



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

Revision 3


September 25, 2025

VERSION HISTORY/CHANGE RECORD

Change Number	Person Posting Change	Change	Reason for Change	Page Number of Change
		Initial Release – September 26, 2019		
N/A	ISE	New guide created to integrate security into DevOps teams.		N/A
		Revision 1 – September 9, 2022		
1	ISE	Updates include: <ul style="list-style-type: none"> • Minor updates in language • Removed links to old AWS checklist • Minor alignment for upcoming major update of the guide 	Periodic Update.	Throughout
2	McCormick/Klemens	<ul style="list-style-type: none"> • Editing and formatting 		Throughout
		Revision 2 – April 19, 2023		
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"> • Collaborated with DevSecOps team on restructure of guide. • Edited and updated to current guide format and structure. 		Throughout
		Revision 3 –September 25, 2025		
1	ISE/Normand/Klemens/Peralta	Updates included: <ul style="list-style-type: none"> • Revised role responsibilities and moved to an Appendix. • Refined scope to focus on cloud systems. • Updated model from an integrated engineer to a cross-functional team. • Revised Sections 3, 4.2-4.5 and 6.1-6.2. 	Align to current GSA process and guidance.	Throughout

Approval

IT Security Procedural Guide: OCISO DevSecOps Program, CIO-IT Security 19-102, Revision 3, is hereby approved for distribution.

DocuSigned by:

CA8EF810EDA7425...
Joseph Hoyt
Acting 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.

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.

Table of Contents

1	Introduction	1
1.1	Applicability and Responsibility.....	1
1.2	Purpose.....	1
1.3	Scope	1
1.4	Roles and Responsibilities.....	1
2	Defining DevSecOps	2
2.1	DevSecOps Program Goals	2
2.2	DevSecOps Capabilities	3
3	ATO and Ongoing Authorization	4
4	ODP DevSecOps Strategy	4
4.1	cross-functional Working Model.....	5
4.2	Operational Security	6
4.3	Application Security	7
4.4	Change Management	8
4.5	Automation	8
5	Separation of Duty	9
5.1	Procedures for SOD	9
6	Engaging with OCISO ODP.....	11
6.1	Rules of ODP Engagement	11
6.2	Process for Initial ODP Engagement	12
	Appendix A: Roles and Responsibilities.....	13

Tables and Figures

Table 2-1. DevSecOps Capabilities.....	3
Figure 4-1. DevSecOps Pillars	5
Table 5-1. Separation of Duty in a DevOps/DevSecOps Model.....	10

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 ODP. While the ODP is available to GSA on-prem or hybrid information systems, its primary focus is on cloud-based systems. ODP resources and services consist of:

- Security engineering consulting;
- Secure architecture design;
- Security tooling integration;
- Reusable code;
- Reference architecture;
- Enterprise security tools and shared services; and
- Onboarding enablement security engineer(s).

1.4 Roles and Responsibilities

[Appendix A](#) contains the roles and responsibilities associated with the ODP.

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. The National Institute of Standards and Technology's (NIST) overview of [DevSecOps](#) refines it as:

"DevOps brings together software development and operations to shorten development cycles, allow organizations to be agile, and maintain the pace of innovation while taking advantage of cloud-native technology and practices. Industry and government have fully embraced and are rapidly implementing these practices to develop and deploy software in operational environments, often without a full understanding and consideration of security."

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 and how we will support cloud platform engineers and customers that come to use the platforms to deliver products.

At a high level, the ODP defines DevSecOps as "integrating security into all DevOps workflows and practices." As the organization adopts the DevSecOps culture, it is dedicated to furthering the "security shift left" mission, where it 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 authorization to operate (ATO) and compliance

assessments to incorporating security considerations into every stage of the system lifecycle and educating 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 Direct Security Engineering Services For a Limited Time.** Currently, OCISO security services are primarily available during Assessment and Authorization (A&A), Incident Response (IR), and for limited Information System Security Officer (ISSO) functions. Upon request from a System Owner/Information System Security Manager (ISSM)/ISSO the ODP shall temporarily provide security engineering services to a system team during critical phases of the compliance or development lifecycle. This engagement is designed to be short-term, providing prescriptive guidance and technical services until teams become self-reliant in security-related functions. In a limited capacity, the security engineer will also support authorization-related functions as needed for the duration of the engagement.
- 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">• Code maintainability• Continuous delivery• Continuous Integration• Continuous testing• Empowered teams• Deployment Automation• Loosely Coupled Architecture	<ul style="list-style-type: none">• Customer feedback and engagements• Streamlined change approvals• Team experimentation• Work in small batches	<ul style="list-style-type: none">• Monitoring system to inform business decisions• Monitoring and observability• Proactive failure notification• Work in process limits	<ul style="list-style-type: none">• Agile decision-making• cross-functional ownership culture• Product ownership and responsibility on product team• Transformational leadership

<ul style="list-style-type: none">• Shifting security to the left• Trunk-based development• Version control		<ul style="list-style-type: none">• Visual management capabilities	
---	--	--	--

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

3 ATO and Ongoing Authorization

All GSA systems require an ATO as per [GSA Order CIO 2100.1](#), GSA Information Technology (IT) Security Policy. All systems must obtain an ATO following the existing GSA process, regardless of whether they are operated and managed by following DevSecOps principles.

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.

The ODP supports system teams in achieving and maintaining ATO/OA by providing:

- Hardened Container Images and Hardened Amazon Machine Images (AMI).
- Infrastructure as Code (IaC) modules or templates aligned with GSA baselines.
- Automated Security Scans (SAST, SCA, IaC, Container) integrated into continuous integration/continuous delivery (CI/CD) with or without fail-gates.
- Security Consulting to advise on architecture, threat modeling, and required security practices to align with GSA requirements.

By integrating these resources and expertise into delivery pipelines and operational practices, ODP helps system teams reduce manual evidence gathering, accelerate ATO preparation, and maintain a strong security posture throughout the system lifecycle.

4 ODP DevSecOps Strategy

The ODP program identifies five different pillars (see Figure 4-1) for DevSecOps program advancement and adoption: Cross Function Working Model, Operational Security, Application Security, Change Management, and Automation. The following sections provide details for each pillar.

GSA OCISO DevSecOps Program



Figure 4-1. DevSecOps Pillars

4.1 cross-functional Working Model

The ODP team envisions a cross-functional system team aligned with business lines, platform teams, application teams, and the OCISO security team. This alignment, guided by DevSecOps principles and current practices, establishes crucial communication and coordination channels, enabling rapid responses to security requirements. Product ownership remains within this single cross-functional team, with the ODP DevSecOps engineer serving as the Cyber Security Domain Experts, collaborating closely 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/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, GitHub, GitLab), code repositories, cloud-based tools, and platforms (e.g., Amazon Web Services [AWS] Cloud Console), Jump Boxes, 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, with defined responsibilities, engaging directly in operational security of the system managed and operated by the system team. Operational security shall be balanced with other priorities for enhancement and new features to avoid degradation of system and application security (AppSec) 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.
- Empower and enable ISSOs to process reports from common vulnerabilities and exposures (CVE) scanning tools.
- Coordinate with ISSOs to triage benchmark scanning reports. Critical, High and Moderate 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.
- On-board and offboard application team users in security tools following processes defined by 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.
- Upon request from an ISSM/ISSO, review system vulnerabilities and coordinate for exception approval 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.
- 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), and optionally for 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. The goal is to mitigate the risk of [software vulnerabilities by adopting secure software development](#) and trying to be in line with [NIST SP 800-218](#).

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

- Adopt agile development, breaking tasks into small, incremental units with defined and prioritized stories.
- Perform security and compliance impact analysis at appropriate levels.
- Define a standard development environment, including tools (linters, IDE plugins for static code/dependency scanning) and code management practices (trunk branch, ownership, branching/merging strategy).
- Continuously use linters and IDE plugins for scanning and remediate any findings.
- Engage with OCISO AppSec regarding Critical/High vulnerabilities.
- Use the latest version of third-party libraries, seek alternatives if vulnerabilities persist.
- Coordinate with the product owner and security engineer if no alternative exists.
- Test code and deploy applications in lower environments before production.
- Verify the deployment pipeline performs SAST, DAST (optionally), and SCA using GSA OCISO-approved tools with fail gate criteria.
- Remediate or grant exceptions for vulnerabilities before production deployment.
- Ensure the pipeline generates SAST, DAST (optionally), SCA, and SBOM reports.
- Confirm all production and internet-facing applications are enrolled in the GSA

vulnerability disclosure and bug bounty program (VDP).

- Establish weekly O&M tasks for ongoing review of scan results (SAST, DAST (optionally), VDP, Bug Bounty, SCA) and create remediation tasks.

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, 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, when applicable, 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, System Security and Privacy Plan (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, the ISSM shall make the final determination.
- Stakeholders such as Product Owners, Scrum Masters, Assigned Engineers, Security Engineers, ISSOs/ISSMs shall review defined scope of task during planning sessions and provide proposed solutions with sufficient detail to perform limited-scope security impact analysis.
- Deploy changes from lower environments to production in phases. All changes must be traceable to their original task in the project management tool. A task is considered complete when its associated changes have been propagated across all environments.
- 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

The ODP is committed to embracing automation tenets revolving around security team engagements throughout the software/systems development lifecycle, leveraging automation to streamline security processes, and fostering collaboration between development, security, and operations teams, and drawing inspiration from principles like the [12-Factor App methodology](#) and [Site Reliability Engineering](#). ODP will actively pursue opportunities to automate compliance

and policy as code, leveraging asset management or governance tools such as Archer or Prisma Cloud Compute to ensure continuous security and compliance checks. ODP will also pursue automated build and scanning of complaint infrastructure artifacts such as AWS Managed Images (AMI) and Container images.

Furthermore, ODP will drive the broader adoption of IaC. This includes targeting specific blueprints like Virtual Private Cloud (VPC)-as-a-Service (VPCaaS) and MCaaS (Managed-Cloud-as-a-Service) with the development and utilization of reusable Terraform modules for IaC.

The ODP team will also play a crucial role in facilitating the adoption of security tools designed to inspect product delivery throughout the entire development lifecycle, specifically focusing on integration with Git flow and CI/CD pipeline automation.

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-05	<ul style="list-style-type: none"> cross-functional roles such as DevOps/DevSecOps engineer shall be defined in the SSPP documentation with clear definition of types and level of access they would require in different system components.
Secure branching and merging	AC-05 CM-03 CM-05	<ul style="list-style-type: none"> 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. 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. 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. Code change (pull request) and review approval shall be performed by separate personnel.
Automate the system build, test, and deployment process	AU-02 AU-12 CM-05	<ul style="list-style-type: none"> 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. Event logs and audit trails defined in CIO-IT Security-01-08: Audit and Accountability (AU) shall be generated for all builds, tests, and deployments. 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.
Access control to code management and version control tools	AC-06	<ul style="list-style-type: none"> Users shall be granted appropriate level of permissions on code management and version control tools (read, write, maintain, admin) based on roles and responsibilities. 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. 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.

Category	NIST Control	Security Practice
Privileged access ¹	AC-06 AU-12	<ul style="list-style-type: none"> 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). 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.
Authentication	IA-05	<ul style="list-style-type: none"> Multi-factor Authentication: Shall ensure multi-factor authentication on every user account. This is recommended but not required for public contributors in public repositories. 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 CIO-IT Security-09-43: Key Management.
Secret Protection	SC-28	<ul style="list-style-type: none"> Secrets shall be stored in an encrypted format and retrieved and decrypted only during runtime. Use a Secrets Management solution like Secrets Manager or Vault to manage secrets. Utilize a controller that manages these secrets as custom resources that can be used in a secure GitOps based workflow.

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

To ensure successful collaboration, the ODP program team and integrated system teams agree to the following rules of engagement:

- Operate under agreed principles** — Both ODP and the system team will follow the operating principles and engagement framework outlined in this guide.
- Adopt OCISO-approved agency security tools** — The system team will integrate and actively use OCISO-approved security tools and services as they become available.
- Define and report metrics** — The system team will identify key security metrics and provide regular reports on those metrics.
- Use IaC and Security-as-Code** — All infrastructure and configuration changes will be implemented through IaC and Security-as-Code pipelines.

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

5. **Share knowledge and practices** — The system team will share lessons learned, recommended practices, and reusable approaches with ODP so they can be incorporated into templates, workflows, or guidance for other GSA teams.
6. **ODP provides tooling support, not embedded engineers** — System teams are responsible for being the primary engineers and integrators. ODP will support by delivering templates, workflows, and baseline criteria to help teams adopt and customize security practices, but ownership of implementation rests with the system team.

6.2 Process for Initial ODP Engagement

Teams interested in an engagement with the ODP should contact the OCISO DevSecOps team via email (ocisodevsecops@gsa.gov) with the following information:

- System Owner and ISSO/ISSM points of contact;
- Hosting/platform details (e.g., AWS account(s), VPCs, environments);
- Repository URLs (app + IaC), CI/CD summary, and current scanning tools
- ATO status (initial/renewal/OA), latest Plan of Action & Milestones (POA&M); and
- Inventory of internet-exposed assets (domains/URLs, ingress, Web Application Firewall [WAF]/Content Delivery Network [CDN]).

The ODP team will review requests and schedule calls for further discussion. During this time, the following activities will occur:

Step 1: Intake

- The DevSecOps team will provide an ODP Intake Form which must be completed and sent to the program mailbox for further review.
- ODP Team reviews for fit and assigns an DevSecOps/OA engineer with an agreed time allocation, with a turnaround time of 3-5 business days.

Step 2: Discovery & Maturity Baseline

- The ODP Team will work with the integrated system team on finding agreed upon time slot(s) as needed for discovery of their environment and an assessment of their DevSecOps maturity.

Step 3: 30/60/90 Plan & Success Metrics

- The ODP Team will work with the integrated system team on defining a plan for 30 day, 60 day, and 90 day milestones along with success metrics to be tracked during the course of an engagement, which will be shared with the integrated system team stakeholders on a quarterly basis.

Step 4: Working Model

- The ODP Team works with the integrated system team on supporting continuous releases, upgrades, and changes while fully maintaining security posture, principle, and compliances of the application/system.

Step 5: Sustainment & Exit

- The ODP Team will support ongoing authorization related functions and provide technical services for ODP security-related functions.

Appendix A: 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.

ODP DevSecOps Engineer

The priority of the DevSecOps Engineer is security, focusing on security design, operational 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.
- For new GSA systems or those undergoing substantial architectural modifications, the DevSecOps team will collaborate with system-integrated teams through either a limited-time engagement or on an as-needed basis. The availability of these engagement patterns will be directly contingent on the ODP team's current capacity at the time of the request. This ensures that resources are allocated efficiently and that the DevSecOps team can provide focused support where and when it is most impactful, maintaining operational fluidity while systems are integrated or updated.
- 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.
- Provides support, code, and consultation for integration with security tools and services.
- Works with the System Owner, ISSM and ISSO to provide guidance when creating and maintaining compliance documentation, such as the SSPP and POA&M. The ODP staff offers support only and does not author these documents.
- 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).
- Coordinates with the system team to define criteria to integrate the system, establish fail gate, and enforce fail gate.

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.
- Defines 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.
- Creates 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.
- Builds repetitive decision-making processes and guidelines to empower security engineers.

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.

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).
- Takes full responsibility and ownership of the system, related decisions, and outcomes.

Chief Information Security Officer (CISO)

The 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 ATO sign-off for integrated DevSecOps team systems/applications, as per existing policy and procedural guidelines.