShiftLeft NG-SAST for all findings so that these rules are run with every code analysis (if you choose to deploy ShiftLeft code analysis, this process will also be started)
- Strategic recommendations for longer-term improvement