Secure by Design

The SolarWinds Approach to Secure Software Development

Recognizing the additional security requirements being levied on our federal customers, SolarWinds has formed a cross-functional team of experts to conduct a gap analysis of our Next-Generation Build System alongside the minimum guidelines our federal customers are directed to follow. We found that our build processes and procedures, with few exceptions, meet or exceed the minimum guidelines outlined in the *Executive Order on Improving the Nation's Cybersecurity (EO 14028)* and the National Institute of Standards and Technology (NIST) *Secure Software Development Framework Version 1.1 (SSDF)*. For those few exceptions, we have a similar protocol or control that we believe satisfies the requirement. The table below provides a detailed mapping of SolarWinds Secure by Design practices and procedures to EO 14028 subsections and SSDF practices and tasks outlined in Appendix A of the SSDF.

BACKGROUND

In December 2020, we experienced one of the most sophisticated and deliberate cyberattacks in history. The SUNBURST breach into the SolarWinds software build environment rattled the digital ecosystem and revealed the increasingly sophisticated evolution of malicious actors' intents and capabilities.

Post-SUNBURST, we set out to revolutionize the software development industry with our Next-Generation Build System, designed to provide the most powerful, affordable, and secure observability and IT operations management solutions available. Instead of merely adopting zero-trust principles, our Secure by Design effort took it further by embracing an assume breach (user or asset) mindset to eliminate implicit trust in applications and services and assume users are most likely already compromised regardless of authentication practices. Operating with this assumption not only reshapes detection and response strategies but also aligns with the National Cybersecurity Strategy to bolster the security and integrity of the software supply chain.

In May 2021, the President of the United States issued EO 14028 aimed to modernize and implement stronger cybersecurity standards in the Federal Government and improve the software supply chain in response to SUNBURST and other high-profile cyberattacks. Executive Order 14028 directed NIST to develop, update, and implement zero-trust architecture and framework guidance to enhance the security of the software supply chain. The EO also directed the Office of Management and Budget (OMB) to require agencies to comply with the NIST and other guidance derived from EO 14028.

Over the rest of 2021 and into 2022, federal agencies began to implement EO 14028 by releasing an updated NIST Secure Software Development Framework Version 1.1 (SSDF) and the *Enduring Security Framework (ESF)* for Developers from the Cybersecurity and Infrastructure Security Agency (CISA), National Security Agency (NSA), and Office of the Director of National Intelligence (ODNI) that guides best practices and recommendations to increase the resiliency of the software supply chain.

Subsequently, OMB published a memorandum to federal departments and agencies titled *Enhancing the Security of the Software Supply Chain through Secure Software Development Practices (M-22-18)*. The OMB memorandum identified the SSDF and the NIST Software Supply Chain Security Guidance as the "NIST Guidance." It directed that federal agencies must only use software provided by software suppliers that incorporate the minimum elements of the SSDF. By conducting our gap analysis and implementing Secure by Design protocols and controls, SolarWinds is confident that we can meet the needs of U.S. federal customers.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Using administratively separate build environments;	4e(i)A	PO.5.1	The **SolarWinds Secure by Design Next-Generation Build System** utilizes three isolated and distinct build environments – standard, validation, and security. The standard environment executes build definitions resident in project repositories on **GitHub** and performs a variety of activities to validate them for correctness and security. All build steps are automated with access controls and peer reviews, and artifacts generated from build steps are cryptographically signed and verified in subsequent build steps. The steps are logged and sent outside the environment for secure immutable storage for an extended period. All build jobs are then handed to our highly secure validation environment, which has zero ingress or egress to the internet. The standard environment build jobs are transmitted over a message bus to the validation environment. There is no way to interact with the system other than through highly secured and audited channels available only to essential DevSecOps personnel. The purpose of this environment is to perform the exact same build process, producing attested build steps, just as produced in the standard environment. Our production environment acts as the third layer, performing a variety of scans and security checks to validate the product before release.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Auditing trust relationships;	4e(i)B	PO.5.1, PO.5.2	Each build environment has very limited access and no single person has access to all three build environments. Internal audits are conducted regularly to inspect the build environments, as well as internal and external red-team penetration testing. Early in our design efforts, we crystallized what we call the "Golden Rule" of our Next-Generation Build System: > *"Developers shall have* **fine-grained control** *over the things they build—but have* **zero control** *over how those things are validated and secured."* One of our key Secure by Design requirements is to conduct our builds in parallel, producing multiple, highly secure duplicates of our new build system and building all artifacts in parallel, across all systems, at once, to establish a basis for integrity checks. We refer to the products of such a system as consensus-attested builds. All code in consensus-attested builds is peer-reviewed prior to commit. The validation and security environments are compared prior to release to ensure they are identical and authentic.
Establishing multifactor, risk-based authentication and conditional access across the enterprise;	4e(i)C	PO.5.1, PO.5.2	We've taken zero trust a step further by adopting and implementing an assume breach position. Assume breach means we start with assuming something has been breached (a user or asset), look at the possible result, and determine how to limit the exposure. By using the guiding principles of Secure by Design, we aim to eliminate implicit trust in applications and services and assume users aren't secure and are most likely already compromised regardless of authentication practices. This approach securely connects the right user to the right data at the right time under the right conditions.
Documenting and minimizing dependencies on enterprise products that are part of the environments used to develop, build, and edit software;	4e(i)D	PO.5.1	Our Next-Generation Build System is based on ephemeral operations to eliminate dependences and remove long-lived environments available for attackers to compromise. The build environment is reconstructed every time it is run, through a process that spins up resources on-demand and destroys them when they complete each discrete task, removing the opportunity for attackers to establish a "home base" in our systems, and making it even harder for threat actors to attempt and attack. All artifacts are signed and verified in the subsequent build steps. If any of the artifact signatures cannot be verified, the build steps fail and result in complete build failure. Build steps are not editable by developers with access control, all changes to build steps are peer-reviewed and approved, and all step executions are logged and sent to immutable storage for record-keeping.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Employing encryption for data;	4e(i)E	PO.5.2	The endpoints and resources for all three build environments are configured to enforce the principle of least privilege for each DevSecOps user role in their assigned production environment. Roles are implemented allowing only certain personnel access to code repositories and build environments. Every user utilizes a two-factor **YubiKey** authentication device for an additional layer of security to secure identities and roles to access their specific production environment and resources. Encrypted traffic is required between the build processes products implement industry standard encryption algorithms. All data in the build environment is encrypted at rest, including build agents, artifact stores, and storage services, and all communication in and out is encrypted in transit.
Monitoring operations and alerts and responding to attempted and actual cyber incidents;	4e(i)F	PO.3.2, PO.3.3, PO.5.1, PO.5.2	All SolarWinds build environments are automated and require no human interaction to complete the build outside of the manual command to invoke the build script. We use the security scanning tools and methodology recommended by NIST for developer verification of software (NISTIR 8397) for static, dynamic, and software composition analysis and ensure high severity issues are addressed. The cloud infrastructure elements used for the build are under continuous monitoring by our multiple internal and external security operations centers (SOCs) to analyze and detect any policy violations and anomalies. Risk acceptance is time-bound depending on the severity and criticality of the violation or anomaly detected.
Generating and, when requested by a purchaser, providing artifacts that demonstrate conformance to the processes set forth in subsection (e)(i) of this section;	4e(ii)	PO.3.2, PO.3.3, PO.5.1 PO.5.2	Artifacts are stored in an access-controlled repository that can only be fetched over secure channels. Cached artifacts stored in the repository are purely content-addressable to prevent tampering in subsequent usage. Prior to downloading the artifacts, the package manager verifies the integrity of each artifact through vulnerability scans before introducing them to the build environment
Employing automated tools, or comparable processes, to maintain trusted source code supply chains, thereby ensuring the integrity of the code;	4e(iii)	PO.3.1, PO.3.2, PO.5.1, PO.5.2, PS.1.1, PS.2.1, PS.3.1, PW.4.1, PW.4.4	By building in parallel, we can produce consensus-attested builds in separate environments using automated tools to strip the timestamps and other non-deterministic data, resulting in a fully reproducible file. This means the files we use to build our products have been built deterministically for most artifacts. However, we deliver **Microsoft Installers (MSI)** to customers, which cannot be built deterministically. We've developed tooling for cases like this, in which we record the digests of the top-level artifact (the MSI, in this example) as well as the reproducible digest available from the componentry. This gives us a basis for consensus attestation in our system to prove supply chain integrity.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Employing automated tools, or comparable processes, that check for known and potential vulnerabilities and remediate them, which shall operate regularly, or at a minimum prior to product, version, or update release;	4e(iv)	PO.4.1, PO.4.2, PS.1.1, PW.2.1, PW.4.4, PW.5.1, PW.6.1, PW.6.2, PW.7.1, PW.7.2, PW.8.2, PW.9.1, PW.9.2, RV.1.1, RV.1.2, RV.2.1, RV.2.2, RV.3.3	There are multiple established processes, approvals, and reviews in place for our products. Peer reviews are required for code check-in and prior to committing new code. New product features require an architecture design review by select personnel on the security team. The independent security team reviews security-specific features and the design of the overall build system, the identity and access management controls, and the security requirements related to access to the environment. There are also multiple automated tools that run on a recurring basis to scan for vulnerabilities throughout the development process: • Open-source software vulnerability checks are run on a reoccurring basis to identify vulnerabilities in libraries and potential license issues. • Static code analysis tools also run on a reoccurring basis to identify vulnerabilities in the source code and helps to prioritize and quickly remediate security issues. • Complementary to these tools and prior to release, software composition analysis and binary software inspection tools are run to generate a software bill of materials (SBOM) that provides a comprehensive picture of the transitive dependency tree and historical basis for auditing and forensics purposes. This also verifies the artifacts embedded in our software checks and performs an inspection of the executable. • Dynamic application security testing tools are run to further reduce the attack aperture and risk of the build prior to shipping. Our products are shipped in a default configuration that is secure in nature and the customer is guided to a secure configuration guide to install the product(s) appropriately. Vulnerabilities identified internally, externally, or through tools are entered into **JIRA**, given a medium, high, or critical tag of security, and prioritized for remediation by the Common Vulnerability Scoring System (CVSS) score. The security team oversees and monitors the development team to ensure appropriate action is taken to remediate the vulnerabilities in a timely manner. Critical vulnerabilities are given priority and fixed in special releases. A Risk Acceptance Form (RAF) model is implemented to highlight when appropriate timelines are not met for remediation.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Providing, when requested by a purchaser, artifacts of the execution of the tools and processes described in subsection (e)(iii) and (iv) of this section, and making publicly available summary information on completion of these actions, to include a summary description of the risks assessed and mitigated;	4(v)	PO.3.2, PO.3.3, PO.4.1, PO.4.2, PO.5.1, PO.5.2, PW.1.2, PW.2.1, PW.7.2, PW.8.2, RV.2.2	A complete third-party risk model was performed on our Orion® Platform in 2021. This has been a guiding factor in the improvements over the last few years. Software security requirements, vulnerability prioritization, and reviews are routinely performed with the engineering and security team. The product management team continuously evaluates additional product security features to enhance resilience and harden the build environment. Our build implementation checks include generation of builds from multiple environments, continuous verification of the artifacts embedded into our software checks, and a thorough review of the results prior to approval of releases.
Maintaining accurate and up-to-date data, provenance (i.e., origin) of software code or components, and controls on internal and third-party software components, tools, and services present in software development processes, and performing audits and enforcement of these controls on a recurring basis;	4e(vi)	PO.1.3, PO.3.2, PO.5.1, PO.5.2, PS.3.1, PS.3.2, PW.4.1, PW.4.4, RV.1.1, RV.1.2	Aside from some of our previously discussed internal and external controls implemented throughout the software development lifecycle, third-party risk management is an ongoing evaluation process at SolarWinds. Open-source software vulnerability reviews and vendor verifications are continuously reviewed and scanned with an open-source vulnerability scanning tool to ensure only secure third-party components are integrated into our build environments. Historical software releases are internally maintained with accurate and up-to-date data and source code with versioning maintained in GitHub for auditing and forensic purposes. Our open-source approval process is designed to ensure that any open-source product we use is reviewed for the source validation, license validation, and is maintained in our artifact repository prior to using in the builds. Build systems are blocked from pulling artifacts from the open source and internet without prior approval and reviews.
Providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;	4e(vii)	PS.3.2	We can provide an SBOM for most products, however we are awaiting further guidance from the U.S. federal government on the standardized content and format of SBOMs before publishing them to our public-facing website. We plan to make our SBOMs available to customers by the end of 2023. Our SBOM files are generated by open-source vulnerability scanning, software composition analysis, and binary software inspection tools at build time and are used in the build process in three ways: 1. To validate third-party dependencies haven't changed underlying code while still retaining previous version numbers. This helps to mitigate a third-party library code compromise. 2. To provide a comprehensive picture of the transitive dependency tree available on a current build and historical basis for auditing and forensics purposes. 3. To power a process wherein we use a universal software composition analysis solution to perform build-time checks and enforce policies based on CVSS scoring. We only attest to the security of a dependency tree if no dependencies include vulnerabilities with high CVSS scores.

EXECUTIVE ORDER 14028 CONTROL AND PARAGRAPH		SSDF PRACTICES AND TASKS	SOLARWINDS SECURE BY DESIGN
Participating in a vulnerability disclosure program that includes a reporting and disclosure process;	4e(viii)	RV.1.1, RV.1.2, RV.1.3, RV.2.1, RV.2.2, RV.3.3	Our security team monitors the National Vulnerability Database and subscribes to the CISA Cybersecurity Alerts and Advisories for vulnerability reports. They utilize VEX documents to understand if a particular software component is affected or not by gaining a complete picture of risk in a specific context with our SBOMs. The toolchain is configured to perform automated code analysis and testing on a continuous basis for supported releases. The SolarWinds Product Security Incident Response Team (PSIRT) has defined processes and procedures in place for fixing verified and validated vulnerabilities reported to us for products covered under the *.solarwinds.com domain and products available here: www.solarwinds.com/downloads. Vulnerabilities and/or other sensitive security information should be sent to the PSIRT through encrypted communications or utilizing the SolarWinds PSIRT PGP Public Key. Our PSIRT communicates these reports/incidents to the CISO and security team directly to expedite any mitigations that may be necessary. Additional training is provided to our DevSecOps personnel to minimize the potential for recurrence in future builds. SolarWinds is also a CVE Numbering Authority (CNA) and issues Common Vulnerabilities and Exposures (CVEs) for our products.
Attesting to conformity with secure software development practices; and	4e(ix)	All practices and tasks consistent with a risk-based approach	SolarWinds is awaiting CISA's publication of the final common self-attestation form for federal departments and agencies to collect.
Ensuring and attesting, to the extent practicable, to the integrity and provenance of open-source software used within any portion of a product.	4e(x)	PS.2.1, PS.3.1, PS.3.2, PW.4.1, PW.4.4	Historical software releases are internally maintained with accurate and up-to-date data and source code with versioning maintained in GitHub for auditing and forensic purposes. All releases are signed with a SolarWinds certificate and hashes are published for the components that make up the build. We have a defined process and legally approved open-source software (OSS) policy in place that requires the OSS review board to evaluate requests to utilize new libraries. If a previously approved and utilized library will not meet the requirement(s), a systematic review and vendor verification is conducted and scanned with an open-source vulnerability scanning tool to ensure only secure third-party components are integrated into our build environments.