GENERAL SERVICES ADMINISTRATION
Washington, DC 20405

CIO IL-22-01
March 10, 2022

GSA INSTRUCTIONAL LETTER

SUBJECT: Separation of Duties in a DevOps/DevSecOps Model

1. Purpose. This instructional letter (IL) is to provide the security practice instructions and procedure guidance for teams to achieve Separation of Duty (SOD) in a Development Operations/Development Security Operations (DevOps/DevSecOps) working model.

2. Background. The SOD refers to “the principle that no user should be given enough privileges to misuse the system on their own”. “Separation of duties can be enforced either statically (by defining conflicting roles, i.e., roles which cannot be executed by the same user) or dynamically (by enforcing the control at access time).” [1]

Devops/DevSecOps is “a culture shift in the software industry that aims to bake security into the rapid-release cycles that are typical of modern application development and deployment, also known as the devops movement. Embracing this shift-left mentality requires organizations to bridge the gap that usually exists between development and security teams to the point where many of the security processes are automated and handled by the development team itself.” [2]

In a traditional development life cycle, the SOD is often implemented statically as a separation of team functions, such as between development and operation teams, or a separation of individual roles, such as software developers and operation administrators. However, in a modern software development environment leveraging DevOps, DevSecOps or agile management approach, the underlying principle is to break down silos between business line, development, operation and security teams and personnel to increase agility, operational efficiency and bake security into a rapid release cycle. DevSecOps/DevOps teams are often cross functional teams with the same personnel taking developer, operational support, production support, database support, application security, data administrator and application/system architect roles. These cross functional roles are generally called DevSecOps or DevOps engineers.

[1] NIST SP 800-192, Verification and Test Methods for Access Control Policies/Models.
[2] CSO Online DevSecOps Definition.

DevOps/DevSecOps team shall adopt standard security practices for code management, code migration, release management, production changes and high level of automation. Adopting the standard GitOps[3] based change management approach and high level of automation will reduce the level of manual access required on a regular basis. However, the manual access to tools, servers, databases has to be maintained for emergency operational support and purposes.

Due to the nature of their cross functional job roles and duties, it is an acceptable practice for DevOps or DevSecOps engineers to have different levels of access into production and non-production servers, code repository, continuous integration/continuous delivery tools and database servers.

DevOps/DevSecOps team shall clearly define cross functional DevOps/DevSecOps roles, job duties and access required associated with these roles in their system security documentation. DevOps/DevSecOps team shall follow security practices listed below and other industry standard security best practices as technology evolves for their code management, code review, change review/approval and release management practices with high level of automation where possible.

3. Applicability. This IL applies to all GSA Federal employees, contractors, and vendors of GSA, who manage, maintain, operate, procure, or protect GSA systems and data, as well as all GSA Office of the Chief Information Officer (GSA IT) systems, and any GSA data contained on, or processed by, IT systems owned and operated by, or on behalf of, any GSA Service or Staff Office.

4. Effective Date. This IL is effective from the date of signature. Applicable teams as identified in Section 3 and Section 7 have 12 months to be compliant with the instructions and procedural guidance identified in this IL.

5. Policy. DevOps/DevSecOps teams must follow the procedures/instructions in Section 6, as well as applicable GSA IT Security Policies and Procedural Guides.

6. Procedures. After each procedure in this section, any associated National Institute of Technology and Standards (NIST) Special Publication (SP) 800-53, Revision 5 security control identifiers are listed in parentheses.

a. Cross functional roles such as DevOps/DevSecOps engineer shall be defined in the System Security and Privacy Plan documentation with clear definition of types and level of access they would require in different system components. (AC-5)

b. Secure branching and merging (AC-5, CM-3, CM-5)

[3] GitOps is an operational framework that takes DevOps best practices used for application development such as version control, collaboration, compliance, and continuous integration/continuous delivery (CI/CD), and applies them to infrastructure automation.

(1) 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.

(2) Configure protected branches and use protected branches to prevent pull requests from being merged into the main branch or master branch until conditions of change review and approval are met.

(3) 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/personnel for review and approval shall be defined clearly in change management process documents.

(4) Code change (pull request) and review approval shall be performed by separate personnel.

c. Automate the system build, test and deployment process. (AU-2, AU-12, CM-5)

(1) 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.

(2) Event logs and audit trails defined in IT Security Procedural Guide: Audit and Accountability (AU) CIO-IT Security-01-08 shall be generated for all builds, tests and deployments.

(3) Notifications shall be generated from build, test and deployment, and sent to responsible personnel including the Information Security Security Officers (ISSOs). Example notification includes, but is not limited to success/failure build job, starting/ending of job, test success/failure, deployment success/failure, change summary.

d. Access control to code management and version control tools (AC-6)

(1) Users shall be granted appropriate level of permissions on code management and version control tools (read vs write vs maintain vs admin) based on roles and responsibilities.

(2) 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.

(3) 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.

e. Privileged access[4] (AC-6, AU-12)

(1) Use automated build, deployment and testing tools/pipelines. 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 defined in IT Security Procedural Guide: Audit and Accountability (AU) CIO-IT Security-01-08.

(2) Isolate privilege roles from non-privileged roles. If the same user performs multiple roles, they shall assume privilege role/account when performing privilege function and default shall be non-privilege role/account.

f. Authentication (IA-5)

(1) Multi-factor Authentication: Ensure 2-factor-authentication on every user account. This is recommended but not required for public contributors in public repositories.

(2) Rotate SSH keys and Personal Access Tokens: If the access to code management and version control tools is done using SSH keys or personal user tokens, rotate the keys and tokens periodically as per Key Management CIO-IT Security-09-43

g. Protect Secrets (SC-28)

(1) Store secrets in an encrypted format that can only be accessed/decrypted during runtime.

(2) Use a Secrets Management solution like Secrets Manager or Vault to store and manage the secrets.

(3) Utilize a controller that manages these secrets as custom resources that can be used in a secure GitOps based workflow.

7. Responsibilities. All federal and contract teams supporting GSA information systems operating under a DevOps or DevSecOps model have the responsibility to adhere to the requirements stated in Section 6. The product and/or system owner and authorizing official (AO) has primary responsibility to ensure the requirements are met. In accordance with General Services Acquisition Regulation (GSAR) part 511.171,

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

Contracting Officers must include compliance with this policy in the contract or task order for contractor employees.

8. Signature.

/S/____________________

DAVID SHIVE
Chief Information Officer
Office of GSA IT