BLACK DUCK®

GUIDE

WHAT YOU NEED TO KNOW ABOUT THE NIST SSDF GUIDELINES

WHAT IS THE NIST SSDF?

In May 2021, President Biden signed Executive Order 14028, "Improving the Nation's Cybersecurity" (EO 14028). This EO requires software suppliers selling directly or indirectly to the U.S. government to conform to Secure Software Development Framework (SSDF) standards as defined by the National Institute for Standards and Technology (NIST). As of September 2024, organizations are required to self-attest to these standards using GSA 7700: Secure Software Development Attestation.

While the EO is a U.S.-specific order, it nonetheless provides a useful framework for all organizations to improve their security posture. The EO specifies that software developed after Sept. 14, 2022, including software delivered via continuous update or continuous delivery as part of an SaaS solution, and software sold directly to the U.S. government, must provide attestations of conformance to NIST SSDF using GSA 7700. Similar legislation is in the process of being implemented in EMEA, Japan, and other regions.

But even if you do not sell directly to the government, the guidelines and best practices set out by the SSDF are designed to enhance the security and integrity of software development processes. We often say that "every business is a software business." Whether you're selling software directly to your customers, developing it to run your operations, or both, the security of your software forms the basis of trust that your customers rely on. Therefore, it is vital that your code meets these standards.

ARE YOU READY?

Many organizations began preparing for this requirement when EO 14028 was announced in the spring of 2021, but as with all things regulatory, organizations meet these security milestones at different points in their own development. In fact, many organizations that are not bound by this regulatory obligation are implementing the NIST SSDF guidelines as a means of improving their security posture.

Now that these deadlines are upon us, how should you get started? First, address the following questions:

- Have you identified all the activities, including activities beyond the software development life cycle (SDLC), that go into securing your products and software?
- Do your products and software follow a common set of software security activities or does it vary product by product?
- Do you maintain a current view of all your software assets, including internal code, third-party code, open source, automation scripts, infrastructure-as-code, and other software assets?
- Are you using a Software Bill of Materials (SBOM) that details all components in your software portfolio as part of your risk management processes? Are you prepared to provide SBOMs to your customers?

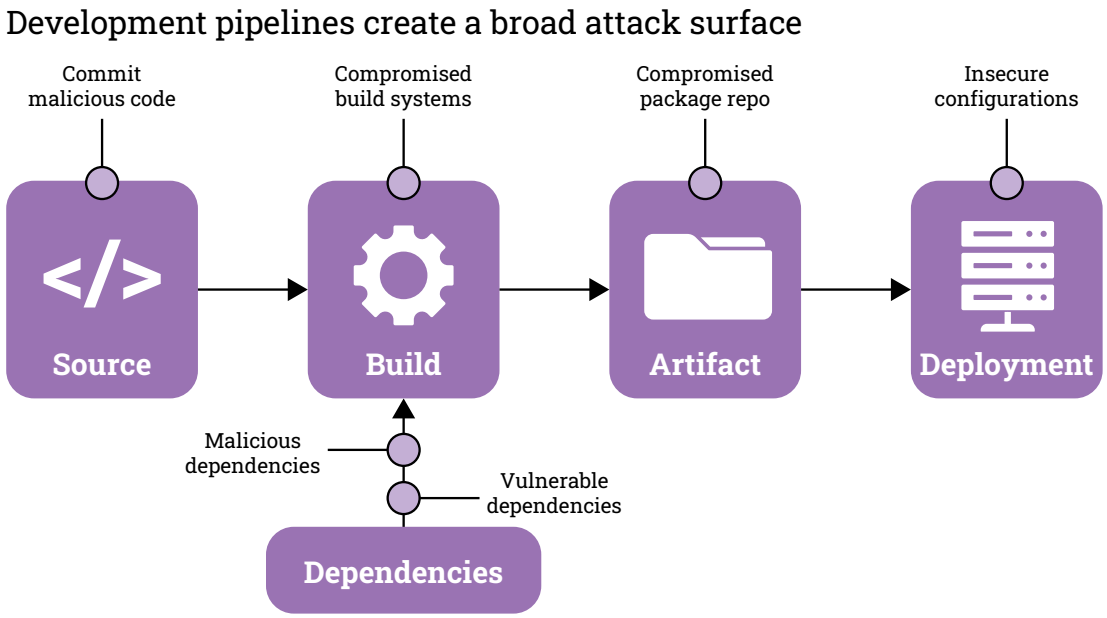
Preparing for NIST SSDF attestation is a complicated process with serious legal consequences. This is where Black Duck can help. Black Duck is an independent provider of Building Security in Maturity Model (BSIMM) assessments. BSIMM is a descriptive model that provides a baseline of observed activities for software security programs, and it provides a common vocabulary for software security programs. Since BSIMM is an acceptable reference for NIST SSDF attestation, we've created an SSDF Readiness Assessment to assist customers in this process.

As part of the overall Black Duck strategy for software supply chain security and risk management, an SSDF Readiness Assessment can help you define gaps and inconsistent implementations in your application security (AppSec) programs. It can also help your company meet your NIST SSDF obligations, as well as help you institute practices that will increase the overall security of your software development.

HOW DOES THE SSDF WORK?

The SSDF standards are meant to help software developers reduce the number of vulnerabilities in deployed software, mitigating the potential effect of exploitation of undiagnosed or untreated flaws, and addressing the fundamental causes of vulnerabilities to prevent future occurrences.

Current development pipeline protocols can make organizations vulnerable at numerous points across the SDLC. In combination with the siloed nature of development, this can leave organizations open to breaches by malicious actors.



The SSDF establishes a common vocabulary for secure software development by implementing four broad activities, supported by 19 practices and 42 tasks. These are enumerated in Section 4 of EO 14028, but the key takeaways are that the SSDF requires tasks to be completed at the program level as well as at the product level, and organizations need to attest on a per-product basis, although some attestation may be possible by product family or group. However, one drawback of this mandate is that while the standards enumerate *what* needs attestation, it does not enumerate *how* to implement controls or how to build an efficient, effective, and scalable program.

WHAT ARE THE FOUR ACTIVITIES OF THE SSDF?

The SSDF is composed of four activities, each of which encompasses several practices. These activities cover the development infrastructure—that is, the places where your organization is touching code, building code, developing code, and distributing code—as well as outline requirements for the software itself. This includes the security requirements, both functional and nonfunctional, that go into the code that you're writing. The four activities are

- **Prepare the Organization (PO)**: This activity requires you to ensure that the organization's people, processes, and technology are prepared to perform secure software development at the organization level, and in some cases, at the level of individual development groups or projects.

  In practice, this means identifying all your security requirements including development infrastructure and software requirements, and communicating these requirements to your teams. It's crucial that you're preparing the people in your organization by defining roles and responsibilities, and setting up role-specific training on secure software development practices.

  You must also prepare to secure your tool chains by specifying the tools and tool types your developers require, and how they are integrated into your SDLC. You will need best practices for these tools, and they need to be configured to generate the artifacts required by the SSDF mandate.

  Further, software security check criteria must be defined and implemented, and you must apply processes that gather and protect evidence that the criteria were met.

  Finally, you need to ensure that your development environments and workstations are secure, and that development occurs in separate and protected environments with appropriate security protocols.
- **Protect the Software (PS)**: This activity requires you to attest to how you are protecting all components of your software from tampering and unauthorized access.

  In practice, this means storing and protecting all forms of your code following the principle of least privilege. You need to

ensure the integrity of your software by enabling users to verify that what they get is the code you released. And you will need to securely archive and protect each release, including all relevant artifacts such as release criteria evidence, integrity verification data, and provenance data. You will also need to collect and protect provenance data and share it with customers.

- **Produce Well-Secured Software (PW)**: This activity requires you to attest that you are producing well-secured software with minimal security vulnerabilities in your releases.

  In practice, this means designing software with security in mind. by implementing risk modeling (threat and attack modeling) to define your security requirements. You'll need to document why you made certain decisions, and why you chose one security feature over another. While this might seem onerous, it provides documentation that can be referred to at a later date, should you need to revisit these decisions. You should also implement standard security designs where possible.

  Reviewing these designs with security in mind is a crucial part of this activity, and you must independently verify the security of each team's design. These designs should use, acquire, and develop well-secured components, and maintain their security over time.

  This activity also entails demonstrating that your coding practices meet security requirements. Your teams must define and use build tools that can improve security, as well as define and use the correct security settings of those tools.

  Code reviews are crucial to this process. Teams must determine what combination of static application security testing (SAST), software composition analysis (SCA), interactive application security testing (IAST), and penetration testing works for them, and then perform reviews and fix identified issues. Teams must also test running code with dynamic application security testing (DAST), and determine how and when it is to be done.

- **Respond to Vulnerabilities (RV)**: This activity requires you to attest that you have identified vulnerabilities in your software releases and responded appropriately to address them and prevent similar vulnerabilities from occurring in the future.

  In practice, this means identifying vulnerabilities by gathering vulnerability information from all available sources, determining if the reported vulnerability affects your software, and developing a vulnerability disclosure and remediation policy.

  You must assess the risk of each vulnerability, and then plan and implement a response to prioritize and fix those vulnerabilities. You must also determine the root cause of vulnerabilities, look for patterns in the root causes over time, and determine if similar vulnerabilities exist in other software in your portfolio, so you can fix them. You must also document your updated processes and procedures that address patterns discovered in your root cause analysis.

HOW DOES THIS WORK IN THE REAL WORLD?

Since 2006, Black Duck has used BSIMM to provide a baseline of observable activities for software security programs. Because these programs often use different methodologies and terminology, BSIMM creates a common vocabulary for software security programs. The framework consists of 12 practices divided into four domains.

- **Governance:** Practices that assist organize, manage, and measure a software security program
- **Intelligence:** Practices that result in collections of corporate knowledge used in carrying out software security activities throughout the firm
- **SSDL touchpoints:** Practices linked with analysis and assurance of particular software development artifacts and procedures
- **Deployment:** Practices that integrate with standard network security and software maintenance organizations

Black Duck has performed dozens of SSDF assessments since fall 2023 for customers also undertaking a BSIMM assessment, because of the natural overlap between the two. Some customers have asked for a standalone SSDF assessment. Among the positive trends these assessments discovered were that mature, centralized software security programs and groups were better prepared for the NIST attestation process, in part because they had security champions at the product and team level, and in part because strong, shared, central security services made self-attestation easier.

The most well-prepared organizations were those already complying with industry-specific regulatory frameworks like ISASecure, and flight safety regulations like DO-326A, DO-356A, and others. Among the negative trends the assessments uncovered were that organizations with team-centric business cultures face more challenges with SSDF self-attestation, in part because a lack of strong central security services results in inconsistent and uneven results.

WHAT CAN WE LEARN FROM THESE ASSESSMENTS?

First of all, a word of caution. This data set reflects only the experience of firms with proactive application and product security efforts, so it will not apply universally. However, our experience with combined BSIMM/SSDF assessments has shown some interesting results.

As one might expect, organizations with mature security programs were better prepared to self-attest to SSDF conformance as required by EO 14028. That is, high-maturity programs will have low-touch self-attestation efforts, whereas medium- and low- maturity programs will have high-touch self-attestation efforts.

For the four activities of SSDF attestation, our assessments found the following:

- **Prepare the Organization (PO)**: Role-based training remains a major weakness, and even in those cases where it is offered, it is not being used.
- **Protect the Software (PS)**: There are problems with provenance data handling. While planning was in good shape, implementation had not caught up.
- **Produce Well-Secured Software (PW)**: Secure coding tools provided de facto standards instead of explicitly defined standards, and the security features of build tools were not clearly defined.
- **Respond to Vulnerabilities (RV)**: Immature root cause analysis was a problem, and most organizations were still relying on finding and fixing vulnerabilities instead of working to eradicate classes of vulnerabilities.

Key takeaways included the following:

- Mature, strong software security groups and software security programs provided better readiness for SSDF.
- Organizations with centralized policies, standards, processes, and procedures showed better preparedness and more consistency than those with decentralized policies, processes, and standards.
- Organizations that rely on a central services model to perform or guide software security activities had an easier time assessing their firm, while those that lacked a software security program or that had a weak program found self-attestation more difficult.

Across the board, we found that in most organizations, there was plenty of work to be done.

HOW CAN BLACK DUCK HELP?

Black Duck Security Consulting Services offers SSDF Readiness Assessment consulting. This process can assist your organization in identifying whether your software development practices align with the practices and tasks of the SSDF. It also provides an assessment of which controls are lacking in conformance. This assessment, and the associated corrective recommendations, can be used when completing U.S. government attestations.

Built on the proven BSIMM, this assessment empowers you to analyze and benchmark your software security program against 100+ organizations across several industry verticals. It's an objective, data-driven analysis that allows you to base decisions about resources, time, budget, and priorities as you seek to improve your security posture. The SSDF Readiness Assessment quantifies the security of development environments in addition to the governance, culture, and process measurements of a BSIMM assessment.

Ready to get started?

ABOUT BLACK DUCK

Black Duck® meets the board-level risks of modern software with True Scale Application Security, ensuring uncompromised trust in software for the regulated, AI-powered world. Only Black Duck solutions free organizations from tradeoffs between speed, accuracy, and compliance at scale while eliminating security, regulatory, and licensing risks. Whether in the cloud or on premises, Black Duck is the only choice for securing mission-critical software everywhere code happens. With Black Duck, security leaders can make smarter decisions and unleash business innovation with confidence. Learn more at www.blackduck.com.