



**IT Security Procedural Guide:  
OCISO DevSecOps Program  
CIO-IT Security-19-102**

**Revision 2**


**April 19, 2023**

**VERSION HISTORY/CHANGE RECORD**

<b>Change Number</b>	<b>Person Posting Change</b>	<b>Change</b>	<b>Reason for Change</b>	<b>Page Number of Change</b>
		<b>Initial Release – September 26, 2019</b>		
N/A	ISE	New guide created to integrate security into DevOps teams.		N/A
		<b>Revision 1 – September 9, 2022</b>		
1	ISE	Updates include: <ul style="list-style-type: none"> <li>• Minor updates in language</li> <li>• Removed links to old AWS checklist</li> <li>• Minor alignment for upcoming major update of the guide</li> </ul>	Periodic Update.	Throughout
2	McCormick/Klemens	<ul style="list-style-type: none"> <li>• Editing and formatting</li> </ul>		Throughout
		<b>Revision 2 – April 19, 2023</b>		
1	ISE	Major changes to identify DevSecOps strategy, goals, and incorporate separation of duties guidance.	Align to current GSA process and guidance.	Throughout
2	McCormick/Klemens	Updates include: <ul style="list-style-type: none"> <li>• Collaborated with DevSecOps team on restructure of guide.</li> <li>• Edited and updated to current guide format and structure.</li> </ul>		Throughout

## Approval

IT Security Procedural Guide: OCISO DevSecOps Program, CIO-IT Security 19-102, Revision 2, is hereby approved for distribution.

DocuSigned by:  
  
FD717926161544F...

---

Bo Berlas  
GSA Chief Information Security Officer

**For questions concerning the DevSecOps Program, contact: GSA Office of the Chief Information Security Officer (OCISO), Security Engineering Division (ISE) at [ocisodevsecops@gsa.gov](mailto:ocisodevsecops@gsa.gov).**

**For questions concerning GSA Policy and Compliance, contact: GSA Office of the Chief Information Security Officer (OCISO), Policy and Compliance Division (ISP) at [ispcompliance@gsa.gov](mailto:ispcompliance@gsa.gov).**

## Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
1.1	Applicability and Responsibility .....	1
1.2	Purpose .....	1
1.3	Scope .....	1
1.4	Roles and Responsibilities.....	1
<b>2</b>	<b>Defining DevSecOps</b> .....	<b>4</b>
2.1	DevSecOps Program Goals.....	5
2.2	DevSecOps Capabilities .....	6
<b>3</b>	<b>ATO/Ongoing Authorization</b> .....	<b>6</b>
<b>4</b>	<b>ODP DevSecOps Strategy</b> .....	<b>7</b>
4.1	Integrated Working Model.....	7
4.2	Operational Security .....	8
4.3	Application Security (App Sec) .....	9
4.4	Change Management .....	10
4.5	Automation .....	11
<b>5</b>	<b>Separation of Duty</b> .....	<b>11</b>
5.1	Procedures for SOD .....	12
<b>6</b>	<b>Engaging with OCISO ODP</b> .....	<b>13</b>
6.1	Rules of ODP Engagement.....	13
6.2	Process for Initial ODP Engagement .....	14
	<b>Figure 4-1. OA Enablement via DevSecOps</b> .....	<b>7</b>
	<b>Table 2-1. DevSecOps Capabilities</b> .....	<b>6</b>
	<b>Table 5-1. Separation of Duty in a DevOps/DevSecOps Model</b> .....	<b>12</b>

Note: Hyperlinks will be provided on first occurrence of a reference, thereafter only the reference will be listed.

## 1 Introduction

With more teams at the General Services Administration (GSA) leveraging Development and Operations (DevOps) practices, ensuring effective security practices has become paramount. The Office of the Chief Information Security Officer's (OCISO) DevSecOps Program (ODP) aims to ensure that GSA teams who practice DevOps adopt security-forward thinking and facilitate the creation and operation of DevSecOps practices with related tools and processes at GSA.

### 1.1 Applicability and Responsibility

The instructions and procedural guidance identified in this guide apply to both federal and contract teams supporting GSA information systems operating under a DevOps or DevSecOps model.

All federal and contract teams supporting GSA information systems operating under a DevOps or DevSecOps model have the responsibility to adhere to the requirements contained in this guide. The product and/or system owner and authorizing official (AO) have primary responsibility to ensure the requirements are met.

### 1.2 Purpose

This guide serves to establish and integrate the ODP throughout GSA's development, operations, and security organizations and to facilitate the creation of DevSecOps-related tooling and processes for adoption of DevSecOps at GSA.

The ODP emphasizes security as the central component of DevOps teams, effectively creating DevSecOps teams at GSA. The ODP will be run by the OCISO Security Engineering (ISE) Division. This procedural guide establishes a process and operating principles for the ODP and team adoption of DevSecOps practices.

### 1.3 Scope

This guide describes the OCISO DevSecOps Program, which is available to GSA on-prem or hybrid information systems with a primary focus on cloud-based systems. ODP resources and services consist of security requirements and guidance, security engineering consulting, policy/process/standards review, secure architecture design, security tooling integration, reusable code, documentation, reference architecture, enterprise security tools and shared services, and embedded security engineer(s).

### 1.4 Roles and Responsibilities

The ODP envisions DevSecOps teams as cross functional agile teams and aims to integrate security engineers as subject matter experts (SME) from the OCISO side into these teams. Well defined roles and responsibilities are imperative for cross functional DevSecOps teams. GSA system teams desiring integration with the OCISO DevSecOps program shall adhere to these high-level roles and responsibilities. However, to support agility and based on the maturity of the team, these roles and responsibilities will be reviewed prior to each engagement. Roles and responsibilities will be finalized mutually between the ODP program and the integrated system teams.

### 1.4.1 ODP DevSecOps/Ongoing Authorization (OA) Engineer

The priority of the embedded DevSecOps Engineer is security, focusing on security design, operational security, application security (AppSec), security and compliance impact analysis during change management, and security/compliance automation. The DevSecOps Engineer serves as the overall Security SME/Champion for the assigned system team. The engineer assigned to this role could also be designated as OA engineer, upon agreement between ODP and the integrated team.

#### Responsibilities:

- Works with the system team on all aspects of system security in collaboration with the DevSecOps team which includes security designs, security architecture, implementation, operations, and compliance.
- Engages directly with the integrated team in solution design, sprint planning, story creation, defining acceptance criteria, and ensuring security requirements are properly addressed in early phases.
- Interprets security requirements, policy, standards, control statements, and its applicability for system team and/or system implementation.
- Provides threat modeling and threat analysis (if required).
- Acts as a liaison between the security organization and divisions as needed.
- Coordinates directly with ODP and other OCISO divisions for security-related questions, clarifications, decision points, reviews, etc., as needed.
- Integrates into the existing change management process as security reviewer/contributor.
- Collaborates with the system team for security code review and compliance impact analysis.
- Establishes and maintains a security-related operating procedure for system teams (e.g., rapid risk assessment procedure, a procedure for engaging GSA Incident Response [IR] team).
- Provides support, code, and consultation for integration with security tools and services.
- Collaborates with the System Owner, Information System Security Manager (ISSM), and Information System Security Officer (ISSO) to support development and maintenance of compliance documentation (e.g., System Security and Privacy Plan [SSPP], Plan of Action & Milestones [POA&M]).
- Builds reusable security code, build code library, security automation, security checklist, security best practices, security wiki, etc.
- Provides vulnerability management in the form of standardizing tooling policies across tools, providing initial triage, vulnerability exceptions, and/or coordination for the system team.
- Supports system specific user on-boarding, off-boarding and access management, inventory management, environment on-boarding and off-boarding , establish, and provide alert review/remediation/suppression, and escalation.
- Supports application, databases and business logic related logs and dashboard monitoring (as needed).
- Supports Security Operations Center (SOC) escalation review, triage, and decisions. Coordinates and performs IR exercises and tailors enterprise IR playbooks as per system specific needs.
- Defines system specific patterns, indicators of compromise (IoC), and behavioral indicators for custom SOC dashboards (as needed).

- Coordinates with the system team to define criteria to integrate the system, establish fail gate, and enforce fail gate.
- Supports ticket creation and reviews related to firewall, Domain Name System (DNS), tooling integration, tooling access, and system specific tooling policy changes and tuning, Git pull request, Infrastructure as Code (IaC), and application feature minor release.

### 1.4.2 ODP Core Team

The ODP Core Team provides authoritative decisions on technical aspects of cybersecurity to system teams. The ODP Team acts as a security advisor, provides day to day support and a collaborative platform for all security engineers, DevSecOps engineers, and security champions.

Responsibilities:

- Runs collaborative platform and scrums to support integrated team and security engineers.
- Develops and maintains DevSecOps security checklists, wiki, guardrails, implementation guides, and processes and procedures for DevSecOps best practices.
- Provides authoritative technical guidance, decisions, and approvals for questions and requests received by security engineers and/or integrated DevSecOps teams.
- Define standardized enterprise-wide policies for the security tools to scan the resources for both compliance and vulnerabilities. Manage the Policies as Code (PaC) in the Git repositories and follow GSA's change management process for policy changes and deployments.
- Create automated build pipelines for building security hardened virtual machine images and container images to meet GSA benchmark compliance requirements, scanning them for compliance, and certifying them before distributing for consumption.
- Build repetitive decision-making processes and guidelines to empower security engineers.
- Works closely with CISO and/or CISO designee for authoritative decision making when the team needs additional guidance.
- Works closely with CISO and/or CISO designee for Assessment & Authorization (A&A) related assessment and final sign-off.

### 1.4.3 System Product Team

The System/Product Team, which includes the ODP DevSecOps or OA engineer(s), provides the day to day operations of all the aspects of the system and/or application. The team could have different SMEs, but they are ultimately responsible for all aspects of the system and/or application including security and compliance.

Responsibilities:

- Manages system/application, including system design, code development, operation, and security including secure design and integration, alerts and incident monitoring, security documentation and compliance, etc. based on business objectives and mission.
- Adopts the DevSecOps culture and working model, which supports continuous releases, upgrades, and changes while fully maintaining security posture, principle, and compliances of the application/system.

- Develops standard operating procedures for security monitoring, investigating alerts, taking corrective action, and engaging incident response teams as needed.

#### 1.4.4 System Owner

The system owner provides overall ownership of a system/application including security and compliance. System Owner drives one or more workflow teams which are responsible for different aspects/components and/or sub-components of the system.

Responsibilities:

- Provides high level system requirements, resolves security versus functional priorities, and makes operational decisions.
- Sets priorities and manages integrated team, time, and resources.
- Manages tasks and priorities of security engineers for allocated time on each sprint cycle.
- Manages the onboarding, integration, and establishment of a working model for effective collaboration between security engineers and existing teams.
- Measures and monitors program success against established and agreed metrics continuously.
- Develops a plan and implements security requirements, checklist, guardrails, policy, and procedure agreed as part of the integration.
- Collaborates with OCISO along with integrated security engineers for high-level decision making, review, and approvals as needed. (Especially, when such decisions are outside of established standards)
- Take full responsibility and ownership of the system, related decisions, and outcomes.

#### 1.4.5 Chief Information Security Officer (CISO)

The Chief Information Security Officer (CISO) provides leadership for the implementation and maintenance of the IT security program, including the ODP. The Chief Information Security Officer also provides a final authoritative decision on all questions, concerns, and guidelines requested by the ODP.

Responsibilities:

- Designates role or personnel to provide authoritative decisions, approvals, and guidelines for questions, concerns, approvals, and reviews requested by security engineers and integrated DevSecOps teams.
- Provides final authoritative decision on all questions, concerns, and guidelines requested by the ODP team and security engineers.
- Provides risk-based decisions and Authorizations to Operate (ATO) sign-off for integrated DevSecOps team systems/applications, as per existing policy and procedural guidelines.

## 2 Defining DevSecOps

DevSecOps is an iteration of the term DevOps, which originates from the idea of combining two previously siloed groups, Development and Operations. DevOps practices, tools, and a shifting cultural approach enable build and delivery of applications and/or services at greatly increased speed and at scale. DevSecOps makes security an equal partner in the workflow.



Like DevOps, DevSecOps can have slightly different definitions depending on the industry, organization, or teams. A DevOps team can range from “a simple cross-functional collaborative team including IT security personnel” to “a self-sustained, highly agile, self-managed team driving cultural shift.” While this guide will not establish a universal definition of DevSecOps, it describes how the ODP views DevSecOps.

At a high level, the ODP defines DevSecOps as “integrating security into all DevOps workflows and practices.” As the organization DevSecOps, which is dedicated to furthering the DevSecOps mission, states, “Everyone is responsible for security.”

The ODP envisions cross functional agile DevSecOps teams led by product owners operating in close collaboration with the business line, platform/shared services teams, and security teams. DevSecOps teams will utilize the GSA enterprise platform and shared services and tools when available for a particular capability and collaborate with GSA enablement teams across GSA enterprises as needed.

## 2.1 DevSecOps Program Goals

The GSA ODP has five goals:

- 1. Improve Security and Quality.** The ODP aims to ensure security is considered and implemented in the design, development, and operation phases. The ODP provides full security support in areas including:
  - Secure architecture design
  - Application security
  - Security tooling integration
  - Change management
  - Operational security
  - Security and compliance automation
  - Supporting security audits
  - Educating and empowering DevOps teams.
- 2. Facilitate a Cultural Shift.** Security is often equated with compliance. While compliance is an important part of the system life cycle, security is more than just compliance. The ODP aims to shift focus from ATO and compliance assessments to baking security considerations into every stage of the system lifecycle and educate GSA technical teams to adopt a “How can we do this securely?” mindset. Furthermore, the ODP encourages leveraging the team topology model to break down silos and build a cohesive environment to achieve the speed of delivery, stability of operations, and the security of the ecosystem as a whole.
- 3. Reduce Silos and Communication Barriers.** Security engagements with system teams usually occur when there is an incident or an assessment. A lack of information and code sharing can lead to “reinventing the wheel” development cycles. The ODP aims to provide a simple and consistent communication channel where solutions, code, and more can be shared amongst teams to make development cycles more efficient and secure.
- 4. Provide Security Services Through an Integrated Engineer.** Most often, OCISO security services are often available only during A&A, during IR, and in support of limited

ISSO functions. The ODP seeks to integrate a security engineer into a system's DevOps team to deliver security services and prescriptive guidance provided by different divisions and teams. The integrated engineer can also support ongoing authorization related functions and provide technical services for ODP security-related functions.

5. **Develop DevSecOps Policy, Process, Tooling, and Shared Services.** The ODP will develop necessary policies, processes, and products for the adoption of DevSecOps at GSA. The ODP will also establish shared security services, tooling, standards, and reusable components in collaboration with other DevSecOps teams at GSA.

## 2.2 DevSecOps Capabilities

GSA system teams require various capabilities to effectively adopt a DevSecOps practice. Adoption of various industry tools and capabilities depends on the maturity of the DevSecOps team. Adoption of some basic capabilities and tools like those listed in Table 2-1 are considered prerequisites to adopt DevSecOps; however, associated tools and instructions to adopt these capabilities are not in scope for this document. The ODP envisions the adoption of some foundational capabilities to be considered DevSecOps practice in GSA.

**Table 2-1. DevSecOps Capabilities**

Technical Capabilities	Process Capabilities	Measurement Capabilities	Cultural Capabilities
<ul style="list-style-type: none"> <li>• Code maintainability</li> <li>• Continuous delivery</li> <li>• Continuous Integration</li> <li>• Continuous testing</li> <li>• Empowered teams</li> <li>• Deployment Automation</li> <li>• Loosely Coupled Architecture</li> <li>• Shifting security to the left</li> <li>• Trunk-based development</li> <li>• Version control</li> </ul>	<ul style="list-style-type: none"> <li>• Customer feedback and engagements</li> <li>• Streamlined change approvals</li> <li>• Team experimentation</li> <li>• Work in small batches</li> </ul>	<ul style="list-style-type: none"> <li>• Monitoring system to inform business decisions</li> <li>• Monitoring and observability</li> <li>• Proactive failure notification</li> <li>• Work in process limits</li> <li>• Visual management capabilities</li> </ul>	<ul style="list-style-type: none"> <li>• Agile decision-making</li> <li>• Cross Functional ownership culture</li> <li>• Product ownership and responsibility on product team</li> <li>• Transformational leadership</li> </ul>

Source: [DevOps Research & Assessment \(DORA\) Capability Catalog](#)

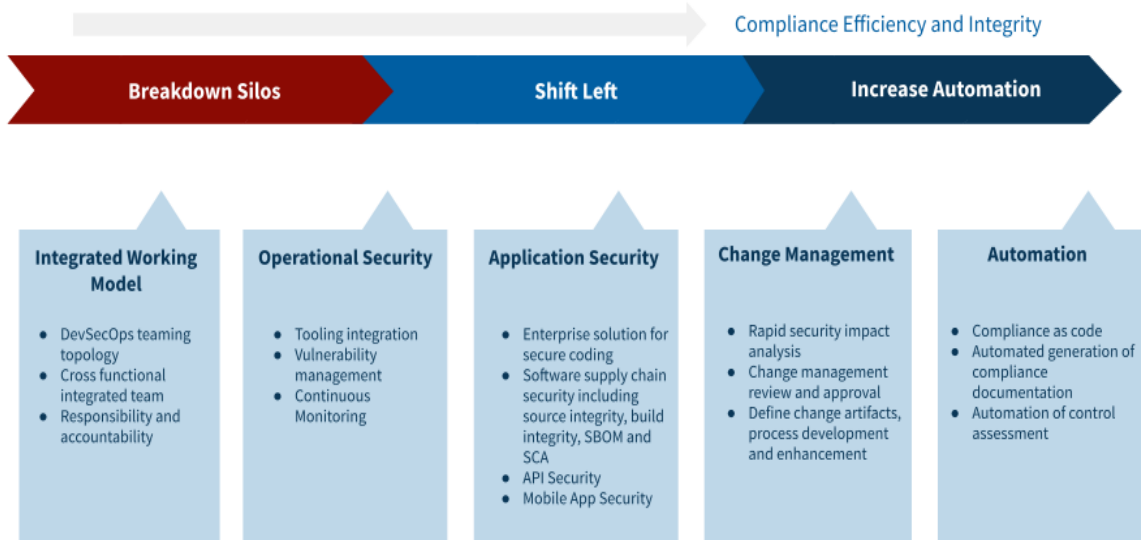
## 3 ATO/Ongoing Authorization

All GSA systems require an ATO as per [GSA Order CIO 2100.1](#), "GSA Information Technology (IT) Security Policy." All systems shall obtain an ATO following the existing GSA process regardless of whether they are operated and managed by following DevSecOps principles or not. The ODP program recommends system teams adhere to DevSecOps best practices, achieve full ATO, and onboard into the GSA ongoing authorization program, which is equivalent to continuous ATO, also called cATO industry wide.

System teams shall follow [CIO-IT Security 12-66](#): Information Security Continuous Monitoring (ISCM) Strategy & Ongoing Authorization (OA) Program to onboard into ongoing authorization for continuous delivery and release.

## 4 ODP DevSecOps Strategy

The ODP program identifies five different pillars for DevSecOps program advancement and adoption: Integrated Working Model, Operational Security, Application Security, Change Management, and Automation. Figure 4-1 provides details for each pillar.



**Figure 4-1. OA Enablement via DevSecOps**

### 4.1 Integrated Working Model

The ODP team envisions a system team as a cross functional team integrated among business line, platform team, application team, and OCISO security team and following DevSecOps principles and best practices. Team integration establishes communication and coordination channels among teams and provides rapid response for security requirements. Product ownership stays within a single cross functional team. The ODP DevSecOps/OA engineer acts as a cybersecurity SME for the system team and works in close collaboration with OCISO divisions.

#### 4.1.1 DevSecOps Team Components

The following teams and individual roles may provide DevSecOps expertise, as needed.

- ODP Core Team.
- System Product Team.
- System Owner.
- Chief Information Security Officer.
- ODP DevSecOps/Ongoing Authorization (OA) Engineer.

Listed below are general practices ODP DevSecOps/OA engineers should follow when engaged with the system teams.

- **Technology Alignment**
  - Adopt the same toolset used by the system team to define security related tasks and priorities in coordination with the scrum master and system owner. System owners will make final decisions on prioritizing activities.
  - Follow agile team practices such as Program Implement (PI) Planning and development in sprint cycles and adopt the same communication channel as the team for collaboration and coordination.
  - The ODP DevSecOps/OA engineers will require access to the toolset (e.g., JIRA, Trello, GitHub, GitLab), code repositories, cloud-based tools, and platforms (e.g., AWS Cloud Console), Jump Boxes, continuous integration/continuous delivery (CI/CD) pipelines, and other management tools (e.g., Jenkins, New Relic, Splunk) similar to those used by other DevOps engineers on the team.
- **Communication and Reporting**
  - Act as the direct line report to the ODP Team.
  - Collaborate with system product owners, ISSOs, and ISSMs, to work on assigned stories and determine work priorities. The GSA ODP program manager and product owner will collaborate on priorities and make decisions as needed. The GSA ISE director and system team director will collaborate on a scheduled basis.

## 4.2 Operational Security

The ODP program team envisions DevSecOps OA engineers engaging directly in operational security of the system managed and operated by the system team with defined responsibilities. Operational security shall be balanced with other priorities for enhancement and new features to avoid degradation of system and application security posture post assessment or ATO and maintain ongoing authorization.

The ODP team envisions full integration with applicable OCISO enterprise security tools and services along with effective vulnerability management, tracking and remediation. ODP integrated engineers shall play a central role in security tooling integration, operational maintenance of tooling integration and vulnerability management in coordination with other stakeholders including the ISSO/ISSM.

ODP integrated engineers should follow the operational security practices listed below when engaged with the product DevSecOps teams.

- Verify and generate status reports of security tooling integration and its functional status.
- Coordinate with ISSO to process reports from common vulnerabilities and exposures (CVE) scanning tools.
- Coordinate with ISSO to triage benchmark scanning reports. All findings need to be addressed with justification. For findings needing remediation, tasks should be created for the respective team.
- Support the GSA Enterprise SOC team for dashboard review, if needed, for system specific activities and behavioral patterns.
- Review any firewall change requirements based on technical needs and coordinate with the ISSO and SecOps for approvals and implementation. In the longer run, OA engineers may be able to review and approve these tickets for their specific application.
- On-board and offboard application team users in security tools following processes defined by security product owners.

- Tune system specific security policy and enforcement configuration in security tools following the process outlined by security product owners.
- Define, build, and maintain criteria for scanning Git repos for secrets, and security-related settings of Git repos.
- Review system vulnerabilities and coordinate for exception approval with ISSO/ISSM as required (e.g., operating system [OS] benchmark exceptions, CVE exceptions, exceptions in security tool policies). Once approved, implement an exception in the tool or coordinate with SecOps for exception. In the longer run, OA Engineers may be able to review and approve certain types of exceptions.
- Review IaC, configuration as code changes, and provide approval as needed based on assigned task(s) and security impact analysis.
- Fix broken security tools integration and agent connection in coordination with system teams.
- Ensure system inventory (e.g., servers, Uniform Resource Locators [URLs], containers) is up to date in security tools for scanning purposes in coordination with the ISSO. Further automation of the ISCM dashboard should be coordinated as an IS wide effort.
- Assist in the development of a system-wide IR plan, including responding to IoC, and support and define the scope of threat hunting in coordination with the ISSO and the GSA IR team.
- Define, build, and maintain fail gate criteria for security checks in the pipeline (e.g., container scanning fail gates, admission controllers, AppSec scan score in Static Application Security Testing (SAST)/Dynamic Application Security Testing (DAST) tools.

### 4.3 Application Security

The ODP program team envisions application security embedded into the software development life cycle and continued during the operations and maintenance (O&M) phase during the life of the software. Software development and DevSecOps teams shall integrate tooling and process to identify software supply chain security vulnerabilities in source code, perform dependency checks, enumerate software bill of materials (SBOM), and verify source and build integrity in the software development life cycle.

Listed below are application security practices that software development and DevSecOps teams should follow during software development and O&M phases.

- Adopt agile software development process in collaboration with business lines and related stakeholders.
- Follow agile processes and break down tasks in small, incremental units.
- Task story scope that has been clearly defined, reviewed, approved, and prioritized for development by product owner and scrum master.
- Establish a process to perform security and compliance impact analysis at the story, task, or epic level as appropriate.
- Define a standard development environment and practices for the team. The standard development environment should include a tool set for application developers and DevSecOps teams including a set of linters, and an integrated development environment (IDE) plugin for static code and dependency scanning. Standard code management practices such as uses of trunk branch, code ownership, branching and merging strategy shall be defined.
- Developers shall use linters and IDE plugins for static code and dependency scanning on an ongoing basis during code development. Security findings and code vulnerabilities identified by the plugin and linters shall be remediated. If critical or high vulnerabilities

from code scanning tools persist, developers shall engage with the OCISO AppSec Program for triage and remediation.

- If a developer needs to use a third-party library or dependency, the developer shall utilize the latest version of the library or dependency if possible. Despite using the latest version, if the dependency or library still has vulnerability findings from code scanning tools or plugins, developers shall seek alternative libraries or dependencies as applicable. If no viable alternative exists, developers shall coordinate with product owner and security engineer to ensure the use of such library or dependency is acceptable to continue development.
- The code shall be tested, and the application shall be deployed in the lower environment(s) prior to production. The deployment pipeline shall be configured to perform SAST, DAST, and Software Composition Analysis (SCA) using GSA OCISIO defined and approved code scanning tools with defined fail gate criteria. Security vulnerabilities and findings identified in an application or build delivery pipeline shall be remediated or granted exception(s) prior to deployment into the production environment(s). Application delivery pipeline shall generate reports for SAST, DAST, SCA, and SBOMs as security artifacts for each version of software release.
- Ensure all production and internet facing applications are enrolled into GSA vulnerability disclosure and bug bounty program.
- Teams shall establish weekly O&M tasks for ongoing review of scan results including, but not limited to SAST, DAST, Vulnerability Disclosure Program [VDP], Bug Bounty, and SCA, and create associated tasks for remediation of findings from scanners and reports on an ongoing basis.

#### 4.4 Change Management

The ODP team envisions adoption of lean changes and a release management process aligned with DevSecOps friendly agile and incremental releases models. Point-in-time assessment becomes necessary due to consistent changes in the environment. Hence, adopting a change management process which minimizes drift between security and compliance is a key requirement to prepare systems for functional ongoing authorization. System teams that adopt DevSecOps practices shall define lean change management processes with integrated security impact analysis processes associated with changes.

Listed below are practices system teams that adopt DevSecOps practices should follow for change and release management:

- Feature enhancements, new capabilities and architectural enhancements requested by stakeholders including business partners, product owners, program managers, system engineers, architects, or security personnel shall be tracked on a single project and/or task management tool (e.g., JIRA, Trello, GitHub, GitLab) used by the team.
- Each task shall be groomed with clear requirements and should be defined with clear scope of work and acceptance criteria. Tasks shall be reviewed by the product owner and scrum master to assign priority.
- A process shall be developed to perform limited-scope security and compliance impact analysis for each task or group of tasks. Outcome of the security impact analysis shall dictate associated security and compliance work such as engagement with security engineers, selection of security tooling and process integration, assessments, and delta assessments, SSPP updates and other compliance document updates etc. Team consensus is necessary for the result of the security impact analysis. If there is difference in opinion, ISSM shall make the final determination.



- Stakeholders such as Product Owners, Scrum Masters, Assigned Engineers, Security Engineers, ISSOs/ISSM shall review defined scope of task during planning sessions and provide proposed solutions with sufficient detail to perform limited-scope security impact analysis.
- When a change is ready for deployment, the change shall be deployed from lower environment(s) to production environment in a phased approach. All changes associated with a task shall be trackable to the original task in the project management tool. When change is propagated throughout all environments, tasks are considered complete.
- For non-technical changes, such as process changes, the change is considered complete when process documents, standard operating procedures (SOPs), and compliance documents are updated and signed off by all stakeholders.

## 4.5 Automation

OCISO DevSecOps Program has been collaborating with other security teams to define a roadmap, strategy, and approach for the following areas:

- Compliance as code,
- Automated generation of compliance documentation,
- Automation of control assets.

## 5 Separation of Duty

Separation of Duty (SOD) refers to “the principle that no user should be given enough privileges to misuse the system on their own.” This section provides the security practice instructions and procedure guidance for teams to achieve SOD in a DevOps/DevSecOps working model.

Traditionally, SOD has been implemented as a separation of team functions or roles (e.g., software developer and system administrator). However, system teams that practice DevOps or DevSecOps are often cross functional teams with the same personnel taking on more than one role. DevSecOps engineers, for example, may have different levels of access into production and non-production servers, code repositories, CI/CD tools and database servers.

To maintain SOD, GSA DevOps/DevSecOps teams shall:

- Adopt standard security practices for code management, code migration, release management, production changes and high level of automation.
- Adopt the standard GitOps based change management approach and high level of automation to reduce the level of manual access required on a regular basis. However, the manual access to tools, servers, databases must be maintained for emergency operational support and purposes.
- Clearly define cross functional roles, job duties, and access required associated with these roles in their system security documentation.
- Follow security practices listed in Table 5-1 of this guide and other industry standard security best practices as technology evolves for code management, code review, change review and approval and release management practices with high level of automation where possible.

## 5.1 Procedures for SOD

Procedure categories, associated [National Institute of Standards and Technology \(NIST\) Special Publication \(SP\) 800-53, Revision 5](#) “Security and Privacy Controls for Information Systems and Organizations,” controls, and associated security practices are listed in Table 5-1.

**Table 5-1. Separation of Duty in a DevOps/DevSecOps Model**

Category	NIST Control	Security Practice
Access control	AC-5	<ul style="list-style-type: none"> <li>Cross functional roles such as DevOps/DevSecOps engineer shall be defined in the System Security and Privacy Plan (SSPP) documentation with clear definition of types and level of access they would require in different system components.</li> </ul>
Secure branching and merging	AC-5 CM-3 CM-5	<ul style="list-style-type: none"> <li>Use code management and version control tools as a singular source for application build, test, and deployment. Manual and emergency changes shall be documented and applied as defined in the change management process documents.</li> <li>Configure protected branches and use them to prevent pull requests from being merged into the main branch or master branch until conditions of change review and approval are met.</li> <li>All merges shall be reviewed and approved for code changes, including code changes initiated by repository administrators. All merges to the production branch shall be reviewed and approved beforehand. Branch review, approval and merging process shall be clearly defined in the change management process documents. Responsible role and personnel for review and approval shall be defined clearly in change management process documents.</li> <li>Code change (pull request) and review approval shall be performed by separate personnel.</li> </ul>
Automate the system build, test, and deployment process	AU-2 AU-12 CM-5	<ul style="list-style-type: none"> <li>Build and deployment of code and artifacts shall be performed by using automated build, test and deployment processes and tools. Avoid manual steps and activities as much as possible.</li> <li>Event logs and audit trails defined in <a href="#">CIO-IT Security-01-08: Audit and Accountability (AU)</a> shall be generated for all builds, tests, and deployments.</li> <li>Notifications shall be generated from build, test, and deployment, and sent to responsible personnel including the ISSO(s). Example notification includes but is not limited to: result (success or failure), start time and end time and change summary of build, test, and deployment.</li> </ul>
Access control to code management and version control tools	AC-6	<ul style="list-style-type: none"> <li>Users shall be granted appropriate level of permissions on code management and version control tools (read, write, maintain, admin) based on roles and responsibilities.</li> <li>Private code repositories shall limit access to users who have the need for access to limit the potential attack surface in the event of a security breach. Different levels of access shall be granted depending on the role the user performs. Access to merge codes in the main and/or protected branch shall be limited.</li> <li>Public repositories shall limit public access to read only and allow contribution from public users. All public contributions shall go through extensive code review, approval and merging process. Access to merge codes in the main and/or protected branch shall be limited.</li> </ul>



Category	NIST Control	Security Practice
Privileged access <sup>1</sup>	AC-6 AU-12	<ul style="list-style-type: none"> <li>Build, deployment and testing tools and pipelines shall be automated. Direct privileged access to the production environment shall be limited to the greatest extent possible; When directly accessing resources, the access shall follow the documented change management process and be fully logged as defined in CIO-IT Security-01-08: Audit and Accountability (AU).</li> <li>Isolate privileged roles from non-privileged roles. If the same user performs multiple roles, they shall assume privileged roles or use privileged accounts when performing privileged functions. The default shall be a non-privileged role or account.</li> </ul>
Authentication	IA-5	<ul style="list-style-type: none"> <li>Multi-factor Authentication: Shall ensure multi-factor authentication on every user account. This is recommended but not required for public contributors in public repositories.</li> <li>SSH keys and Personal Access Tokens: If the access to code management and version control tools is done using SSH keys or personal access tokens, rotate the keys and tokens periodically as per <a href="#">CIO-IT Security-09-43: Key Management</a>.</li> </ul>
Secret Protection	SC-28	<ul style="list-style-type: none"> <li>Secrets shall be stored in an encrypted format and retrieved and decrypted only during runtime.</li> <li>Use a Secrets Management solution like Secrets Manager or Vault to manage secrets.</li> <li>Utilize a controller that manages these secrets as custom resources that can be used in a secure GitOps based workflow.</li> </ul>

## 6 Engaging with OCISO ODP

The OCISO DevSecOps Program is designed to fit the agile working model and fluid requirements. The ODP is flexible for any discussion within constraints of core security requirements and resource availability.

### 6.1 Rules of ODP Engagement

- Commitment to the operating principle as outlined on this guide and agreement between ODP program team and integrated system team.
- Commitment to integrating and using OCISO enterprise security tools when available and applicable.
- Commitment to identifying security metrics and providing reports on metrics.
- Commitment and priority towards IaC and Security as Code.
- Commitment to sharing code, process, technology, and best practices with other GSA teams.
- Commitment to communicating the level of effort needed from an integrated security engineer in terms of percentage of the work week and providing a notice of agreed upon weeks from either side to end engagement.
- Commitment to providing agreed upon funding and chargeback for tools, licenses, and labor as applicable.

---

<sup>1</sup> Privileged access is any access above user level (e.g., administrator, root, super user, power user, etc.)

## **6.2 Process for Initial ODP Engagement**

- Teams interested in an engagement with the ODP can contact the OCISO DevSecOps team via email. (Email: [ociso-devsecops@gsa.gov](mailto:ociso-devsecops@gsa.gov))
- The ODP team will review requests and schedule calls for further discussion, with a turnaround time of 3-5 business days.
- If a team meets the basic prerequisites, the ODP will schedule a kickoff meeting to discuss the details of engagement with the team.