ActiveState

How US Government ISVs Can Quickly Verify CISA Attestation

Executive Summary

Independent Software Vendors (ISVs) that create and sell software solutions to the US government will need to prove they are aligning with the National Institute of Standards and Technology (NIST) Secure Software Development Framework (SSDF) by signing the Cybersecurity and Infrastructure Security Agency (CISA) Attestation.

Failure to comply can mean the loss of lucrative US government contracts.

The requirements to meet the NIST SSDF specification are both broad and deep. Verifying the CISA Attestation is a necessary first step, indicating progress toward implementing SSDF by attesting to meeting a subset of requirements, which include:

- **Development Environment Security:** implement controls to ensure code is being developed, checked in/out, and built in a manner that minimizes risk.
- **Software Supply Chain Security:** implement controls to ensure the security and integrity of open source and other third-party software.
- **Code and Artifact Provenance:** create and maintain provenance in order to validate that software artifacts have been sourced and built securely.
- **Vulnerability Remediation:** identify, disclose and remediate vulnerabilities in a timely manner depending on risk level.

Complying with these requirements can be both time and resource intensive, diverting attention from creating innovative software to implementing processes, procedures and tooling.

Solutions like those from ActiveState can help bridge the gap between current capabilities and NIST/CISA requirements by providing a single platform that can be integrated with most software development processes in a matter of days to address many of the key requirements.

Introduction

Faced with a sharp increase in sophisticated cyberattacks on the US public sector and critical infrastructure, President Biden issued Executive Order 14028 (EO 14028) in May 2021 imploring ISVs to "make bold changes and significant investments" that prioritize software security. The US government has chosen to lead by example, and since US agencies and departments deploy software built by private contractors, those contractors have become the tip of the spear for creating "secure by design/ secure by default" software.

This initiative by the US government (as well as others, such as the UK) is in response to the fact that most ISVs tend to tack on security at the end of their Software Development Lifecycle (SDLC) rather than making the implementation of security a key goal throughout the process. Security teams end up overburdened by a tremendous amount of work that needs to be resolved in a very short amount of time, bound by a pressing release date.

The result is all too predictable:

- 81%[1] of organizations ship applications with known vulnerabilities.
- 91%[2] of organizations experienced a software supply chain attack in 2023.
- Cyber attacks resulted in $46B[3] in damages in 2023.

Given that the onus for securing software rests with the creators, ISVs have been tasked with becoming the first line of defense for securing the software the US government relies on. This has resulted in two primary requirements for US government ISVs:

- Ultimately, adopting secure software development best practices defined by the National Institute of Standards and Technology's (NIST) Secure Software Development Framework (SSDF).
- Verifying progress towards adoption of SSDF by submitting the Cybersecurity and Infrastructure Security Agency (CISA) Attestation form.

1. https://github.blog/2024-02-06-appsec-is-harder-than-you-think-heres-how-ai-can-help/
2. https://www.securitymagazine.com/articles/100402-91-of-organizations-faced-a-software-supply-chain-attack-last-year
3. https://world.einnews.com/pr_news/659375862/software-supply-chain-attacks-to-cost-the-world-60-billion-by-2025

The US government understands that adoption of security best practices by development teams that have traditionally prioritized creativity and speed over security takes time. But then, so does the creation of guidelines that can help break with tradition.

For example, it's taken almost four years to arrive at the guidelines for CISA Attestations:

- **Dec 2020 - Solarwinds Hack**
  - » Solarwinds' Orion software build process was compromised prior to the signing stage. Customers like CISA, FBI, NSA, and the US armed services were affected.
- **May 2021 - Colonial Pipeline Attack**
  - » The computer systems controlling distribution of oil and jet fuel from Huston to the East coast suffered a ransomware cyberattack, shutting it down.
- **May 2021 - EO 14028**
  - » Directed NIST and the Office of Management and Budget (OMB) to define cybersecurity best practices and requirements.
- **Sept 2021 - OMB Memorandum**
  - » Proposed an attestation that would be verified by US government ISVs to show they were on the path to adopting secure software development practices.
- **Feb 2022 - NIST SSDF V1.1**
  - » Provides mappings from EO 14028 to SSDF practices to help ISVs comply with key security requirements.
- **Feb 2024 - NIST Cybersecurity Framework 2.0**
  - » Provides actionable guidance to help organizations reduce cybersecurity risks around six key functions: Identify, Protect, Detect, Respond, Recover & Govern.
- **March 2024 - CISA Secure Software Development Attestation Form**
  - » Allows US government ISVs to attest that they are pursuing the implementation of the NIST SSDF.

While CISA has yet to confirm an Attestation Form submission deadline, the OMB has suggested a date of no later than six months following OMB Paperwork Reduction Act approval, which occurred on March 8, 2024.

CISA Attestations For ISVs

The SSDF specification requires that ISVs undertake what may be a multi-year implementation, impacting multiple stakeholders across the enterprise, as well as involving quite a few new and/or changes to existing processes and controls in order to become compliant.

The good news is that the CISA Attestation requires only a subset of SSDF requirements to be implemented before an ISV can attest “in good faith” that they are aligned with the US government’s requirements. The bad news is that implementing the subset is a non-trivial undertaking for organizations that aren’t already prioritizing security.

In brief, the CISA Attestation requirements include:

- **Development Environment Security:** developer desktops, code repositories, and CI/CD systems must be implemented with secure controls to ensure code is being developed, checked in/out, and built in a manner that minimizes risk.
- **Software Supply Chain Security:** implement controls to ensure the security and integrity of open source and other third-party software.
- **Code and Artifact Provenance:** create and maintain provenance in order to validate that software artifacts have been sourced and built securely.
- **Vulnerability Remediation:** identify, disclose and remediate vulnerabilities in a timely manner depending on risk level.

In some cases, there are a number of best practices organizations may already be employing that can help attest to compliance with these requirements. In other cases, closing the gaps may need significantly more work to meet the requirement set by CISA.

Meeting Environment Requirements

While the CISA requirements may seem like “SSDF lite” they do spell out some stringent goals that need to be met. For example, aligning with the first CISA requirement means verifying that software is being developed and built in secure environments. On a practical level, this means adhering to standard best practices like:

- Logging, monitoring & auditing authorization and authentication checkpoints.
- Enforcing Multi-Factor Authentication (MFA), especially when it comes to code repositories, cloud infrastructure, local servers, etc.
- Preventing leakage of secrets, credentials and other sensitive information, especially in GitHub repositories.

But it also means minimizing the risk of attack vectors being exploited in each environment, such as the hack suffered by Solarwinds in their Orion software build environment, which was not secured against the injection of compromised dependencies into their build pipeline. Some things to consider:

- **Developer Environments** are typically built from a set of “golden dependencies” that have been approved for use by compliance, security and IT teams, and then placed in an artifact repository within the organization. The problem with this approach is the static nature of these repositories, which forces organizations into a reactive mode. For example, when a dependency is found to be vulnerable, or a new dependency is required as part of the development effort it can mean weeks of waiting for teams to sign off, impacting release dates.
  - » Consider creating a proactive process instead based on a more dynamic service that can identify and recommend newer versions of dependencies as and when they become available.
- **Build Environments** are all too often imperative (i.e., script-based), and rarely feature the kinds of check and balances designed to programmatically ensure (for example) build steps and payloads haven’t become compromised at some point during the CI/CD pipeline.
  - » Instead, consider implementing a declarative build environment that breaks down each stage of the pipeline into multiple discrete steps, each of which can incorporate a verification against a hash or an SBOM (for example, to ensure extra dependencies haven’t been injected) or a Provenance Attestation (for example, to ensure dependencies meet sourcing requirements), etc. The goal should be a hardened, tamper-proof build environment.

Implementing Software Supply Chain Security

The second CISA requirement advocates maintaining trusted open source supply chains (as well as other third-party code) by employing automated tools.

Most organizations today rely on binary scans of prebuilt open source components that form the vast majority of their software supply chain. However, binary scans can be problematic for a number of reasons, not least of which is the fact that no two scanners will return the exact same set of open source components.

But the software supply chain extends beyond just the import of open source software to include both the build process and consumption of built artifacts. As described above, a hardened, tamper-proof build service capable of creating reproducible, signed builds is key to ensuring the security and integrity of software artifacts for use in software development. Alternatively, consider sourcing built artifacts from a trusted third party that can provide securely built artifacts.

For example, the ActiveState Platform automatically builds open source dependencies from vetted source code using a hardened build service, eliminating attack vectors like typosquatting, dependency confusion, malware, etc, while providing a complete catalog of open source components based on the build dependency tree, etc., including linked OS-specific binaries – and even includes build components, as well.

Provenance

CISA requirement number three involves the creation and maintenance of provenance for third-party components. Provenance reflects the trustworthiness of the components used in the software creation process: were they acquired from a legitimate source, built in a secure manner, and not tampered with prior to use?

Software attestations are a key way for ISVs to establish trust for their software by offering customers a way to independently validate the security and integrity of their applications. Software attestations include:

- **Provenance Attestations,** which provide metadata about where the component was sourced from, and how it was built.
- **Verifiable Summary Attestations,** which verify the security and integrity of built artifacts as measured against Secure Levels for Software Artifacts (SLSA) Build Levels.

Unfortunately, open source repositories do not currently provide software attestations for their prebuilt packages. Consider obtaining open source packages from a third party that can provide them. The alternative is building open source packages from source code using a service/tool that can generate software attestation.

Automating Vulnerability Remediation

The last CISA requirement advocates the implementation of automated tools that can identify, disclose and address security vulnerabilities in a timely fashion.

Open source vulnerabilities have been a longstanding issue for ISVs, but government regulations can be more demanding than many have been used to. For example, the US government now tracks Known Exploited Vulnerabilities (KEVs), which are vulnerabilities that are known to have already been exploited in the wild (as opposed to theoretically exploitable vulnerabilities). Some critical KEVs may have a remediation SLA of just 4 days, which can be all but impossible to comply with without automation tooling in place.

While most ISVs will already possess tools that help reduce Mean Time To Identification (MTTI) by providing automated notifications when a vulnerability is identified, decreasing Mean Time To Remediation (MTTR) requires tools capable of:

- Automatically identifying reachability, which is an estimate of whether the vulnerable method is actually being called by the software.
- Automatically identifying whether a patch or new version of the affected component is available.
  - » Alternatively automatically generating code snippets that can be used to eliminate the vulnerability.
- Automatically rebuilding the runtime environment, ready for testing.

These capabilities generally require the implementation of multiple tools, but can dramatically decrease or even eliminate the lengthy investigate, patch, test, rebuild, and redeploy cycle.

Conclusions

Rather than cobbling together multiple point solutions, consider implementing a single platform that can help meet the key CISA attestation requirements.

The ActiveState Platform is a SaaS solution designed to be integrated with any software development process in days, not weeks. This can allow you to meet CISA Attestation requirements more quickly and easily than implementing them yourself, including:

- **Discoverability -** Secure development environments built from a vetted catalog of continually updated open source components.
- **Security -** Secure, tamper-proof build service that employs ephemeral, hermetically sealed build steps in conjunction with hash validation in order to ensure reproducible builds that also generate software attestations for each built artifact.
- **Observability -** A complete software supply chain catalog of open source components tracked over time by project and environment in order to provide warnings and notifications when vulnerabilities occur.
- **Remediability -** A continuously updated catalog ensures fixed versions can be leveraged in a timely manner and automatically rebuilt into a non-vulnerable runtime environment in minutes.

In other words, the ActiveState Platform can help quickly bridge the gap between current capabilities and CISA requirements by providing an easy-to-implement solution that addresses many of the most essential features.

ActiveState

Tame open source complexities

www.activestate.com
solutions@activestate.com