SOFTWARE ACQUISITION GUIDE

FOR GOVERNMENT ENTERPRISE CONSUMERS:

Software Assurance in the Cyber-Supply Chain Risk Management (C-SCRM) Lifecycle

PUBLICATION: AUGUST 2024

INFORMATION AND COMMUNICATIONS TECHNOLOGY
SUPPLY CHAIN RISK MANAGEMENT TASK FORCE

FOREWORD

Technology acquisitions are challenged by the level of transparency provided by suppliers of software and cyber-physical devices relative to their development and third-party management practices. While acquisition staff have a general understanding of the core cybersecurity requirements for a particular acquisition, they often lack the ability to assess whether a given supplier has practices and policies in place that better meet the ongoing expectations of enterprise users of the products.

The nature of this problem is called out in the National Cybersecurity Strategy, which highlights that often cybersecurity responsibilities are borne by software operators rather than those best positioned to address the issue—software suppliers. The Information and Communications Technology (ICT) Supply Chain Risk Management (SCRM) Task Force Software Assurance Working Group developed this Software Acquisition Guide (hereafter referred to as the 'guide') in response to the core challenges of software assurance and cybersecurity transparency in the acquisition process, focusing primarily on software lifecycle activities.

The ICT SCRM Task Force is a public-private, cross-sector body organized and co-chaired by the Cybersecurity and Infrastructure Security Agency (CISA), through a National Risk Management Center (NRMC) representative, and representatives from the Information Technology (IT) and Communications Critical Infrastructure Sectors. Created in compliance with the Critical Infrastructure Partnership Advisory Council (CIPAC) to enable the members to deliberate and achieve consensus advice to the federal government, the Task Force serves as the primary mechanism for industry and government collaboration on strategies and policies to address ICT supply chain risks confronted by critical infrastructure owners and operators, civilian federal executive branch departments and agencies, and state, local, tribal, and territorial (SLTT) governments. The Task Force provides advice and recommendations to the federal government, and to private sector owners and operators of critical infrastructure on means for assessing and managing risks associated with the ICT supply chain.

Careful consideration has been made to align to pre-existing work from the National Institute of Standards and Technology (NIST) and CISA, requirements from Office of Management and Budget (OMB) and General Services Administration (GSA), and requirements such as the CISA Secure Software Development Attestation Form[1] (or agency-specific variations of that form).

Further complicating the transparency problem are the labels placed on roles of entities within the software supply chain. For example, is the provider of a piece of software its developer, vendor, integrator, distributor, reseller, or producer? When the software is simple, many of those roles merge, but the more complex the software the more likely that software is created using multiple teams from multiple organizations—many of

[1] https://www.cisa.gov/secure-software-attestation-form

which have no direct contractual relationship with the buyer of the software but where the decisions made by other entities within the software supply chain may have direct bearing on the security of the software. Appendix A contains a complete glossary of terms used in this guide and includes source references for those terms. This guide also aligns with both the CISA secure by design principles for software producers to compete based on security, and the software engineering principles of secure by design for software development and deployment configurations that are secure by default. When combined, there is a focus placed on software operators demanding secure software from the outset while providing a context for interaction among suppliers and enterprise customers and consumers. In October 2023, CISA released an updated version of the "Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software" continuing the push for an ecosystem of secure software and systems, and shifting the onus onto software suppliers rather than the current paradigm where the consequences of vulnerable exploitable systems and software overwhelmingly fall on customers and consumers—the enterprise users.

With a focus on security as a criterion during the purchasing cycle, **consumer expectations for secure software and products are articulated in acquisition and procurement activities and contracts.**

Within government agencies, the mission owner and contracting or requirements office, coordinating with the Office of Information Technology (or similar office, as applicable), have responsibility for communicating enterprise expectations for secure software. This document aims to provide best practices and recommendations to support their efforts on obtaining secure software for their agencies. Since it remains challenging for suppliers to make the required security investments and efforts without associated demand, the CISA Secure by Design guidance also emphasizes the need for consumer demand: ***"…just as we seek to create a pervasive secure by design philosophy within software manufacturers, we need to create a 'secure by demand' culture with their customers."***[2]

This guide focuses on the "secure by demand" elements by providing recommendations for agency personnel, including mission owners and contracting staff or requirements office to engage in more relevant discussions with their enterprise risk owners (such as CIOs and CISOs) and candidate suppliers such that better, risk-informed decisions can be made associated with acquisition and procurement of software and cyber-physical products. The information and insights gathered from suppliers help raise the bar on cybersecurity transparency.

[2] Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software – Page 7

TABLE OF CONTENTS

SOFTWARE ASSURANCE (SWA) WORKING GROUP (WG) MEMBERS

LEADERSHIP TEAM FOR SWA WG:

NAME	COMPANY
Tim MACKEY	Black Duck Software
Joe JARZOMBEK	Synopsys and Department of Homeland Security (DHS) – Retired
Justin MURPHY	Cybersecurity and Infrastructure Security Agency (CISA)

SWA WG CONSISTS OF THE FOLLOWING MEMBERS:

NAME	COMPANY
Dick BROOKS	Business Cyber Guardian
Carol WOODY	Carnegie Mellon University's Software Engineering Institute
Rebecca ADAMS Matthew DAVEY Sabeen FAWAZ Allan FRIEDMAN Laura HERSHON Brian PAAP Katie WILLERS	CISA
Steven CARPENTER	Federal Communications Commission
Robert SALVIA	Fortress Information Security
Kevin FUNK Michael THOMPSON	General Services Administration
Tommy GARDNER	HP
Jon AMIS	LMI
Chris ANDERSON	Lumen
William BARTHOLOMEW	Microsoft
Keith HILL Bob MARTIN	MITRE
Kanitra TYLER	NASA
Glen DUKE Kesha Hill Carol LEE Tim STEVENS	National Security Agency
Sridhar BALASUBRAMANIAN	NetApp
Kathy LYONS-BURKE Ismael GARCIA	Nuclear Regulatory Commission
Hawa IBRAHIM	T-Mobile

OVERVIEW

Cyberattacks often target an enterprise's use of software for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of data or stealing controlled information. Many well-known attacks have exploited vulnerabilities and weaknesses in software and within software supply chains; an issue that spans both proprietary and open source software, which impacts both private sector and government enterprises. This issue has prompted an increased need to rebalance responsibilities for cybersecurity risks between software suppliers and consumers. Both parties need an increased awareness of software supply chain security risks and the potential for them to be exploited by cybercriminals and weaponized by nation-state adversaries using similar tactics, techniques, and procedures—but the responsibility ultimately lies with the software suppliers to take ownership of their customers' security outcomes.

The Executive Order on Improving the Nation's Cybersecurity (EO 14028) established new requirements to secure the U.S. federal government's software supply chain. The EO requirements involve systematic reviews, process improvements, and security standards for software suppliers and developers, and for customers who acquire and use software for the federal government. The U.S. government (USG) has adopted key practices from the NIST Secure Software Development Framework (SSDF) to serve as a compendium of suggested practices to be executed by the supplier, developer, and customer stakeholders to help ensure a more secure software supply chain.

Customers (agency mission owners, their acquisition and procurement organizations, and enterprise risk owners such as CIOs and CISOs) may use this guide as a reference for describing, assessing, and measuring suppliers' security practices relative to the software life cycle. This is applicable to more than the U.S. federal government. The suggested practices listed herein may be applied across government at all levels, and for the industry acquisition, deployment, and operational phases of a software supply chain.

KEY CONSENSUS TERMS

The term ***"supplier"*** or ***"software supplier"*** is used extensively throughout this guide to refer to the organization that is directly providing a software powered solution under consideration for a contract.

The term ***"software operator"*** is used throughout this guide to refer to any entity that operates software or cyber-physical devices for the benefit of a user, consumer, or customer. Software operators are expected to understand all the deployment assumptions, requirements, and cybersecurity considerations made by entities supplying the software. One example of a software operator is an IT department.

A complete glossary of terms used in this guide can be found in Appendix A.

Modern software supply chains are incredibly complex and may be made up of hundreds, if not thousands, of independent development teams when the usage of open source libraries is factored in. [3,4] Understanding how open source is managed within software being acquired is a key consideration that this guide addresses.

Suppliers often assume the responsibility of acting as a liaison between the customer and software development teams. Some of the activities involved in fulfilling this responsibility include ensuring the integrity and security of software in contractual agreements, software releases and updates, notifications, and mitigations of vulnerabilities. Recommended best practices to aid suppliers, customers, and acquisition agents in these activities are contained in the control and supporting task questions in this publication.

While there are many existing practices and baselines to assess the cybersecurity posture of an application or software provided as a service, most focus on deployment requirements and the associated data protections and privacy requirements. Existing C-SCRM practices and controls, as illustrated in Figure 1, have minimal overlap with more software development focused efforts like the SSDF. This guide bridges each of these control groups to provide a software assurance perspective covering software supply chain risks.

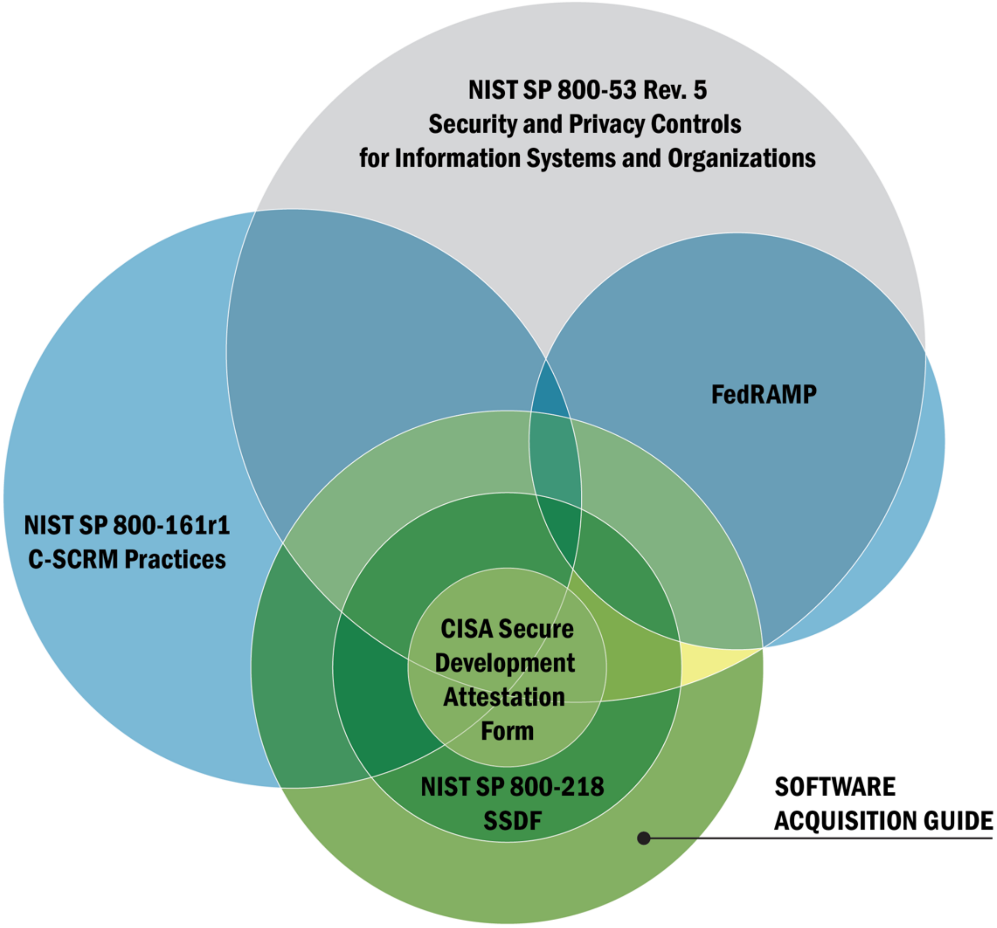

Figure 1. Notional overlap of major cybersecurity control efforts

[3] Synopsys 2024 Open Source Security and Risk Analysis Report - Over 500 components per commercial application: https://www.synopsys.com/blogs/software-security/open-source-trends-ossra-report.html
[4] GitHub Octoverse – Average package contributors and dependencies: https://octoverse.github.com/2019/

From the perspective of software customers, those who acquire and use software products, additional guidance will provide industry best practices and principles whereby the customer should seek adherence. These principles include contextual security requirements for planning and maintaining the security of software and the underlying infrastructure (e.g., environments).

This guide uses the term “product” to refer to software independent of its delivery or deployment model and is not constrained to commercial or contracted software. As such, a product may include:

- Software delivered with cyber-physical devices such as firmware in an Internet of Things device;
- Traditional desktop applications and mobile device applications; and
- Server-based software, including that provided via Software as a Service (SaaS), cloud, or distributed computing models.

This guide is intended to provide context of relevant USG guidance in terms of questions that should be addressed concerning means to mitigate risk exposure to enterprises attributable to software that is obtained from third parties, especially for high assurance and medium assurance environments. This guide goes beyond attestation forms that are a necessary starting point to addressing risks passed to using enterprises.

Acquisition and procurement staff, such as those in a requirements’ office, could leverage this guide to initiate discussions with their cybersecurity staff and enterprise risk owners, such as CIOs and CISOs. By walking through the questions included in this guide, the various stakeholders can discuss means for mitigating risk during the market research phase, performance work statement development, product, service, or solution evaluation, supplier selection, and during post-award monitoring.

SECURE BY DESIGN–FOCUS ON THE DEMAND SIDE

Security by design is achieved when the decision-making processes from detailed software design, through implementation and implementation decisions, product testing, packaging of software into a shippable product, default deployment configurations, and vulnerability management operate in harmony to create a software-enabled product that has security at its core and not as an afterthought. When consumers of software demand transparency from their suppliers, context for security discussions among suppliers and enterprise customers and consumers is achieved. In October 2023, CISA released an updated version of the "Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software" (Principles and Approaches), continuing to advocate for an ecosystem of secure by design software and systems, and shifting the responsibility onto software suppliers rather than the current paradigm where the consequences of vulnerable exploitable systems and software overwhelmingly fall on customers and consumers—the enterprise users.

- ***"'Secure by design' means that technology products are built in a way that reasonably protects against malicious cyber actors successfully gaining access to devices, data, and connected infrastructure."*** (Source: Principles and Approaches, page 8) This sets an expectation for suppliers and includes how suppliers should take ownership of security outcomes for their customers, demonstrate radical transparency and accountability, and lead from the top by making secure by design a top business priority. This applies to the entire software development lifecycle (SDLC)/product lifecycle.

- ***"'Secure by default' means products are resilient against prevalent exploitation techniques out of the box without added charge…without end-users having to take additional steps to secure them…make customers acutely aware when they deviate from safe defaults…"*** (Source: Principles and Approaches, page 9) This means products come with a baseline level of security by default that do not need significant effort by customers to 'harden' the products and software against exploitation. The CISA guidance also states that, "The complexity of security configuration should not be a customer problem…," (Source: Principles and Approaches, page 9) and that customers are not charged extra for implementing added security configurations.

- ***'Secure by demand' means customer expectations for secure software and products are articulated in acquisition and procurement activities and contracts.*** Within government agencies, employees performing requirements and contracting functions have significant roles in communicating enterprise expectations for secure software. Agencies can provide more complete guidance to support their efforts to focus on obtaining secure software. Since it remains challenging for suppliers to make the required security investments and efforts without associated demand, the updated CISA guidance also emphasizes the need for consumer demand: ***"…just as we seek to create a pervasive secure by design philosophy within software manufacturers, we need to create a 'secure by demand' culture with their customers."*** (Source: Principles and Approaches, page 7) To focus on the need to correct market failures of cybersecurity, increased demand for secure products and software from enterprise customers and consumers is needed in acquisition and procurement because it is directly tied to customer demand and spending, which impacts supplier revenue and profits.

In the Recommendations for Customers section of Secure by Design Principles and Approaches, CISA recommends that enterprise customers and consumers hold their suppliers accountable for the security outcomes of their products. This means proverbially ‘voting with your wallet’ and making purchases that prioritize secure by design in products and software. It also means ensuring products are properly vetted by internal security staff prior to procurement, utilizing contractual language, and Requests for Information (RFIs)/Requests for Proposal (RFPs) to help influence specific product and software purchases.

CISA emphasizes that information technology (IT) departments must have the executive support of the organization when enforcing these purchasing decisions.

If insecure or risky products are purchased, then those decisions and inherent risks should be formally documented and approved by the senior business executives who are the enterprise risk owners. This ensures the responsibility does not inherently fall on security teams when the business decisions are driving the purchasing activity and risk acceptance. In other words, the business owns the risk, and documenting risks and having them formally signed off on can help change behavior.

CISA also calls for IT and security leaders to collaborate with industry peers to rally around services and products that value and prioritize the secure by design principles, and this collective consensus with spending decisions can help incentivize suppliers to prioritize secure products, associated with actual customer demand.

Market incentives must shift to change cybersecurity behaviors and address the associated security outcomes.

As one example, in May 2024, CISA launched the Secure by Design pledge.[5] This pledge provides an opportunity for developers of software solutions throughout software supply chains to publicly commit to demonstrating measurable progress in seven areas critical to cybersecurity success throughout the software product lifecycle. These areas include reduction in classes of vulnerabilities and increasing the use of multi-factor authentication—both of which directly impact the risk associated with use of a supplier’s software. Enterprise customers can look to actions resulting from the pledge as valuable signals of a software supplier’s approach to secure development.

This guide focuses on the secure by design elements by providing guidance for agency personnel, including mission owners and contracting staff or requirements officers, to engage in more relevant communications with suppliers to understand their software supply chain activities such that better, risk-informed decisions can be made associated with acquisition and procurement of software products and services. Consumers voting with their procurement funds, by selecting suppliers and products that prioritize secure by design/default products and services, can function as that market signal to drive systemic changes across the software supplier ecosystem.

In summary, the CISA secure by design guidance lays out three primary software product security principles that encourage suppliers to adopt and prioritize:

- Take ownership of customer security outcomes;
- Embrace radical transparency and accountability; and
- Build organizational structure and leadership to achieve these goals.

This guide focuses on Supplier Governance and Attestations and provides questions that agency personnel, including mission owners and contracting staff or requirements officers, could use in communications with suppliers to address software product security principles within the context of secure software lifecycle practices. For suppliers, this could serve as a basis for creating a Plan of Action and Milestones (POA&M) if any improvements are needed.

[5] https://www.cisa.gov/securebydesign/pledge

ORGANIZATION AND USAGE

This guide is organized into five primary sections with each section having its own set of controls and clarifying tasks, including:

- 19 CONTROL questions for Supplier Governance and Attestations;
- 8 CONTROL questions for Software Supply Chain;
- 30 CONTROL questions for Secure Software Development;
- 12 CONTROL questions for Secure Software Deployment; and
- 8 CONTROL questions for Vulnerability Management.

Of these, 25 CONTROL questions could be skipped if the supplier provides a CISA Secure Software Development Attestation Form, or equivalent forms such as the GSA 7700 Secure Software Development Attestation Form, without the need for a POA&M. Affirmatively answering all the Supplier Governance and Attestation questions enables all remaining CONTROL questions in each CONTROL category to be skipped in subsequent sections of this guide.

Software suppliers are expected to understand the security controls used within their development environments, apply similar controls to their software supply chain, and provide guidance to software operators.

An associated fillable spreadsheet has been provided to be used with this guide to assist respondents.

RECOMMENDED USE BY SUPPLIERS

- Suppliers should provide their organization's point of contact (name, email, and phone number) for questions, support, or additional information related to this guide.
  - Suppliers should consider designating one primary coordinator from their organization who will collaborate with the appropriate teams to collect and compile responses for each section.
  - Each section in this guide is designed to be relevant to a different aspect of the SDLC, which may require engagement with individuals responsible for acquisition, procurement, development, supply chain, security, etc.
- If requested, suppliers should be prepared to attach supporting documents for their responses to the questions in this guide. Links may be provided if documentation is available online and accessible.
- A companion spreadsheet is available that can be used to expedite responses.

- If the respondent(s) can provide affirmative answers to the “supplier governance and attestation control questions,” then responses to relevant control questions (and associated task questions) may be skipped in subsequent sections, as indicated.

- The questions are organized into a series of CONTROL questions with most CONTROL questions having a series of informative TASK questions. For each CONTROL question, it is expected that the software supplier will provide a simple response of “Yes,” “No,” “N/A,” or “Partial.”
  - A response of “Yes” to a CONTROL question implies that the supplier believes they have appropriate controls in place to meet the expectations of the CONTROL question, and that they understand there may be a requirement to produce supporting documentation. If a “Yes” response is provided, there is no requirement to respond to any of the underlying TASK questions for that CONTROL question.
  - A response of “No” to a CONTROL question implies that the supplier believes they do not have appropriate controls in place to meet the expectations of the CONTROL question. If the supplier views the CONTROL question as ambiguous, then the TASK questions should provide clarification. A “Yes” response to any of the TASK questions associated with the CONTROL question implies they can respond “Partial” to the CONTROL question. A “Yes” response to all the TASK 8 questions implies that they can respond “Yes” to the CONTROL question.
    Note: Procurement teams can also review the TASK questions to understand the implications of a “No” or “Partial” response.
  - A response of “N/A” implies the supplier believes that in the operational context of their software, the CONTROL question does not apply.
    Note: For the purposes of assessing risk, it is recommended that procurement teams review the TASK questions to determine if further investigation is required.
  - A response of “Partial” to a CONTROL question is expected when the software supplier believes that they meet at least one, but not all, of the subordinate tasks. For “Partial” responses, the software supplier should respond with a “Yes,” “No,” or “N/A” for each of the subordinate TASKS associated with the CONTROL.
    Note: Only in the event of a “Partial” response should a software supplier be expected to respond to the subordinate TASK questions of a CONTROL.

RECOMMENDED USE BY ACQUIRERS AND SUPPLIERS WHO INTEGRATE COMPONENTS FROM OTHERS

Since software risk is primarily experienced when the software is operating, it is important for users of this guide to apply the context of their enterprise usage requirements and the level of software assurance associated with those requirements when evaluating which controls are most important. This may mean that additional scrutiny is placed on the operational controls related to deployment and vulnerability management.

Questions in this guide can be used by enterprise users to inform the structuring of contract language and evaluation criteria to convey expectations more explicitly to candidate suppliers.

Requirements organizations and acquisition/procurement teams could use this guide to facilitate pre-procurement communications with prospective suppliers. As a minimum, enterprise users and those who incorporate the use of third-party components should discuss and take into consideration the CONTROL questions listed in Supplier Governance and Attestations to gain an understanding of the residual risk exposure associated with prospective suppliers’ products or services.

The format of this guide is intended to be used to gather an initial and consistent baseline to assess the cybersecurity development practices used by a software supplier in the software supply chain associated with the software being assessed. Additional follow-up questions for the supplier, including supporting documentation, may be warranted. While it is hoped that most of the software being procured is developed, sustained, and deployed using the cybersecurity baseline outlined in this guide, this situation is contextual. Accordingly, there are additional use cases for this document beyond a review of cybersecurity practices during a late-stage procurement review. Two examples are:

- Development of an RFP or RFI: Buyers creating a new RFP or RFI will naturally have contextual knowledge of the operational requirements for the software that they are attempting to obtain. This contextual knowledge may result in higher levels of software assurance being required, or specific CONTROLs taking priority. Making cybersecurity requirements part of the RFP or RFI process will help ensure that any procured software meets the software assurance and cybersecurity targets from the outset.

- Creation of a POA&M: If the preferred supplier for a contract doesn't provide an affirmative response to all the required CONTROL questions, then responses to the CONTROL questions and associated TASK questions could be used in the creation of a POA&M.

DISCLAIMER

Member organizations of the ICT SCRM Task Force, including CISA, do not endorse any commercial entity, product, company, or service, including any entities, products, or services mentioned within this document. Any reference to specific commercial entities, products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply endorsement, recommendation, or favoring by CISA or any member organization of the ICT SCRM Task Force.

SUPPLIER GOVERNANCE AND ATTESTATIONS

GOVERNANCE CONTROL QUESTIONS

The following governance CONTROL questions are intended to reduce the reporting burden of this guide. For most of these questions, a simple ‘Yes’ or ‘No’ is required. A ‘Yes’ response to the question enables a series of CONTROLs in other sections to be skipped. The exceptions to this are the last CONTROLs of this section.

CONTROL.GOV.01 Does the supplier provide a CISA Secure Software Development Attestation Form, or equivalent such as the GSA 7700 Secure Software Development Attestation Form, without need for a POA&M, signed by the supplier’s designated employee (Chief Executive Officer or designee that can bind the supplier)?

If ‘No,’ then most of the subsequent CONTROL questions should be addressed.

If ‘Yes,’ and a POA&M was not needed, then the following CONTROL questions and associated TASK questions can be skipped:

— for Supply Chain: SC.04, SC.07, SC.08
— for Software Development: DEV.03,DEV.07, DEV.08, DEV.09, DEV.10, DEV.11, DEV.12, DEV.14, DEV.20, DEV.21, DEV.22, DEV.23, DEV.26, DEV.27, DEV.28, DEV.30
— for Software Deployment: DEP.07, DEP.09, DEP.11
— for Vulnerability Management: VULN.01, VULN.04, VULN.07

CONTROL.GOV.02 Does the supplier maintain provenance data for internal and third-party components?

If ‘Yes,’ then the following CONTROL questions and associated TASK questions can be skipped:
— for Supply Chain: SC.01, SC.04, SC.08
— for Software Development: DEV.03, DEV.12, DEV.16, DEV.30

CONTROL.GOV.03 Has the supplier's product(s) or product line employed automated tools or comparable processes including, but not limited to, log management and patch management to maintain integrity of software supply chains and to check for and mitigate security-relevant vulnerabilities in binary, source code, development and build systems?

If ‘Yes,’ then the following CONTROL questions and associated TASK questions can be skipped:
— for Supply Chain: SC.07
— for Software Development: DEV.09, DEV.24
— for Software Deployment: DEP.03, DEP.04, DEP.09, DEP.12
— for Vulnerability Management: VULN.02, VULN.03, VULN.04, VULN.05, VULN.06, VULN.08

CONTROL.GOV.04 Has all the software (including third party and open source) to be delivered undergone rigorous code analysis and multi-level testing according to the supplier's documented testing procedures? Foremost in this testing is the identification of code weaknesses and software vulnerabilities, including those listed in the DHS CISA Known Exploited Vulnerabilities (KEV) Catalog with vulnerable components either patched, rebuilt, or otherwise mitigated.

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:
— for Software Development: DEV.21, DEV.22, DEV.27, DEV.28, DEV.30
— for Vulnerability Management: VULN.02, VULN.05, VULN.06

CONTROL.GOV.05 Does the supplier use industry standards or frameworks for implementing vulnerability scanning and vulnerability management, and to communicate mitigation status for any unpatched vulnerabilities using machine-readable formats, such as a Common Security Advisory Framework (CSAF) Security Advisory, a NIST defined VDR, or a VEX document, for the current version of the application and all future versions and updates?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:
— for Software Development: DEV.30
— for Vulnerability Management: VULN.01, VULN.03, VULN.07,VULN.08

CONTROL.GOV.06 Does the supplier use a secure by default approach for software deployment processes?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:
— for Software Development: DEV.29
— for Software Deployment: DEP.01, DEP.07, DEP.11

CONTROL.GOV.07 Does the supplier use secure by design principles ensuring that their product been developed and built in secure environments using community or industry recognized frameworks, certified against those applicable standards, and tested in an environment following zero-trust principles?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:
— for Software Development: DEV.01, DEV.03, DEV.07, DEV.08, DEV.09, DEV.11, DEV.12, DEV.16, DEV.18, DEV.20, DEV.23, DEV.26
— for Software Deployment: DEP.10

CONTROL.GOV.08 Prior to incorporating any third-party components in its software components, products or services provided to customers, does the software supplier require third-party software suppliers to produce a software bill of materials (SBOMs), to establish a vulnerability disclosure policy, and to follow "NIST Guidance" as specified in OMB M-22-18?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:
— for Software Development: DEV.10, DEV.21, DEV.22
— for Software Supply Chain: SC.01, SC.04, SC.07, SC.08

CONTROL.GOV.09 Does the supplier provide a machine-readable SBOM meeting minimum requirements defined by National Telecommunications Information Administration (NTIA) or successor guidance as published by CISA that covers all software components of the product being delivered to the customer organization?

If 'Yes,' then the following Software Supply Chain CONTROL questions and associated TASK questions, can be skipped: SC.02, SC.08

CONTROL.GOV.10 Does the supplier define and enforce policies governing the responsible use of open source libraries and software, AI generated code, and AI powered toolchains?

If 'Yes,' then the following Software Supply Chain CONTROL questions and associated TASK questions can be skipped: SC.03, SC.04, SC.06

CONTROL.GOV.11 For the products or services being provided, has the supplier received a successful third-party FedRAMP High or Moderate Baseline certification?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:

— for Software Deployment: DEP.01, DEP.02, DEP.03, DEP.04, DEP.05, DEP.06, DEP.07, DEP.08, DEP.09, DEP.10, DEP.11, DEP.12

— for Vulnerability Management: VULN.01, VULN.03, VULN.06, VULN.07

CONTROL.GOV.12 Does the supplier have protections and verification methods in place with third-party suppliers (for software, network, and cloud services) for products or services provided to the customer that are commensurate with contractually specified government protection levels and trust relationships?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:

— for Software Development: DEV.04

— for Software Deployment: DEP.06, DEP.08, DEP.09

CONTROL.GOV.13 Does the software supplier establish and resource a Cyber Supply Chain Risk Management (C-SCRM) Program that includes attack and risk modeling and incident management and response?

If 'Yes,' then the following CONTROL questions and associated TASK questions can be skipped:

— for Supply Chain: SC.05

— for Software Development: DEV.06, DEV.15

— for Software Deployment: DEP.02

CONTROL.GOV.14 Does software supplier provide role-based SDLC-related training and have qualified personnel and/or automated processes that contribute to all parts of the SDLC, including secure architecture, development, testing and threat modeling?

If 'Yes,' then the following Software Development CONTROL questions and associated TASK questions can be skipped: DEV.04, DEV.05, DEV.18

CONTROL.GOV.15 Does the supplier have and enforce defined policies for software security requirements, including securely storing all forms of code (for source code, executable code, binaries, and configuration-as-code), release artifacts, and associated integrity verification information for each release?

If "Yes," then the following Software Development CONTROL questions and associated TASK questions can be skipped: DEV.02, DEV.10, DEV.13.

CONTROL.GOV.16 Does the supplier have policies and procedures that ensure the maximum use of software modules providing standardized implementations of security features and services or the reuse of well-secured software components developed in-house following SDLC processes?

If "Yes," then the following Development CONTROL questions and associated TASK questions can be skipped: DEV.17, DEV.19.

CONTROL.GOV.17 Does the supplier have policies and procedures to use built-in checks and protections supported by programming languages or environments, both compiled and interpreted during development and in the software shipped/distributed?

If "Yes," then the following Software Development CONTROL questions, and associated TASK questions can be skipped: DEV.07, DEV.24, DEV.25.

CONTROL.GOV.18 Do the software supplier's procurement, outsourcing, and contractual agreements, such as Service Level Agreements (SLAs), stipulate that their sub-suppliers and/or service providers follow secure SDLC practices, scan for undocumented, unused, or obsolete functions, and notify the software supplier of identified vulnerabilities or security incidents?

If "Yes," then the following CONTROL questions, and associated TASK questions, can be skipped:
— for Supply Chain: SC.01, SC.04
— for Software Deployment: DEP.02, DEP.05.

If "No," supplier should provide details on vulnerability disclosure and incident response processes within their digital ecosystem.

CONTROL.GOV.19 Does the supplier provide license terms that permit the using enterprise, agency, organization or a trusted third party to scan the delivered software relative to provenance and security?

If "No," provide any restrictive language that prohibit such scanning.

SOFTWARE SUPPLY CHAIN CONTROLS

Software is increasingly composed of, or reliant upon, libraries created by third-party development teams. These libraries might be open source, commercial, or third-party contracted, and each team may create their libraries using any combination of open source, commercial, or third-party contracted libraries. The lack of visibility into the design, development, and implementation decisions made by third-party teams poses risk to all software.

SUPPLY CHAIN CONTROL AND TASK QUESTIONS

CONTROL.SC.01 Does the supplier have policies and procedures to validate the development of software used, including all libraries, and with the exception of Integrated Development Environment (IDE), compiler, build, packaging, and Continuous Integration/Continuous Delivery (CI/CD) tools, occurs using supplier employees or vetted contract employees?

TASK.SC.01.01 Were sub-contractors or third-party contracted development teams used in the creation of the software?

TASK.SC.01.02 Does the product include or depend upon Commercial off-the-shelf/Government off-the-shelf (COTS/GOTS) or commercial libraries?

TASK.SC.01.03 Does the product include or depend upon open source libraries?

TASK.SC.01.04 Does the product include or depend upon AI generated source code or libraries?

CONTROL.SC.02 Does the supplier create a validated SBOM in an NTIA or CISA approved machine-readable format with NTIA or CISA defined minimum fields for all releases of the software, including updates?

TASK.SC.02.01 Does the supplier document its SBOM creation processes?

TASK.SC.02.02 Does the supplier publish its SBOM in an accessible location?

TASK.SC.02.03 Is the SBOM provided in an NTIA or CISA approved machine-readable format?

TASK.SC.02.04 Does the supplier provide a conformance or attestation to ensure the SBOM is accurate and complete?

CONTROL.SC.03 Does the supplier perform cyber risk management on outsourced, or third-party contracted, software development?

TASK.SC.03.01 Does the supplier review any differences in the IDE, compiler, build, packaging, and CI/CD tools used by a third-party contractor relative to those used by internal development teams?

TASK.SC.03.02 Are third-party contractors permitted to sub-contract for contracted work?

TASK.SC.03.03 Are periodic threat models performed as part of a third-party contractor risk management effort?

CONTROL.SC.04 Does the supplier have a defined policy and process for open source governance as typically established by an Open Source Program Office (OSPO)?

TASK.SC.04.01 Is there a defined review process for any new usage of an open source component prior to its first usage in the software?

TASK.SC.04.02 Is there a process to identify and remediate known vulnerabilities in open source components as minimally disclosed in the National Vulnerability Database (NVD)?

TASK.SC.04.03 Is there a process to identify abandoned, unmaintained, obsolete, or compromised open source libraries?

TASK.SC.04.04 Is there a process to perform ongoing security testing on open source libraries used within the software?

CONTROL.SC.05 Does the software supplier establish and resource a C-SCRM Program?

TASK.SC.05.01 Does the software supplier obtain executive leadership support for C-SCRM?

TASK.SC.05.02 Does the software supplier have established C-SCRM policies across enterprise-levels?

TASK.SC.05.03 Does the software supplier have an established C-SCRM governance structure?

TASK.SC.05.04 Does the software supplier have well-documented, consistent, and validated C-SCRM processes?

TASK.SC.05.05 Does the software supplier establish a C-SCRM threat awareness program?

TASK.SC.05.06 Does the software supplier have a quality and reliability program?

TASK.SC.05.07 Does the software supplier integrate C-SCRM into acquisition/procurement policies?

TASK.SC.05.08 Does the software supplier determine impact levels and categorize/assess its systems according to those impact levels (e.g., FIPS 199 impact levels)?

TASK.SC.05.09 Does the software supplier have defined, explicit roles for C-SCRM?

TASK.SC.05.10 Does the software supplier have adequate and dedicated C-SCRM resources?

TASK.SC.05.11 Does the software supplier have a defined C-SCRM control baseline?

TASK.SC.05.12 Does the software supplier have C-SCRM internal checks and balances to assure compliance?

TASK.SC.05.13 Does the software supplier have a supplier management program?

CONTROL.SC.06 Does the software supplier have and enforce policies covering the usage of AI generated code (e.g., ChatGPT or GitHub CoPilot) or code otherwise generated by a third-party tool or service (e.g., Low-Code or No-Code) within its software supply chain?

TASK.SC.06.01 Does the supplier include a review for AI generated code as part of its third-party component review process?

TASK.SC.06.02 Does the supplier perform a risk assessment to determine the impact of AI generated code on the security of the software under review?

TASK.SC.06.03 Does the supplier validate the software license implications covering the usage of AI generated code?

TASK.SC.06.04 Does the supplier review that the usage of AI or cloud powered developer tools aligns with the defined policy?

TASK.SC.06.05 Does the supplier use automated tooling to identify where AI or cloud generated code is used within the software?

TASK.SC.06.06 Does the supplier maintain an approved list of AI code generation tools?

TASK.SC.06.07 Does the supplier perform ongoing periodic reviews for data leakage associated with AI code generation tools?

CONTROL.SC.07 Does the software supplier acquire and maintain current well-secured, vetted software components (e.g., software libraries, modules, middleware) from commercial, open source, and other third-party developers for use by the organization's software throughout the lifespan of the software?

TASK.SC.07.01 Does the software supplier review and evaluate third-party software components and their security aspects in the context of their expected use?

TASK.SC.07.02 If a third-party component is to be used in a substantially different way than when initially approved for use, does the software supplier perform the review and evaluation again with that new context in mind?

TASK.SC.07.03 Does the software supplier determine secure configurations for software components, and make these available (e.g., as configuration-as-code) so developers can readily use the configurations?

TASK.SC.07.04 Does the software supplier implement processes to update deployed software components to newer versions?

TASK.SC.07.05 Does the supplier's software update process include retaining older versions of software components until all transitions from those versions have been completed successfully?

CONTROL.SC.08 Does the software supplier obtain and manage provenance information (e.g., SBOM, source composition analysis, binary software composition analysis) for each software component?

TASK.SC.08.01 Does the software supplier analyze provenance information to better assess the risk that the component may introduce?

TASK.SC.08.02 Does the software supplier verify, safeguard, maintain provenance data for all components of each software release (e.g., in an SBOM)?

TASK.SC.08.03 Does the software supplier make the software component provenance data available to the organization's operations and response teams to aid them in mitigating software vulnerabilities?

TASK.SC.08.04 Does the software supplier protect the integrity of provenance data, and provide a way for recipients to verify provenance data integrity?

TASK.SC.08.05 Does the software supplier establish one or more software repositories to host sanctioned and vetted open source components?

TASK.SC.08.06 Does the supplier maintain a list of organization approved commercial software components and component versions along with their provenance data?

TASK.SC.08.07 Does the software supplier designate that only organization approved components be included in software to be developed?

TASK.SC.08.08 Does the software supplier update the provenance information every time any of the software's components are updated?

TASK.SC.08.09 If the supplier cannot determine the integrity or provenance of acquired binaries, do they verify the source code's integrity, security, and provenance, and rebuild the binaries from source code?

TASK.SC.08.10 Does the software supplier make the software component provenance data available to software acquirers in accordance with the organization's policies?

TASK.SC.08.11 For commercial and cloud software, does the supplier include provenance attributes such as supplier ownership or control, or Data Universal Numbering System (DUNS) verification?

SECURE SOFTWARE DEVELOPMENT CONTROLS

Software development teams are expected to deliver software that is well designed, implemented, and tested. Such designs are commonly referred to as "secure by design." Ensuring that these activities occur on a consistent and reliable basis requires a set of controls over how software development teams function and the various criteria used to release, maintain, and update that software. This section is based on and aligned with NIST's SSDF, SP800-218 v1.1. If an alternative framework is used by the supplier, the SSDF includes a reference column for each task that may prove valuable.

DEVELOPMENT CONTROL AND TASK QUESTIONS

CONTROL.DEV.01 Does the software supplier identify, document, and maintain all security requirements for the organization's software development infrastructures and processes?

TASK.DEV.01.01 Does the software supplier have defined policies for establishing and maintaining secure software development infrastructures and the component elements of those infrastructures (such as build and staging systems) throughout the SDLC?

TASK.DEV.01.02 Does the software supplier have defined policies for establishing and maintaining secure software development infrastructure and processes throughout the SDLC?

TASK.DEV.01.03 Does the software supplier review and update security requirements at least annually?

TASK.DEV.01.04 Does the software supplier review and update security requirements if a major software development security incident occurs?

CONTROL.DEV.02 Does the software supplier identify, document, and maintain all security requirements for organization-developed software to meet?

TASK.DEV.02.01 Does the software supplier have defined policies that specify risk-based software architecture and design requirements (e.g., modular code, security component separation)?

TASK.DEV.02.02 Does the software supplier have defined policies specifying the organization's software security requirements?

TASK.DEV.02.03 Does the software supplier perform risk assessments of applicable technology stacks?

TASK.DEV.02.04 Does the software supplier have defined policies specifying what needs to be archived for each software release?

TASK.DEV.02.05 Does the supplier maintain the security requirements over time?

TASK.DEV.02.06 Does the software supplier ensure that its policies cover the entire software life cycle?

TASK.DEV.02.07 Does the supplier specify how long archives need to be retained based on the SDLC model, software end-of-life, and other factors?

TASK.DEV.02.08 Does the supplier notify users of the impending end of software support and the date of software end-of-life?

TASK.DEV.02.09 Does the software supplier have established processes for handling requirement exception requests?

TASK.DEV.02.10 For any exceptions, does the supplier have a process to periodically review all approved exceptions?

CONTROL.DEV.03 Does the software supplier communicate security acceptance criteria to all third parties who will provide commercial software components to the organization for reuse by the organization's own software?

TASK.DEV.03.01 Does the software supplier have a defined set of core security requirements for third-party software components?

TASK.DEV.03.02 Does the supplier incorporate security requirements in all acquisition documents, software contracts, and other agreements with third parties?

TASK.DEV.03.03 Does the software supplier have defined security related criteria for selecting software from its sources?

TASK.DEV.03.04 Does the software supplier require its suppliers (third parties) to attest that their software complies with the organization's security requirements?

TASK.DEV.03.05 Does the software supplier require its suppliers (third parties) to supply provenance data and integrity verification mechanisms for all components of their software?

TASK.DEV.03.06 Does the software supplier have exception processes to address risk when its security requirements related to acquired third party software components are not met by that supplier/source?

TASK.DEV.03.07 For exceptions to security requirements, does the supplier have a process to periodically review all exceptions to requirements?

TASK.DEV.03.08 Does the supplier require a vulnerability disclosure program and/or product security incident response capabilities from its sources?

CONTROL.DEV.04 Does the software supplier create and maintain new roles and alter responsibilities for existing roles as needed to encompass all parts of the SDLC?

TASK.DEV.04.01 Does the software supplier integrate security roles into the software development team?

TASK.DEV.04.02 Does the software supplier have defined SDLC related roles and responsibilities for all members of the software development team?

TASK.DEV.04.03 Does the software supplier have defined roles responsibilities for other cybersecurity staff and points of contact associated with the SDLC (e.g., security champions, project managers and leads, senior management, software testers, software assurance leads and staff, product owners)?

TASK.DEV.04.04 Does the software supplier conduct an annual review of all roles and responsibilities?

TASK.DEV.04.05 Does the software supplier educate impacted individuals on impending changes to roles and responsibilities?

TASK.DEV.04.06 Does the software supplier use tools and processes to promote communication and engagement among individuals with SDLC related roles and responsibilities?

TASK.DEV.04.07 Does the software supplier designate a group of individuals or a team as the code owner for each project?

CONTROL.DEV.05 Does the software supplier provide periodically updated role based, SDLC related training for all personnel with responsibilities that contribute to secure architecture, development, testing and threat modeling?

TASK.DEV.05.01 Does the software supplier periodically review role based training and update the training as needed?

TASK.DEV.05.02 Does the software supplier periodically review personnel proficiency against their assigned role to determine whether additional training or training updates are needed?

TASK.DEV.05.03 Does the software supplier document the desired outcomes of training for each role?

TASK.DEV.05.04 Does the software supplier define the type of training or curriculum required to achieve the desired outcome for each role?

TASK.DEV.05.05 Does the software supplier acquire or create training for each role?

TASK.DEV.05.06 Does the software supplier measure outcome performance to identify areas where changes to training may be beneficial?

CONTROL.DEV.06 Does the software supplier obtain upper management or authorizing official commitment to secure development practices, and convey that commitment to all with SDLC related roles and responsibilities?

TASK.DEV.06.01 Does the software supplier appoint a single leader or leadership team to be responsible for the entire secure software development process?

TASK.DEV.06.02 Does the software supplier management (at all levels) incorporate secure development support into their communications with personnel (in particular, those with development-related roles and responsibilities)?

TASK.DEV.06.03 Does leadership of the secure software development process include accountability for releasing software to production and delegating responsibilities as appropriate?

TASK.DEV.06.04 Does the software supplier educate management personnel on the importance of secure development to the organization?

TASK.DEV.06.05 Does the software supplier educate all personnel with development related roles and responsibilities on upper management's commitment to secure development and the importance of secure development to the organization?

CONTROL.DEV.07 Does the software supplier define categories (e.g., IDE, compiler, build, and CI/CD tools) within toolchains, and specify the mandatory tools or tool types to be used for each category?

TASK.DEV.07.01 Does the software supplier identify security tools to integrate into the software developers' toolchain?

TASK.DEV.07.02 Does the software supplier provide information (e.g., settings) that can be used to rebuild the software?

TASK.DEV.07.03 Does the software supplier evaluate tools' capabilities to create immutable (signed) records/logs for auditability within the toolchain?

TASK.DEV.07.04 Does the software supplier use automated technology for toolchain management and orchestration?

CONTROL.DEV.08 Does the software supplier follow NIST recommended security practices to deploy, operate, and maintain tools and toolchains?

TASK.DEV.08.01 Does the software supplier include cybersecurity considerations or assessments in its evaluation, selection, and acquisition of its tools?

TASK.DEV.08.02 Does the software supplier use code-based configuration for development toolchains (e.g., pipelines-as-code, toolchains-as-code)?

TASK.DEV.08.03 Does the software supplier implement software development technologies and processes needed for reproducible builds?

TASK.DEV.08.04 Does the software supplier update, upgrade, or replace software development tools as needed to address tool vulnerabilities or add new tool capabilities?

TASK.DEV.08.05 Does the software supplier continuously monitor tools and tool logs for potential operational and security issues, including policy violations and anomalous behavior?

CONTROL.DEV.09 Does the software supplier configure tools to generate artifacts of their compliance with secure software development practices as defined by the organization?

TASK.DEV.09.01 Does the software supplier use automated workflow tooling (e.g., workflow tracking, issue tracking, value stream mapping) to create an audit trail of the secure development-related actions that are performed for continuous improvement purposes?

TASK.DEV.09.02 Does the software supplier determine how often the collected information should be audited, and implement the necessary processes?

TASK.DEV.09.03 Does the software supplier establish and enforce security and retention policies for software development artifact data?

TASK.DEV.09.04 Does the software supplier assign responsibility for creating and managing any needed artifacts that tools cannot generate?

TASK.DEV.09.05 Does the supplier encrypt build-related artifacts at rest and in transit?

CONTROL.DEV.10 Does the software supplier define, gather, and use software security criteria and measures throughout the SDLC?

TASK.DEV.10.01 Does the software supplier ensure that the software development security criteria adequately indicate how effectively security risk is being managed?

TASK.DEV.10.02 Does the software supplier define key performance indicators (KPIs), key risk indicators (KRIs), vulnerability severity scores, and other measures for software security?

TASK.DEV.10.03 Does the software supplier review the artifacts generated as part of the software development workflow system to determine if they meet the criteria?

TASK.DEV.10.04 Does the software supplier use the software development toolchain to automatically gather information that informs security decision-making?

TASK.DEV.10.05 Does the software supplier deploy additional tools if needed to support the generation and collection of information supporting the criteria?

TASK.DEV.10.06 Does the software supplier automate decision-making processes using the security criteria, and periodically review these processes?

TASK.DEV.10.07 Does the software supplier only allow authorized personnel to access the gathered information?

TASK.DEV.10.08 Does the software supplier prevent any alteration or deletion of the gathered information?

CONTROL.DEV.11 Does the software supplier separate and protect each environment used in a phase of software development (build system, staging, and production)?

TASK.DEV.11.01 Does the software supplier use multi-factor, risk-based authentication, and conditional access for each environment?

TASK.DEV.11.02 Does the software supplier use network segmentation and access controls consistent with zero-trust principles to separate each environment from others and from production environments, and to separate components from each other within each non-production environment?

TASK.DEV.11.03 Does the software supplier enforce authentication and tightly restrict connections entering and exiting each software development environment, including minimizing access to the internet to only what is necessary?

TASK.DEV.11.04 Does the software supplier minimize direct human access to toolchain systems such as build services?

TASK.DEV.11.05 Does the software supplier continuously monitor and audit all access attempts and all use of privileged access?

TASK.DEV.11.06 Does the software supplier isolate the use of production environment software and services from non-production environments?

TASK.DEV.11.07 Does the software supplier regularly log, monitor, and audit trust relationships for authorization and access between the environments and between the components within each environment?

TASK.DEV.11.08 Does the software supplier continuously log and monitor operations and alerts across all components of the development environment to detect, respond, and recover from attempted and actual cyber incidents?

TASK.DEV.11.09 Does the software supplier configure security controls and other tools involved in separating and protecting the environments to generate artifacts for their activities?

TASK.DEV.11.10 Does the software supplier continuously monitor all software deployed in each environment for new vulnerabilities?

TASK.DEV.11.11 Does the software supplier follow a risk-based approach to respond to vulnerabilities?

CONTROL.DEV.12 Does the software supplier secure its development endpoints (i.e., endpoints for software designers, developers, testers, builders, etc.) to perform development-related tasks using a risk-based approach?

TASK.DEV.12.01 Does the software supplier configure each development endpoint based on approved hardening guides, checklists, etc.?

TASK.DEV.12.02 Does the software supplier configure each development endpoint and the development resources to provide the least functionality needed by users and services and to enforce the principle of least privilege?

TASK.DEV.12.03 Does the software supplier continuously monitor the security posture of all development endpoints, including monitoring and auditing all use of privileged access?

TASK.DEV.12.04 Does the software supplier configure security controls and other tools involved in securing and hardening development endpoints to generate artifacts for their activities?

TASK.DEV.12.05 Does the software supplier require multi-factor authentication (MFA) for all access to development endpoints and development resources?

TASK.DEV.12.06 Does the software supplier provide dedicated development endpoints on non-production networks for performing all development-related tasks?

TASK.DEV.12.07 Does the software supplier also provide separate endpoints on production networks for non-development related tasks, such as system administration tasks?

TASK.DEV.12.08 Does the software supplier follow a zero-trust architecture to configure each development endpoint?

CONTROL.DEV.13 Does the software supplier have defined policies for securely storing all forms of code (including source code, executable code, and configuration-as-code), release artifacts, and associated integrity verification information for each release?

TASK.DEV.13.01 Does the software supplier store all source code and configuration-as-code in a version controlled code repository to track all changes?

TASK.DEV.13.02 Does the software supplier use commit signing for code repositories?

TASK.DEV.13.03 Does the software supplier have the code owner review and approve all changes made to the code by others?

TASK.DEV.13.04 Does the software supplier use code signing to help protect the integrity of executables?

TASK.DEV.13.05 Does the software supplier use cryptography (e.g., cryptographic hashes) to help protect file integrity?

TASK.DEV.13.06 Does the software supplier securely store the necessary files and supporting data (e.g., integrity verification information) to be retained for each software release (e.g., by keeping it in a separate location from the release files or by signing the data)?

TASK.DEV.13.07 Does the software supplier store all forms of code and associated integrity verification information for each release based on the principle of least privilege (including read-only access for release files) so that only authorized personnel, tools, services, etc. have access?

TASK.DEV.13.08 Does the supplier automatically revoke access to development and/or release repositories when an authorized user is no longer an active member of the development team associated with the repository?

TASK.DEV.13.09 Does the software supplier securely archive the necessary files and supporting data (e.g., integrity verification information) to be retained for each software release?

TASK.DEV.13.10 Does the supplier periodically review source code and integrity verification information access logs for attempted access and compare such access to the list of authorized accounts?

CONTROL.DEV.14 Does the software supplier make software authenticity and integrity verification information available to software acquirers?

TASK.DEV.14.01 Does the software supplier post cryptographic hashes for release files on a well-secured website?

TASK.DEV.14.02 Does the software supplier use an established certificate authority or trusted signing keys for code signing so that consumers' operating systems or other tools and services can confirm the validity of signatures before use?

TASK.DEV.14.03 Does the software supplier periodically review the code signing processes, including signing keys renewal, rotation, revocation, and protection?

CONTROL.DEV.15 Does the software supplier use forms of risk modeling—such as threat modeling, attack modeling, or attack surface mapping—to help assess the security risk for the software?

TASK.DEV.15.01 Does the software supplier train the development team how to use a risk-based approach to communicate the risks and determine how to address them, including implementing mitigations?

TASK.DEV.15.02 Does the software supplier apply additional rigor in performing assessments for high-risk areas, such as protecting sensitive data and safeguarding identification, authentication, and access control including credential management?

TASK.DEV.15.03 Does the software supplier review vulnerability reports and statistics for previous software versions to inform the security risk assessment?

TASK.DEV.15.04 Does the software supplier use data classification methods to identify and characterize each type of data that the software will interact with?

CONTROL.DEV.16 Does the software supplier track and maintain the software's security requirements, risks, and design decisions?

TASK.DEV.16.01 Does the software supplier record the response to each identified risk, including mitigations performed, the rationale for any approved exceptions to the security requirements, and any mitigation additions to the software's security requirements?

TASK.DEV.16.02 Does the software supplier maintain records of design decisions, risk responses, and approved exceptions that can be used for auditing and maintenance purposes throughout the rest of the software life cycle?

TASK.DEV.16.03 Does the software supplier periodically re-evaluate all approved exceptions to the security requirements and implement changes as needed?

CONTROL.DEV.17 Does the software supplier take advantage of modules providing standardized implementations of security features and services where appropriate instead of creating customized implementations of security features and services?

TASK.DEV.17.01 Does the software supplier maintain one or more software repositories of modules for supporting standardized security features and services?

TASK.DEV.17.02 Does the software supplier determine secure configurations for modules for supporting standardized security features and services, and make these configurations available (e.g., as configuration-as-code) so developers can readily use them?

TASK.DEV.17.03 Does the software supplier define criteria for which security features and services must be supported by software to be developed?

CONTROL.DEV.18 Does the software supplier have qualified personnel and/or automated processes to review alignment between the software design and security requirements?

TASK.DEV.18.01 Does the software supplier review the software design to confirm that it addresses applicable security requirements?

TASK.DEV.18.02 Does the software supplier review the risk models created during software design to determine if they appear to adequately identify the risks?

TASK.DEV.18.03 Does the software supplier review the software design to confirm that it satisfactorily addresses the risks identified by the risk models?

TASK.DEV.18.04 Does the software supplier have the software architect correct failures to meet the requirements?

TASK.DEV.18.05 Does the software supplier change the design and/or the risk response strategy if the security requirements cannot be met?

TASK.DEV.18.06 Does the software supplier record the findings of design reviews to serve as artifacts?

CONTROL.DEV.19 Does the software supplier create, maintain, and reuse well-secured software components developed in-house following SDLC processes to meet shared internal software development?

TASK.DEV.19.01 Does the software supplier follow organization-established security practices for secure software development when creating and maintaining software components?

TASK.DEV.19.02 Does the software supplier maintain one or more software repositories for these components?

TASK.DEV.19.03 Does the software supplier promote the use of preexisting and vetted software components?

TASK.DEV.19.04 Does the software supplier maintain vetted software components?

TASK.DEV.19.05 Does the software supplier promote the creation of reusable components by their internal software development teams?

CONTROL.DEV.20 Does the software supplier verify that third-party software components, including open source and commercial components, comply with the requirements, as defined by the supplier for the software, throughout the lifecycle of the component in the application?

TASK.DEV.20.01 Does the supplier review component patches and updates for functional changes that might impact runtime requirements for the software?

TASK.DEV.20.02 If a component patch or update implements changes to security functions, does the supplier perform a threat analysis to determine if those changes impact other security functions or the ability to perform forensic analysis on the software?

TASK.DEV.20.03 Does the supplier define a policy handling end-of-life and end-of-support conditions for third-party components?

CONTROL.DEV.21 Prior to component usage, does the software supplier check whether there are publicly known vulnerabilities in the software modules and services that they (or their component sources) have not yet fixed?

TASK.DEV.21.01 Does the supplier identify where their software suppliers publish vulnerability information?

TASK.DEV.21.02 If a component or module supplier publishes their vulnerability information in a location other than the NVD, does the supplier have a process to automatically process that vulnerability data source?

TASK.DEV.21.03 If a software service provider discloses a vulnerability, does the supplier perform a risk-based review of all points of usage for that software supplier to determine the impact of the vulnerability on the software?

CONTROL.DEV.22 Does the software supplier build into the toolchain automatic detection of known vulnerabilities in software components?

TASK.DEV.22.01 Does the software supplier use existing results from commercial services for vetting the software modules and services?

TASK.DEV.22.02 Does the software supplier ensure that each software component is still actively maintained and has not reached end-of-life (this should include ensuring no new vulnerabilities have been found in the software being remediated)?

TASK.DEV.22.03 Does the software supplier determine a plan of action for each software component that is no longer being maintained or that will not be available in the near future?

TASK.DEV.22.04 Does the software supplier confirm the integrity of software components through digital signatures or other mechanisms?

TASK.DEV.22.05 Does the software supplier review, analyze, and/or test code?

CONTROL.DEV.23 Does the software supplier follow all secure coding practices that are appropriate to the development languages and environment to meet the organization’s requirements?

TASK.DEV.23.01 Does the software supplier follow secure coding practices to validate all inputs?

TASK.DEV.23.02 Does the software supplier validate and properly encode all outputs?

TASK.DEV.23.03 Does the software supplier follow secure coding practices to avoid using known unsafe functions and calls?

TASK.DEV.23.04 Does the software supplier follow secure coding practices to detect and handle errors?

TASK.DEV.23.05 Does the software supplier follow secure coding practices to provide logging and tracing capabilities?

TASK.DEV.23.06 Does the software supplier use development environments with automated features that encourage or require the use of secure coding practices?

TASK.DEV.23.07 Does the software supplier employ just-in-time training in its secure coding practices?

TASK.DEV.23.08 Does the software supplier follow procedures for manually ensuring compliance with secure coding practices when automated methods are insufficient or unavailable?

TASK.DEV.23.09 Does the software supplier use tools (e.g., linters, formatters) to standardize the style and formatting of the source code?

TASK.DEV.23.10 Does the software supplier check for other vulnerabilities that are common to the development languages and environment it uses?

TASK.DEV.23.11 Does the software supplier have the developer review their own human-readable code to complement (not replace) code review performed by other people or tools?

CONTROL.DEV.24 Does the software supplier use compiler, interpreter, and build tools that offer features to improve executable security?

TASK.DEV.24.01 Does the software supplier use up-to-date versions of compiler, interpreter, and build tools?

TASK.DEV.24.02 Does the software supplier follow change management processes when deploying or updating compiler, interpreter, and build tools and audit all unexpected changes to tools?

TASK.DEV.24.03 Does the software supplier regularly validate the authenticity and integrity of compiler, interpreter, and build tools?

CONTROL.DEV.25 Does the software supplier determine which compiler, interpreter, and build tool features should be used and how each should be configured, then implement and use the approved configurations?

TASK.DEV.25.01 Does the software supplier determine and implement optimum compiler, interpreter, and build tool feature configurations for the build tools based on its risk strategy and security policies?

TASK.DEV.25.02 Does the software supplier enable compiler features that produce warnings for poorly secured code during the compilation process?

TASK.DEV.25.03 Does the software supplier implement the “clean build” concept, where all compiler warnings are treated as errors and eliminated except those clearly determined to be false positives or irrelevant?

TASK.DEV.25.04 Does the software supplier perform all builds in a dedicated, highly controlled build environment?

TASK.DEV.25.05 Does the software supplier enable compiler features that randomize or obfuscate execution characteristics (such as memory location usage) that would otherwise be predictable and thus potentially exploitable?

TASK.DEV.25.06 Does the software supplier test to ensure that the features are working as expected and are not inadvertently causing any operational issues or other problems?

TASK.DEV.25.07 Does the software supplier make the approved tool configurations available as configuration-as-code so developers can readily use them?

TASK.DEV.25.08 Does the software supplier continuously verify that the approved configurations are being used?

CONTROL.DEV.26 Does the software supplier have and follow established policies and procedures to determine when and how to perform code reviews, code analysis, and/or testing methodologies?

TASK.DEV.26.01 Does the software supplier mandate the organization’s policies or guidelines for when code review (human looking at code) vs. code analysis (using tools to find coding issues) vs. testing should be performed, and how it should be conducted for all code whether developed in-house or via third party?

TASK.DEV.26.02 Does the software supplier prescribe code review, analysis methods, and/or testing methods based on the stage of the software?

TASK.DEV.26.03 Does the software supplier perform the code review code analysis and/or testing based on the organization’s secure coding standards?

TASK.DEV.26.04 Does the software supplier perform peer review of code, and review any existing code review, analysis, or testing results as part of the peer review?

TASK.DEV.26.05 Does the software supplier use peer reviewing tools that facilitate the peer review process, and document all discussions and feedback?

TASK.DEV.26.06 Does the software supplier use a static analysis tool to automatically check code for vulnerabilities and compliance with the organization’s secure coding standards with a human reviewing the issues reported by the tool?

TASK.DEV.26.07 Does the software supplier use review checklists to verify that the code complies with the requirements?

TASK.DEV.26.08 Does the software supplier use expert reviewers to check code for backdoors and other malicious content?

TASK.DEV.26.09 Does the software supplier use automated tools to identify and verify unsafe software practices on a continuous basis as human-readable code is checked into the code repository?

TASK.DEV.26.10 Does the software supplier identify and document the root causes of discovered issues?

TASK.DEV.26.11 Does the software supplier record, triage, and address all discovered issues and recommended remediations in the development team's workflow or issue tracking system?

TASK.DEV.26.12 Does the software supplier follow up to capture metrics on defect error rates, remediation time, and resolution types to ensure defect patterns are identified and issues are properly resolved?

TASK.DEV.26.13 Does the software supplier document lessons learned from code review and analysis and make the lessons available to developers?

CONTROL.DEV.27 Does the software supplier determine whether executable code testing should be performed to find vulnerabilities not identified by previous reviews, analysis, or testing?

CONTROL.DEV.28 Does the software supplier have a rigorous testing process that includes functional and dynamic testing, documenting of test results to include types of testing, recording, and triaging all discovered issues in a workflow or issue tracking system?

TASK.DEV.28.01 Does the supplier's testers follow the organization's policies or guidelines for when code testing should be performed and how it should be conducted (e.g., within a sandboxed environment).

TASK.DEV.28.02 Does the software supplier ensure that its policies and guidelines for code testing are followed for third-party executable code and reusable executable code modules written in-house?

TASK.DEV.28.03 Does the software supplier perform robust functional testing of security features?

TASK.DEV.28.04 Does the software supplier integrate dynamic vulnerability testing into the project's automated test suite?

TASK.DEV.28.05 Does the software supplier incorporate tests for previously reported vulnerabilities into the project's test suite to ensure that errors are not reintroduced?

TASK.DEV.28.06 Does the software supplier take into consideration the infrastructures and technology stacks the software will be used with when developing test plans in production?

TASK.DEV.28.07 Does the software supplier use fuzz testing tools to find issues with input handling?

TASK.DEV.28.08 Does the software supplier review, analyze, and/or test the software's code to identify or confirm the presence of previously undetected vulnerabilities?

TASK.DEV.28.09 Does the software supplier configure the toolchain to perform automated code analysis and testing on a regular or continuous basis for all supported releases?

TASK.DEV.28.10 Does the software supplier use penetration testing to simulate how an attacker might attempt to compromise the software in high-risk scenarios?

TASK.DEV.28.11 Does the software supplier identify and record the root causes of discovered issues?

TASK.DEV.28.12 Does the software supplier document lessons learned from code testing with lessons provided to developers?

TASK.DEV.28.13 Does the software supplier use source code, design records, and other resources when developing test plans?

TASK.DEV.28.14 Does the software supplier conduct testing to ensure that the settings, including the default settings, are working as expected and are not inadvertently causing any security weaknesses, operational issues, or other problems?

CONTROL.DEV.29 Does the software supplier implement the default settings (or groups of default settings, if applicable), and document each setting for software administrators?

TASK.DEV.29.01 Does the software supplier verify that the approved configuration is in place for the software?

TASK.DEV.29.02 Does the software supplier document each setting's purpose, options, default value, security relevance, potential operational impact, and relationships with other settings?

TASK.DEV.29.03 Does the software supplier use authoritative programmatic technical mechanisms to record how each setting can be implemented and assessed by software administrators?

TASK.DEV.29.04 Does the software supplier store the default configuration in a usable format? If so, does the software supplier follow change control practices for modifying it (e.g., configuration-as-code)?

CONTROL.DEV.30 Does the software supplier use a vulnerability disclosure program that gathers information on potential vulnerabilities in the software and its third-party components?

TASK.DEV.30.01 Does the software supplier monitor vulnerability databases, security mailing lists, and other sources of vulnerability reports through manual or automated means?

TASK.DEV.30.02 Does the software supplier use threat intelligence sources to better understand how vulnerabilities in general are being exploited?

TASK.DEV.30.03 Does the software supplier investigate all credible vulnerability reports?

TASK.DEV.30.04 Does the software supplier automatically review provenance and software composition data for all software components to identify any new vulnerabilities they have?

SECURE SOFTWARE DEPLOYMENT CONTROLS

This section of the guide focuses on those aspects of the software supply chain that relate to secure deployment practices implemented by software suppliers and software consumers, such as federal agencies. Responses are provided by parties with ownership for risk management activities pertaining to the secure deployment of software within their respective environments and are intended to ensure that best practices for secure deployment of software are being applied.

DEFINITION OF CONTEXTUAL TERMS

For the purposes of this document the following consensus terms apply:

Deployment means, “the act of putting a software product into operation, making it available for use within a digital ecosystem.”

Secure Deployment means, “the application of processes and procedures that aim to identify and mitigate software risk prior to, and after, deployment of software within a digital ecosystem.”

Operations means, “the act of running or operating a piece of software, whether that software is part of a physical device, such as firmware, exists as a mobile application, is part of a server or cloud-based application, or is a desktop application.”

DEPLOYMENT CONTROL AND TASK QUESTIONS

CONTROL.DEP.01 Has the software supplier implemented multi-factor authentication (MFA) for all login access to accounts for both local and remote access?

TASK.DEP.01.01 Do all interactive administrative and privileged functions accessible by a user within the product require MFA?

TASK.DEP.01.02 Is MFA required for local account access?

TASK.DEP.01.03 Is MFA required for all user accounts by default?

TASK.DEP.01.04 If the product supports application program interface (API) access, is MFA required for any user to generate an API access token?

TASK.DEP.01.05 Does the supplier employ toolsets that are verifier impersonation-resistant for authenticating login access?

CONTROL.DEP.02 Does software supplier's organization maintain updated cybersecurity incident response plans related to operating the software and development of the software?

TASK.DEP.02.01 Does supplier provide details of the cybersecurity incident response communication process for its customers?

TASK.DEP.02.02 Does the supplier provide details of patch or other mitigation steps required as part of a cybersecurity incident or vulnerability response process?

TASK.DEP.02.03 Does the supplier communicate the occurrence of a cybersecurity incident to its customers?

TASK.DEP.02.04 Does the supplier have a Business Continuity Plan that is regularly exercised?

TASK.DEP.02.05 For SaaS or cloud-based services, does the supplier provide written SLAs related to restoration of service because of an incident?

TASK.DEP.02.06 For SaaS or cloud-based services, are all backups of the software supplier's systems that are necessary for operations regularly checked/exercised for restoration?

CONTROL.DEP.03 Does the software supplier follow best practices for log management as defined in NIST SP800-92 or OMB M-21-31?

TASK.DEP.03.01 Does the product support centralized logging?

TASK.DEP.03.02 Does the product generate log files in a consistent manner supporting automated analysis for unexpected events?

TASK.DEP.03.03 Does the product support policy-based configuration of log content?

TASK.DEP.03.04 If the product is deployed in a distributed manner, does the product synchronize timestamps between components of the product?

TASK.DEP.03.05 If the software supplier provides a SaaS solution, does the software supplier meet Event Logging Tier 1 (EL1) for low assurance environments?

TASK.DEP.03.06 If the software supplier provides a SaaS solution, does the supplier have a plan to meet higher logging tier levels consistent with the timelines outlined in M-21-31?

TASK.DEP.03.07 If the software supplier provides a SaaS solution, is the software supplier monitoring log activity on a regular basis for anomalous or suspicious events?

TASK.DEP.03.08 If the software supplier provides a SaaS solution, does the supplier retain log information consistent with the retention timelines outlined in M-21-31?

CONTROL.DEP.04 Does the supplier have a defined patch process for all software, tools, and systems used in the delivery of the software?

TASK.DEP.04.01 Does the patch process focus on response to vulnerabilities in a timely manner to reduce the cyberattack susceptibility window timeframe?

TASK.DEP.04.02 Does the software supplier maintain a patch history for all software, tools, and systems used in the delivery of the software?

TASK.DEP.04.03 Does the software supplier maintain a log of when patches are applied to all software, tools, and systems used in the delivery of the software?

TASK.DEP.04.04 If the supplier provides a SaaS solution, does the software supplier provide a documented SLA for patch application?

CONTROL.DEP.05 Does the supplier require its suppliers to have a process in place, such as scanning, to address undocumented or obsolete code or functions that might allow unauthorized access or use of the software product, or cause the software product to behave outside of the specified requirements or in an unreliable manner?

TASK.DEP.05.01 Does the software supplier require sub-suppliers to have a process to account for hidden functions and vulnerable features embedded in the code, describing their purpose and their impact on the integrity and reliability of software product?

TASK.DEP.05.02 Does the software supplier require sub-suppliers to have hidden functions removed or (as a minimum as part of security hardening procedures) addressed (e.g., as part of the failure modes and effects analysis of the software) to prevent any unauthorized access or degradation of the reliability of the software product?

CONTROL.DEP.06 Does the supplier have protections in place between software supplier's network and cloud service providers, including protections associated with contractually specified government protection levels and personnel background checks if applicable?

TASK.DEP.06.01 If the supplier has connectivity to a government system requiring special protection levels, does the supplier have established policies in place requiring background checks and screening for personnel and requisite information handling procedures (including clearance levels and formal access approvals) appropriate to the protection level associated with the interconnected government system?

TASK.DEP.06.02 Does the supplier enforce the screening and formal access approval process for its personnel and sub-contractors (including service providers) that have access to the government system requiring special protection levels?

TASK.DEP.06.03 Does the software supplier convey cloud security requirements to sub-suppliers and sub-contractors?

CONTROL.DEP.07 Does the supplier's product support hardened security configurations that enforce the principles of least privilege, separation of duties, and least functionality?

TASK.DEP.07.01 Is hardened security configuration implemented as default, out of box behavior in the software supplier's product?

TASK.DEP.07.02 Does the software supplier provide accessible documentation on hardening the security configurations?

CONTROL.DEP.08 Does the supplier have methods to verify the trust relationship between a product supplier and the party identified within a digital signature for a product installation package?

TASK.DEP.08.01 Does every party in the supply chain ensure that software received from their suppliers is validated, trusted, and authorized by an electronic certificate?

TASK.DEP.08.02 Does the supplier verify that the certification used to sign any software from each supplier is authorized by their supplier?

TASK.DEP.08.03 Does the supplier provide a means to verify that their certificate was used to sign this software?

TASK.DEP.08.04 Does the software supplier restrict access to code signing certificates?

TASK.DEP.08.05 Does the supplier audit the use of their code signing certificates?

CONTROL.DEP.09 Do the binary packages distributed by the software supplier implement cryptographic signatures to ensure they are not manipulated or tainted?

TASK.DEP.09.01 Do the signatures also provide means to assure authenticity and ownership of the publisher for each individual component in the distribution?

TASK.DEP.09.02 Do the binary packages distributed by the software supplier implement verification of the integrity of the individual components in the distribution (e.g., Hash files representing checksum, cryptographically signed packages, and individual components included in the distribution and listed in the SBOM)?

TASK.DEP.09.03 Do software updates and upgrades verify the authenticity and ownership of the individual components included in the distribution through cryptographic signatures prior to installing the binary files (e.g., Trusted boot, Verified boot, or Secure boot)?

TASK.DEP.09.04 Are the binary packages distributed by the software supplier cryptographically signed using a valid public certificate authority for code signing so that consumers' operating systems or other tools and services can confirm the validity of signatures before use?

CONTROL.DEP.10 Does the software supplier certify their software against applicable standards and maintain that certification?

TASK.DEP.10.01 If the software supplier provides a SaaS solution, is the supplier's service examined by an accredited third-party certified to audit data protection and cybersecurity controls such as against the Association of International Certified Professional Accountants SOC 2 Trust Services Criteria for service organizations?

TASK.DEP.10.02 Is the software provided by the supplier certified against Common Criteria Protection Profile Tracks?

TASK.DEP.10.03 Is the software provided by the supplier certified against any U.S. federal government approved products list?

TASK.DEP.10.04 Does the software supplier follow an industry standard or framework such as NIST Risk Management Framework (RMF) for supplier's internal or third-party cloud deployments, as applicable?

TASK.DEP.10.05 Does the software supplier use a third-party auditing function or certifier?

CONTROL.DEP.11 Does the supplier provide detailed deployment guidance for the software?

TASK.DEP.11.01 Does the provided deployment guidance, and any associated deployment scripts or installers, follow secure by default principles?

TASK.DEP.11.02 Does the supplier document the expected network configuration, including any requirements for network isolation?

TASK.DEP.11.03 If the software is deployable in a distributed manner, including via the use of containers, are interdependencies between components in distributed systems documented?

TASK.DEP.11.04 If the software requires the use of a service provider, are service provider configuration requirements documented?

TASK.DEP.11.05 If the software requires the use of user downloaded components (e.g., a mobile application), are the deployment requirements for the product documented, including any required security configuration of the user device?

CONTROL.DEP.12 If the software requires the use of a service provider, does the supplier perform any pre-installation integrity or validation checks to ensure that the software continues to meet deployment or acceptance criteria associated with an authority to operate?

TASK.DEP.12.01 Does the supplier require vulnerability scanning of all software prior to installation?

TASK.DEP.12.02 Does the supplier have a process in place to identify and mitigate any inadvertent or inappropriate alterations of the software package that is delivered to consumers for installation?

TASK.DEP.12.03 Does the supplier perform a pre-installation review of configuration requirements to ensure that any deployment dependencies meet deployment or acceptance criteria associated with an authority to operate?

VULNERABILITY MANAGEMENT CONTROLS

Software suppliers are expected to monitor their software products for vulnerabilities following "NIST Guidance" as specified in OMB M-22-18, coordinated vulnerability disclosure standards and processes, and other on-going methods. Software suppliers issue security advisories to identify the list of products in their product catalog that are affected whenever a new vulnerability is reported. Customers use these security advisories to evaluate risk within their ecosystems and take mitigating action based on guidance provided in the security advisory. Ideally, a software supplier will provide their security advisories in both human-readable and machine-readable forms.

CONTEXTUAL CONSIDERATIONS

The OASIS Common Security Advisory Framework (CSAF) has gained traction as an international, open standard for producing, distributing, and discovering machine-readable security advisories. A software supplier issues a security advisory whenever a new vulnerability is reported, making this a "vulnerability centric" artifact.

Vulnerability Exploitability eXchange (VEX) allows a supplier to assert whether specific vulnerabilities affect their product. A VEX document is a form of a security advisory that indicates whether a product or products are affected by a known vulnerability or vulnerabilities. A VEX advisory can also indicate that a product is not affected by a vulnerability. Not all vulnerabilities are exploitable and put an organization at risk. To help reduce effort spent investigating vulnerabilities, suppliers can issue a VEX advisory that states whether a product is or is not affected by a specific vulnerability in a machine-readable, automated way. VEX can be implemented in CSAF, CycloneDX, and SPDX; and OpenVEX has implementations as well. VEX data can also support more effective use of SBOM data, as VEX advisories support linking to an SBOM and specific SBOM components. While SBOM gives suppliers information on where they are potentially at risk, a VEX document helps an organization find out where they are affected by known vulnerabilities, if at all, and if actions need to be taken to remediate based on exploitation status (affected, not affected, fixed, under investigation).

NIST SP 800-161r1 RA-5 and "NIST Guidance" as specified in OMB M-22-18, call for the issuance of vulnerability disclosure reports (VDR) as an attestation showing that a software supplier has checked each component in a software product SBOM for vulnerabilities and reports the status of each vulnerability discovered, following recommendations for coordinate vulnerability disclosure programs contained in IEC 29147:2018. A VDR is a machine-readable artifact and is considered a "product centric" artifact. A VDR is issued simultaneously with a SBOM at product release and remains online as a living document. It is updated by the software supplier when new vulnerabilities affect the product to which the VDR is attesting. This enables a consumer to know if the software product is affected when a new vulnerability is reported.

VULNERABILITY MANAGEMENT CONTROL AND TASK QUESTIONS

CONTROL.VULN.01 Does the supplier provide a NIST defined vulnerability disclosure report for the current version of the product and all future versions, including updates?

TASK.VULN.01.01 Are software operators expected to directly update individual components from suppliers providing patches for the identified vulnerability or update the product directly from the supplier?

TASK.VULN.01.02 Does the supplier provide mitigation guidance following "NIST Guidance" per OMB memo M-22-18?

TASK.VULN.01.03 Does the mitigation guidance allow software operators to effectively remove or reduce the vulnerability risk?

TASK.VULN.01.04 Does the supplier include descriptions of mitigations performed by the supplier to address vulnerabilities?

TASK.VULN.01.05 Does the supplier have established remediation time thresholds for levels of severity/criticality in software vulnerabilities?

TASK.VULN.01.06 Do acceptance teams implement a trust but verify model where the product is analyzed by a software composition analysis or risk assessment tool capable of performing a binary analysis?

CONTROL.VULN.02 Does the supplier scan for and mitigate potential vulnerabilities at the component level within the product?

TASK.VULN.02.01 Has the supplier ensured these processes operate on an ongoing basis and, at a minimum, prior to product, version, or update releases?

TASK.VULN.02.02 Has the supplier remediated or mitigated material security relevant vulnerabilities prior to product release?

CONTROL.VULN.03 Does the software supplier provide a notification process for vulnerability disclosures?

TASK.VULN.03.01 Does vulnerability notification occur in advance of patch availability or only upon release of a patch?

TASK.VULN.03.02 Does vulnerability notification include mitigation options?

TASK.VULN.03.03 Does vulnerability notification include root cause and/or impact analysis via a Common Weakness Enumeration (CWE)?

TASK.VULN.03.04 Does the supplier provide both a human-readable and machine-readable security advisory, such as CSAF formatted Security Advisory (profile 4)?

CONTROL.VULN.04 Does the supplier have documented policies or procedures for internal identification and management of vulnerabilities within their networks and enterprise systems?

TASK.VULN.04.01 Are automated mechanisms employed to detect and notify authorized personnel of the presence of unauthorized software on networks and enterprise systems?

TASK.VULN.04.02 Are automated mechanisms employed to compare the results of vulnerability scans over time to determine trends in networks and enterprise systems vulnerabilities and mitigation/flaw remediation activities?

TASK.VULN.04.03 Does the supplier discern and document what information associated with the networks and enterprise systems is discoverable publicly over the internet?

TASK.VULN.04.04 Does the supplier employ vulnerability scanning tools and techniques that promote interoperability among tools and vulnerability management automation?

TASK.VULN.04.05 Does the supplier have established remediation time thresholds for levels of severity/criticality in networks and enterprise systems?

CONTROL.VULN.05 Does the supplier employ vulnerability scanning procedures that maximize the breadth and depth of coverage within their own digital ecosystem, especially software development and build environments (i.e., networks and enterprise system components scanned, and vulnerabilities checked)?

TASK.VULN.05.01 Does the supplier analyze vulnerability scan reports regularly (at least weekly)?

TASK.VULN.05.02 Does the supplier employ vulnerability scanning tools that include the capability to update the list of cyber vulnerabilities scanned?

TASK.VULN.05.03 Does the supplier update the list of vulnerabilities scanned regularly (at least weekly) and when new vulnerabilities are identified and reported?

TASK.VULN.05.04 Does the supplier include privileged access authorization to networks and enterprise systems for selected vulnerability scanning activities to facilitate more thorough scanning?

TASK.VULN.05.05 Does the supplier perform security testing to determine the level of difficulty in circumventing the security controls of the networks and enterprise systems?
Note: Testing methods include penetration testing, malicious user testing, and independent verification and validation.

CONTROL.VULN.06 Does the software supplier have a policy and program to monitor for vulnerabilities within the supplier's running ecosystem and remediate such security-relevant vulnerabilities?

TASK.VULN.06.01 Has the supplier ensured these processes operate on an ongoing basis and, at a minimum, prior to product, version, or update releases?

TASK.VULN.06.02 Has the supplier remediated or mitigated material security relevant vulnerabilities prior to product release?

CONTROL.VULN.07 Does the supplier implement industry standards or frameworks for vulnerability management?

TASK.VULN.07.01 Does the software supplier have an intrusion detection and monitoring system in place to detect unauthorized activities?

TASK.VULN.07.02 Does software supplier’s organization maintain updated indicators of compromise?

TASK.VULN.07.03 Does the supplier's organization scan for vulnerabilities in externally obtained software (e.g., pen testing of enterprise and non-enterprise software)?

TASK.VULN.07.04 Does the supplier conduct threat hunting exercises and pen testing on a reasonable cadence?

CONTROL.VULN.08 Does the supplier implement a documented process to resolve and disclose identified vulnerabilities in a software product?

TASK.VULN.08.01 Does the supplier have a reasonable policy and process to disclose security-relevant vulnerabilities?

TASK.VULN.08.02 Does the software supplier have processes to ensure sub-suppliers disclose vulnerabilities?

APPENDIX A

GLOSSARY OF KEY TERMS

The following terms are used in this document and understanding our definition for each will be beneficial. Where practical, the authors have attempted to use pre-existing definitions. Where that was not possible, we have adapted an existing definition and cited the source material. In the event that a term is used, but is not included in this glossary, the authors intend other terms to align with definitions present in NIST SP 800-161-r1.

APPLICATION

A system for collecting, saving, processing, and presenting data by means of a computer [ISO19770-2:4.1.1]. The term application is generally used when referring to a component of software that can be executed. The terms application and software application are often used synonymously. (Source: NISTIR 7695)

ARTIFICIAL INTELLIGENCE (AI)

A branch of computer science devoted to developing data processing systems that performs functions normally associated with human intelligence, such as reasoning, learning, and self-improvement. While a distinct branch of computer science, AI is related to, but distinct from, machine learning (see separate machine learning definition). (Source: Modified from ANSI INCITS 172-220 [R2007])

APPLICATION PROGRAM INTERFACE (API)

A set of definitions and protocols for building and integrating application software. It acts as an intermediary layer that processes data transfers between systems, letting companies open their application data and functionality to external third-party developers, business partners, and internal departments within their companies. (Source: Adapted from RedHat and IBM)

C-SCRM

A systematic process for managing exposure to cybersecurity risks throughout the supply chain and developing appropriate response strategies, policies, processes, and procedures. (Source: NIST SP 800-161r1)

DEVELOPMENT ENDPOINT

A compute device used in the creation of software by a software designer, tester, builder, etc. (Source: ICT SCRM Task Force Software Assurance Working Group)

ENVIRONMENT

Aggregate of external procedures, conditions, and objects affecting the development, operation, and sustainment of software. (Source: Derived from FIPS 200)

CERTIFIED

A seal of approval from a third-party body that a supplier produces software following a published standard that meets the requirements of that standard. (Source: Adapted from ISO).

CLOUD GENERATED

Data generated from a cloud computing-based service, independent of the deployment model for the cloud computing service. For the purposes of this document, such data could be source code, binary artifacts, deployment recommendations, or security testing results that impact the release criteria for the software under evaluation. (Source: Adapted from NIST SP800-145)

COMMON SECURITY ADVISORY FRAMEWORK (CSAF)

CSAF, developed by the OASIS CSAF Technical Committee, is an international, open standard for producing, distributing, and discovering machine-readable security advisories and allowing for automated vulnerability assessment. The CSAF standard uses profiles to define the necessary content for specific use cases, extending from the base profile by requiring additional fields. The security advisory (profile 4) and VEX (profile 5) are two of the more popular use cases. (Source: Adapted from OASIS CSAF Version 2.0 documentation)

COMPONENT

An entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis (ISO19770-2:4.1.3). Component refers to a part of a whole, such as a component of a software product, a component of a software identification tag, etc. (Source: NISTIR 7695)

CONSUMER

The organization or person (or customer) that receives a product or service. (Source: NIST SP 800-213 and under Customer from ISO 9000:2015)

ISOLATION

The ability to keep multiple instances of software separated so that each instance only sees and can affect itself. (Source: NIST SP 800-190)

ISOLATED

System functions are separated from other functions by means of an isolation boundary implemented within a system via partitions and domains. The isolation boundary controls access to and protects the integrity of the hardware, software, and firmware that perform system security functions. (Source: NIST SP 800-53r5 – SC-3 Modified)

KNOWN EXPLOITED VULNERABILITIES (KEV) CATALOG

An authoritative source of vulnerabilities that have been exploited in the wild; provided by DHS CISA, it represents a subset of Common Vulnerabilities and Exposures in the NIST NVD. It should be an input to vulnerability management prioritization. (Source: CISA Known Exploited Vulnerabilities Catalogue)

MACHINE LEARNING

Machine learning is a foundational branch of AI and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. (Source: Modified from IBM)

OPERATIONAL TECHNOLOGY ASSET

Programmable systems or devices that interact with the physical environment (or manage devices that interact with the physical environment). These systems or devices detect or cause a direct change through the monitoring or control of devices, processes, and events (e.g., industrial control systems, building management systems, fire control systems, and physical access control mechanisms). (Source: NIST SP 800-37 Rev. 2)

OPEN SOURCE PROJECT OFFICE (OSPO)

An organizational focal point designed to do the following: (1) be the center of competency for an organization's open source operations and structure; and (2) place a strategy and set of policies on top of an organization's open source efforts. (Source: Linux Foundation's TODO group)

PRODUCT

A complete set of computer programs, procedures, and associated documentation and data designed for delivery to a software consumer (ISO19770-2:4.1.19). (Source: NISTIR 7695)

PROVENANCE

The chronology of the origin, development, ownership, location, and changes to a system or system component and associated data. It may also include personnel and processes used to interact with or make modifications to the system, component, or associated data. (Sources: NIST SP 800-161r1, NIST SP 800-37 Rev. 2, NIST SP 800-53 Rev. 5)

RISK MANAGEMENT

The program and supporting processes to manage information security risk to organizational operations (including mission, functions, image, and reputation), organizational assets, individuals, other organizations, and the nation, which includes: (1) establishing the context for risk related activities; (2) assessing risk; (3) responding to risk once determined; and (4) monitoring risk over time. (Source: NIST SP 800-39)

SERVICE

An offering, capability, or delivery of ICT functionality that does not require the user or customer to purchase, own, and operate the underlying ICT product, including managed services that require a software component be installed on an end item which allows access to the functionality. (Source: "ICT SCRM Task Force Threat Evaluation Working Group: Supplier, Products, and Services Threat Evaluation (to Include Impact Analysis and Mitigation) Version 3.0")

SOFTWARE

All or part of the programs, procedures, rules, and associated documentation of an information processing system (ISO19770-2:4.1.25). (Source: NISTIR 7695)

SOFTWARE AS A SERVICE (SaaS)

The capability provided to the consumer to use the service provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email) or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. (Source: NIST SP 800-145)

SOFTWARE ASSURANCE (SwA)

The level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its lifecycle, and that the software functions as intended by the purchaser or user. (Source: NISTIR 8074 Vol.2; CNSSI 4009-2105; DoDI 5200.44)

SOFTWARE BILL OF MATERIALS (SBOM)

A formal record containing the details and supply chain relationships of various components used in building software. It is a nested inventory of software, a list of ingredients that make up software components. (Source: "The Minimum Elements for a SBOM" published by the NTIA.)

SOFTWARE OPERATOR

Software operators are responsible for the delivery and maintenance of software services, including maintaining service level agreements, service up-time, and mitigating risk, e.g., patching vulnerabilities. Operators use a combination of acquired software and subscriptions to SaaS products to create value for their users. (Source: Software Transparency in SaaS Environments)

SOFTWARE SUPPLY CHAIN

A software supply chain consists of everything that goes into software until the point when users touch it. (Source: Synopsys Glossary Article – "What is software supply chain security?")

SUB-CONTRACTOR

Any supplier, distributor, vendor, or firm that furnishes supplies or services to or for a prime contractor or another sub-contractor. (Source: FAR 44.101 Definitions)

SUBCONTRACT

Any contract as defined in subpart 2.1 entered into by a sub-contractor to furnish supplies or services for performance of a prime contract or a subcontract. It includes but is not limited to purchase orders, and changes and modifications to purchase orders. (Source: FAR 44.101 Definitions)

SUPPLIER ROLES

Roles for the supplier, developer, system integrator, external system service provider, and other ICT/OT-related service provider personnel responsible for the success of the program should be noted in an agreement between the acquirer and these parties (e.g., contract). (Source: NIST SP 800-161r1)

SUPPLIER

Organization or individual that enters into an agreement with the acquirer or integrator for the supply of a product or service. This includes all of those in the supply chain: developers or manufacturers of systems, system components, or system services; systems integrators; software producers; providers; vendors; distributors; product resellers; and third-party partners. (Source: NIST SP 800-53 Rev. 5,)

VENDOR

A commercial supplier of software or hardware. (Source: NISTIR 4734)

SOFTWARE PRODUCER

Attests to the use of secure software development practices (NIST SSDF, SP 800-218) by providing federal agencies with attestations of secure software development practices for the produced software product. (Source: OMB M-23-16)

SERVICE PROVIDER

Delivers services, such as network, application, infrastructure, and security via ongoing and active administration within a third-party data-center. Common service providers are cloud service providers, providers of SaaS solutions, and managed service providers which themselves may be suppliers within a software supply chain. (Source: Derivative of Gartner Definition for Managed Service Provider)

VULNERABILITY DISCLOSURE REPORT (VDR)

Used to demonstrate proper and complete vulnerability assessments for components listed in SBOMs in accordance with NIST Guidance. The VDR should include the analysis and findings describing the impact (or lack of impact) that the reported vulnerability has on a component or product. The VDR should also contain information on plans to address the Common Vulnerability Exposure. (Source: NIST SP 800-161r1 RA-5) "Maintain vendor vulnerability disclosure reports at the SBOM component level"

VULNERABILITY EXPLOITABILITY EXCHANGE (VEX)

Indicates the status of a software product or component with respect to a vulnerability. A common VEX use case is to indicate that software is or is not affected by a vulnerability. While primarily designed for software vulnerabilities, VEX can convey status for vulnerabilities involving hardware, specifications, and other causes. (Source: "CISA Minimum Requirements for Vulnerability Exploitability eXchange (VEX)")

APPENDIX B

UNDERSTANDING THE GOVERNANCE CONTROL IMPLICATIONS

The Supplier Governance and Attestations section of this guide contains 19 specific questions designed to reduce any application security and software assurance knowledge gaps that may exist between a supplier of a software-based product and those in the acquisition and procurement process. It is not intended to replace any review questionnaires that an acquisition team might have, but instead to augment such questionnaires by focusing attention on the risks associated with software throughout its lifecycle.

At a minimum, the desired outcome is to select software that meets both functional and operational requirements and was created with an appropriate level of security awareness. This goal is accomplished if the selected supplier can affirmatively answer all the questions in the Governance section without need for a POA&M; reducing the need to then provide responses to any remaining questions in the sections covering Supply Chain, Development, Deployment, and Vulnerability Management.

This appendix provides a rationale for each of the questions in the Governance section and should serve as a baseline to understand the rationale for all other CONTROL questions. It is worth noting that “No” answers to CONTROL questions where TASK questions exist provide a baseline for POA&M definition.

RATIONALE FOR GOV.01

In OMB memo M-22-18, OMB placed a strong focus on the importance of the NIST SSDF. In doing this, OMB also created an expectation that attestation to portions of the SSDF would be required. OMB memo M-23-16 provided a timeline for collection of attestation forms. The CISA Secure Software Development Attestation Form itself aligns with the SSDF. Suppliers who are not able to attest to their conformance to the SSDF, without needing a POA&M, likely have software development processes in need of maturation. The list of controls that can be skipped in GOV.01, following an affirmative response, outlines software assurance areas the supplier should focus attention on as they mature their processes. However, suppliers that have not completed the attestation form or may not be able to make a positive affirmation to all the criteria on the form may still be able to skip some of the questions covered by GOV.01 if they are able to answer “Yes” to the other GOV questions.

RATIONALE FOR GOV.02

This document follows the NIST SP 800-161r1 definition of provenance, which defines provenance information to include origin, development locations, ownership, and location for software. Software supply chains are very complex and will include third-party contracted, commercial, and open source software. As an example of this complexity, some studies have placed the average number of open source components used in commercial software to be 600, though that number varies based on the programming language used. If a supplier is not tracking the origin of the software components used in their products, then they likely also are not keeping track of any associated updates or patches. Similarly, if they do not have adequate visibility into the development process used by their third-party suppliers or the ownership of the development teams, then unexpected functionality might be present in the supplier product.

RATIONALE FOR GOV.03

Modern software development processes are complex. The source code used to create modern software powered products is also complex. Manual security reviews and intermittent testing practices simply do not scale to the reality of how complex software development processes have become.

This makes the usage of automated tooling, risk-based deployment of automation, and tactical usage of manual review processes an important aspect to reduce software development risks.

Such checks should not be limited to software development by a supplier but should extend to reviews performed prior to component or service provider selection and have a focus on the ongoing sustainability of a chosen software dependency.

RATIONALE FOR GOV.04

Testing for exploitable weaknesses and vulnerabilities in software should be part of any software development process. This includes verification and validation that there are no vulnerabilities in software libraries used by the supplier, tooling powering the development of software, or automation of the software integration process. If a supplier is not performing such detailed vulnerability analysis, ideally with tooling supporting continuous monitoring for new risks, and proactive notification of any identified risks, then should the suppliers' software development process become compromised the supplier might struggle to identify the nature of the compromise. Log management and patch management are among key processes needed to reduce risk to both development and operations.

RATIONALE FOR GOV.05

A supplier's release criteria for software-powered products might include "post-ship" items which are known software defects that they intend to resolve in a future release. Some of these post-ship items might include vulnerabilities that the development of a patch was deemed too costly to the release timeline. Independent of the feasibility of creating a patch for post-ship items, the disclosure of their existence provides software operators with an opportunity to apply contextually appropriate mitigations to how they deploy the software. Any software provider who cannot provide a report detailing the existence of unpatched vulnerabilities lacks a process to either review their software for unpatched vulnerabilities or to determine their impact.

Most suppliers will communicate new vulnerabilities in the software that they produce. If that communication model does not align with existing patch management best practices, then maintaining the software in a patched state becomes overly difficult. CSAF and Open Source Vulnerability (OSV) represent existing models that are widely deployed and VEX represents an emerging standard for vulnerability disclosures. Note: CSAF is the replacement for the Common Vulnerability Reporting Framework (CVRF). It enhances the capabilities of CVRF including different profiles (e.g., CSAF Base, Informational Advisory, Incident Response, VEX, etc.). Each profile extends the base profile "CSAF Base"—directly or indirectly through another profile from the standard—by making additional fields from the standard mandatory. A profile can always add, but never subtract nor overwrite requirements defined in the profile it extends. CSAF also provides several additional enhancements that were not supported in CVRF.

RATIONALE FOR GOV.06

Software whose default deployment configuration represents a secure deployment is inherently more resilient to the impact of misconfiguration than software that requires expertise by a software operator to properly maintain operational security. This is due to the reality that the software operator rarely has the level of contextual awareness that a supplier has in the implications of a misconfiguration. By using a secure by default model, suppliers are able to use their domain expertise to improve the security outcome of the software.

RATIONALE FOR GOV.07

Software development processes are complex, and development teams tend to trust that their processes are free from external compromise. As many cyberattacks have proven, trust placed on team members by the team is often exploitable—even if only by accident. Through the use of secure software development processes that rely on secure development environments, where all access to code and resources is based on contextually appropriate authorization consistent with zero-trust principles, risks associated with human trust within development teams can be minimized. Certification of software provides evidence that it was developed in compliance with the respective standards or frameworks.

RATIONALE FOR GOV.08

OMB Memo M-22-18 defines a requirement for suppliers to attest to their software development practices. This governance question seeks to identify if the supplier verifies the M-22-18 attestation status for their direct suppliers as a matter of policy. Security controls, processes, and policies should naturally evolve based on the threat landscape and associated best practices for software development. Best practices for software development and security controls should have direct line of sight to recognized standards, such as those from NIST. Having the ability to both trace the implementation and validation of controls back to standards is a sign of an organization that is prioritizing security requirements.

RATIONALE FOR GOV.09

An SBOM is a machine-readable file detailing the software components, or libraries, used to create the supplier's software. In response to Executive Order 14028, the NTIA defined a set of minimum fields an SBOM should contain to be usable. The NTIA also outlined that acceptable formats for an SBOM are SPDX, CycloneDX, and SWID. Both SPDX and CycloneDX have received multiple updates since NTIA issued its guidance. Note: SBOM-related work has transitioned to CISA from NTIA.

RATIONALE FOR GOV.10

Usage of community source software, commonly known as open source software, enables software development teams to increase the pace of innovation. Since the composition of the community creating each piece of open source software differs, as do their testing and coding standards, it is important for consumers of open source software to have policies surrounding its usage.

Given that open source software development follows a community development model, the release criteria, important security targets, and testing processes for code release may not align with the security targets of the software utilizing an open source library or solution. Security is one example of a risk present in open source software that an OSPO helps mitigate. An OSPO typically will work with both governance and development teams to create policies related to the responsible consumption of open source software and contribution back to open source efforts.

While generative AI and large learning models (LLMs) differ greatly from an open source community, they share a common attribute of indeterminant transparency for code generation, testing, and release criteria. ChatGPT is arguably the most well-known LLM, but it is not the only LLM that can be used to create or test software. Since the LLM space is nascent, acquisition teams should look for suppliers to have a policy surrounding the usage of an LLM by their software development teams. This CONTROL question is not seeking to make a judgement call on the utility of LLM solutions, but to understand if the supplier has policies indicating that they are attempting to understand the implications of LLMs and similar generative solutions.

RATIONALE FOR GOV.11

FedRAMP focuses on the ability to use cloud technologies in a secure and risk-informed manner. Since modern distributed software is often powered by cloud services and deployed using scalable cloud technologies, it is important for acquisition teams to know if suppliers are following a risk-informed deployment model. An affirmative answer to GOV.16 demonstrates that the supplier has received an independent third-party review of its deployment practices.

RATIONALE FOR GOV.12

Suppliers should have protections in place and convey cloud security requirements to third-party suppliers depending upon customer needs, including the possible need for people with background checks, especially for suppliers with access to government operational information.

RATIONALE FOR GOV.13

Software testing that focuses on whether features perform as intended may not operate as intended when those features are exposed to a hostile environment, such as instances on the modern internet, or where critical functions or critical data entice an adversary. This governance control seeks to identify whether a supplier has defined a C-SCRM program that routinely incorporates contextual risk assessments and mitigation efforts during product development and throughout the product lifecycle.

RATIONALE FOR GOV.14

Developing software in a secure manner, with an understanding of the potential threats that software might experience, requires continuous education and skills verification. Multiple studies have shown that software development teams who embrace continuous education based on the role individual developers have within the team are able to produce higher quality software with fewer weaknesses at a lower cost.

RATIONALE FOR GOV.15

As identified in Executive Order 14028, the SolarWinds experience reminded the industry that suppliers need to have complete control over their software production processes and artifacts that are produced from such processes. If an attacker can compromise any element in the production process, then the supplier is no longer able to reliably assert that the software performs as intended.

RATIONALE FOR GOV.16

Suppliers should prioritize re-use of known validated software libraries over reimplementing functions or creating novel implementations for standardized functionality—particularly when the library is part of a security function in the software. It is a safe assumption to note that all software has bugs, but the quantity and severity of those bugs are often greater earlier in the lifespan of a library. Through the reuse of known validated libraries and secure configurations, development teams benefit from prior testing efforts that might be unrelated to the current product's development.

RATIONALE FOR GOV.17

Executable security requires properly configured compilers, interpreters, and build tools; it is important that customers convey these expectations to the software suppliers.

RATIONALE FOR GOV.18

The effective security of any software is related to the effective security of each of the constituent components making up that software. This control seeks to identify whether the supplier holds their direct suppliers, contractors, and service providers to the same standards that they hold their in-house development teams. Since most questions in this guide focus on the development practices of a supplier's in-house development teams, if a supplier does hold their supply chain to that same standard, then they are effectively managing the risk from their direct suppliers.

RATIONALE FOR GOV.19

Many commercial license agreements have exclusionary clauses related to the analysis of the software covered by that license agreement. If a license agreement precludes analysis of the software for security issues or to determine the provenance of the software components used in the software, then acquisition teams may have limited options to verify that the software meets the security requirements of the purchasing organization.

APPENDIX C

SECURE DEVELOPMENT AND DEPLOYMENT GUIDANCE

The USG (e.g., OMB, NIST, CISA) has provided software consumers with guidelines to assist with the Secure Software Development and Deployment within a digital ecosystem. This guidance includes, but is not limited to the following materials:

- NIST SP 800-161 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations
- NIST SP 800-218 SSDF Version 1.1
- Attesting to Conformity with Secure Software Development Practices
- CISA Secure Software Development Attestation Form
- CISA Secure by Design
- Software Security in Supply Chains: Software Bill of Materials (SBOM)
- Cross-Sector Baseline Cybersecurity Performance Goals (CPGs)
- ICT SCRM Task Force Threat Evaluation Working Group: Supplier, Products, and Services Threat Evaluation (to Include Impact Analysis and Mitigation) Version 3.0)
- OMB memo M-21-31 – Improving the Federal Government's Investigative and Remediation Capabilities Related to Cybersecurity Incidents
- OMB memo M-22-18 – Enhancing the Security of the Software Supply Chain through Secure Software Development Practices and other NIST Guidance
- OMB memo M-23-16 – Update to Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices
- OMB memo M-23-18 – Administration Cybersecurity Priorities for the FY 2025 Budget
- Regulatory Guide 1.152, Revision 3, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants"
- CISA Known Exploited Vulnerabilities Catalog
- National Cybersecurity Strategy, published March 2, 2023
- Enduring Security Framework documents on SCRM and SBOM Management:

  Securing the Software Supply Chain for Customers
  Securing the Software Supply Chain for Developers
  Securing the Software Supply Chain for Suppliers
  Recommended Practices for Software Bill of Materials Consumption
  Recommended Practices for Managing Open Source Software and Software Bill of Materials
- NSA Recommendations for Software Bill of Materials (SBOM) Management
- Cross walk between ISO 23894 and NIST AI framework
- Understanding Baselines and Impact Levels in FedRAMP
- GSA Acquisition Letter MV-2023-02, Supplement 1

APPENDIX D

CONSIDERATIONS FOR FEDRAMP, CAPABILITY MATURE MODEL CERTIFICATION, AND NIST SPECIAL PUBLICATIONS

NIST Special Publications (SPs) form the basis of several frameworks and USG guidelines and provide excellent criteria from which to establish practices and procedures; yet by themselves do not offer a means for third-party assessment or certification. The Capability Mature Model Certification, while offering a corresponding independent assessment and certification, addresses protection of information; yet does not offer insight to secure software development or deployment. The following FedRAMP requirements satisfy the Deployment Controls in this Guide:

GUIDE CONTROL NUMBER	SOFTWARE DEPLOYMENT CONTROL DESCRIPTIONS	FEDRAMP MAPPING
CONTROL.DEP.01	Has the software supplier implemented MFA for all login access to accounts for both local and remote access?	IA-2 and IA-8
CONTROL.DEP.02	Does the software supplier's organization maintain updated cybersecurity incident response plans related to operating the software and development of the software?	IR-01
CONTROL.DEP.03	Does the software supplier follow best practices for log management as defined in NIST SP800-92 or OMB M-21-31?	AU-02
CONTROL.DEP.04	Does the supplier have a defined patch process for all software, tools, and systems used in the delivery of the software?	SA-22
CONTROL.DEP.05	Does the software supplier require its suppliers to have a process in place, such as scanning, to address undocumented codes or functions that might allow unauthorized access or use of the software product or cause the software product to behave outside of the specified requirements or in an unreliable manner?	AC-17
CONTROL.DEP.06	Does the supplier have protections in place between the software supplier's network and cloud service providers, including protections associated with contractually specified government protection levels and personnel background checks if applicable?	PS-03

GUIDE CONTROL NUMBER	SOFTWARE DEPLOYMENT CONTROL DESCRIPTIONS	FEDRAMP MAPPING
TASK.DEP.06.01	If the supplier has connectivity to a government system requiring special protection levels, does the supplier have established policies in place requiring background checks and screening for personnel and requisite information handling procedures (including clearance levels and formal access approvals) appropriate to the protection level associated with the interconnected government system?	SR-05
TASK.DEP.06.02	Does the supplier enforce the screening and formal access approval process for its personnel and sub-contractors (including service providers) that have access to the government system requiring special protection levels?	PS-03
TASK.DEP.06.03	Does the software supplier convey cloud security requirements to sub-suppliers and sub-contractors?	AU-02
CONTROL.DEP.07	Does the software supplier's product support hardened security configurations that enforce the principles of least privilege, separation of duties, and least functionality?	CM-06
CONTROL.DEP.08	Does the software supplier have methods to verify the trust relationship between a product supplier and the party identified within a digital signature for a product installation package?	IA-05
CONTROL.DEP.09	Do the binary packages distributed by the software supplier implement cryptographic signatures to ensure they are not manipulated or tainted?	CM-07(05)
CONTROL.DEP.10	Does the software supplier certify their software against applicable standards and maintain that certification?	MA-01
CONTROL.DEP.11	Does the supplier provide detailed deployment guidance for the software?	CM-07
CONTROL.DEP.12	If the software requires the use of a service provider, does the supplier perform any pre-installation integrity or validation checks to ensure that the software continues to meet deployment or acceptance criteria associated with an authority to operate?	RA-03

APPENDIX E

MAPPING OF GOVERNANCE CONTROLS TO SKIPPABLE QUESTIONS

The following table lists which CONTROL questions in each section that may be skipped if an affirmative response to the associated Governance question is provided.

GOVERNANCE CONTROL	SUPPLY CHAIN	SOFTWARE DEVELOPMENT	SOFTWARE DEPLOYMENT	VULNERABILITY MANAGEMENT
GOV.01 Secure Software Development - Attestation Form	SC.04, SC.07, SC.08	DEV.03, DEV.07, DEV.08, DEV.09, DEV.10, DEV.11, DEV.12, DEV.14, DEV.20, DEV.21, DEV.22, DEV.23, DEV.26, DEV.27, DEV.28, DEV.30	DEP.07, DEP.09, DEP.11	VULN.01, VULN.04, VULN.07
GOV.02 Provenance Data	SC.01, SC.04, SC.08	DEV.03, DEV.12, DEV.16, DEV.30		
GOV.03 Vulnerability Mitigation	SC.07	DEV.09, DEV.24	DEP.03, DEP.04, DEP.09, DEP.12	VULN.02, VULN.03, VULN.04, VULN.05, VULN.06, VULN.08
GOV.04 KEV Testing		DEV.21, DEV.22, DEV.27, DEV.28, DEV.30		VULN.02, VULN.05, VULN.06
GOV.05 Vulnerability Mitigation		DEV.30		VULN.01, VULN.03, VULN.07, VULN.08
GOV.06 Secure by Default		DEV.29	DEP.01, DEP.07, DEP.11	

GOVERNANCE CONTROL	SUPPLY CHAIN	SOFTWARE DEVELOPMENT	SOFTWARE DEPLOYMENT	VULNERABILITY MANAGEMENT
GOV.07 Zero-Trust and Secure by Design		DEV.01, DEV.03, DEV.07, DEV.08, DEV.09, DEV.11, DEV.12, DEV.16, DEV.18, DEV.20, DEV.23, DEV.26	DEP.10	
GOV.08 Requirements Flow Down	SC.01, SC.04, SC.07, SC.08	DEV.10, DEV.21, DEV.22		
GOV.09 SBOM	SC.02, SC.08			
GOV.10 AI and Open Source	SC.03, SC.04, SC.06			
GOV.11 FedRAMP Certification			DEP.01, DEP.02, DEP.03, DEP.04, DEP.05, DEP.06, DEP.07, DEP.08, DEP.09, DEP.10, DEP.11, DEP.12	VULN.01, VULN.03, VULN.06, VULN.07
GOV.12 Protection Level		DEV.04	DEP.06, DEP.08, DEP.09	
GOV.13 C-SCRM	SC.05	DEV.06, DEV.15	DEP.02	
GOV.14 SDLC Training		DEV.04, DEV.05, DEV.18		
GOV.15 Software Security Requirements		DEV.02, DEV.10, DEV.13		

GOVERNANCE CONTROL	SUPPLY CHAIN	SOFTWARE DEVELOPMENT	SOFTWARE DEPLOYMENT	VULNERABILITY MANAGEMENT
GOV.16 Secure Components		DEV.17, DEV.19		
GOV.17 Executable Security		DEV.07, DEV.24, DEV.25		
GOV.18 Requirements Flow Down	SC.01, SC.04		DEP.02, DEP.05	
GOV.19 Provenance & Security Scanning				

APPENDIX F

ESTIMATED TIME FOR RESPONSE TO CONTROL QUESTIONS

It is conceivable that a small supplier with a small number of parties contributing to the SDLC process and a small software product could answer the entire set of Governance questions within 10 staff hours, including research per product sold to the USG. Some products will require more time and effort due to multiple factors including the number of third-party suppliers and products used, the total number of externally acquired components that contribute to the final product distributed to customers, and the number of SDLC teams involved in product manufacturing, design, development, and maintenance/distribution.

A large product with numerous third-party components could require considerable effort and time to complete the spreadsheet.

See the "Software Acquisition Guide for Government Enterprise Consumers" spreadsheet.

SOFTWARE ACQUISITION GUIDE

FOR GOVERNMENT ENTERPRISE CONSUMERS:

Software Assurance in the Cyber-Supply Chain Risk Management (C-SCRM) Lifecycle

PUBLICATION: AUGUST 2024

INFORMATION AND COMMUNICATIONS TECHNOLOGY
SUPPLY CHAIN RISK MANAGEMENT TASK FORCE