13_seguridad

May 16, 2023

1 Seguridad.

Existen varias recomendaciones en materia de seguridad para poder utilizar *Github Actions* de forma adecuada.

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions

1.1 Uso de secretos.

Los valores sensibles jamás deben almacenarse como texto plano en archivos de flujo de trabajo, sino más bien como secretos. Los secretos se pueden configurar en el nivel de organización, repositorio o entorno, y permiten almacenar información confidencial en *GitHub*.

Para mayor información, consultar:

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions#using-secrets

1.2 Uso de `CODEOWNERS` para supervisar los cambios.

Es posible aprovehcar la característica `CODEOWNERS` para controlar cómo se realizan los cambios en los archivos de los *workflows*.

Por ejemplo, si todos los archivos de *workflow* se almacenan en `.github/workflows`, es posible agregar este directorio a la lista de propietarios de código para que cualquier cambio propuesto a estos archivos requiera primero de una aprobación de un revisor designado.

Para mayor información, consultar:

- https://docs.github.com/es/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-code-owners

1.3 Entender el riesgo de las inyecciones de código.

Cuando se crean *workflows*, acciones personalizadas y acciones compuestas, siempre debe plantearse si el código podría ejecutar entradas no fiables creadas por algún atacantes.

Esto puede ocurrir cuando un atacante agrega comandos y *scripts* malintencionados a un contexto. Cuando un *workflow* se ejecuta, estas secuencias podrían interpretarse como código que luego se ejecutará en el *runner*.

Para mayor información, consultar:

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions#understanding-the-risk-of-script-injections

1.4 Utilizar `OpenID connect` para acceder a los recursos en la nube .

Si los *workflows* de *GitHub Actions* necesitan acceder a los recursos de un proveedor de servicios en la red que sea compatible con *OpenID Connect* (*OIDC*), es posible configurarlos para que se autentiquen directamente con dicho proveedor. Esto permitirá dejar de almacenar estas credenciales como secretos de duración larga y proporcionará otros beneficios de seguridad.

Para mayor información, consultar:

- https://docs.github.com/es/actions/deployment/security-hardening-your-deployments/about-security-hardening-with-openid-connect

1.5 Ser cuidadoso con *actions* de terceros.

Los *jobs* individuales en un *workflow* pueden interactuar con (y ponerse enriesgo con) otros *jobs*.

Por ejemplo:

- Un *job* que consulta las variables de entorno que se utilizan por otro *job* subsecuente.
- Escribir archivos en un directorio compartido que el *job* subsecuente proces.
- Si un *job* interactúa con el conector de *Docker* e inspecciona a otros contenedores en ejecución y ejecuta comandos en ellos.

Para mayor información, consultar:

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions#using-third-party-actions

1.6 El uso de *Dependabot version updates*.

Es posible usar *Dependabot version updates* para asegurartse de que las referencias a las acciones y los *workflows* reutilizables usados en el repositorio se mantienen actualizados. Las acciones a menudo se actualizan con correcciones de errores y con nuevas características para que los procesos automatizados sean más confiables, rápidos y seguros. *Dependabot version updates* acaba con la necesidad de mantener las dependencias, ya que *Dependabot* lo hace automáticamente.

Para mayor información, consultar:

- https://docs.github.com/es/code-security/dependabot/working-with-dependabot/keeping-your-actions-up-to-date-with-dependabot

1.7 Impedir que *GitHub Actions* cree o apruebe *pull requests*.

Es posible restingir la creación o aprobación de *pull requests* desde un *workflow*.

Permitir que los *workflows*, o cualquier otra automatización, creen o aprueben *pull requests* podría ser un riesgo de seguridad sin la supervisión adecuada.

Para mayor información, consultar:

- https://docs.github.com/es/github/setting-up-and-managing-organizations-and-teams/disabling-or-limiting-github-actions-for-your-organization#preventing-github-actions-from-creating-or-approving-pull-requests
- https://docs.github.com/es/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#preventing-github-actions-from-creating-or-approving-pull-requests

1.8 Considerar el acceso entre repositorios.

GitHub Actions tiene fue diseñuado para gestionar el ámbito de un solo repositorio a la vez. `GITHUB_TOKEN` concede el mismo nivel de acceso que el de un usuario con acceso de escritura, ya que cualquier usuario con este tipo de acceso puede acceder a este *token* creando o modificando un archivo de *workflow*, lo que eleva los permisos de `GITHUB_TOKEN` en caso de ser necesario.

Los usuarios tienen permisos específicos para cada repositorio, por lo que permitir que `GITHUB_TOKEN` para un repositorio otorgue acceso a otro afectaría al modelo de permisos de *GitHub* si no se implementa con cuidado. De forma similar, se debe tener cuidado al agregar *tokens* de autenticación de *GitHub* a un flujo de trabajo, ya que esto también puede afectar el modelo de permisos de *GitHub* al otorgar inadvertidamente un acceso amplio a los colaboradores.

Para mayor información consultar:

- https://docs.github.com/en/actions/security-guides/security-hardening-for-github-actions#considering-cross-repository-access

1.9 Revisar de la cadena de suministro para *runners* alojados en *GitHub*.

Es posible consultar una lista de materiales de software (*SBOM*) a fin de comprobar qué software se ha instalado previamente en la imagen del *runner* hospedado en *GitHub* usada durante las ejecuciones de *workflows*.

Es posible proporcionar a los usuarios la *SBOM*, que pueden ejecutar con un analizador de vulnerabilidades para validar si hay alguna vulnerabilidad en el producto. Si se pretende crear artefactos, es posible incluir esta *SBOM* en la lista de materiales para obtener una lista completa de todo lo que se ha utlizado en la creación del software.

Las SBOM están disponibles para imágenes de los *runners* de *Ubuntu*, *Windows* y *MacOS*. Es posible consultar la *SBOM* de la compilación en los recursos de versión en https://github.com/actions/runner-images/releases. Una *SBOM* con un nombre de archivo en formato `sbom.<IMAGE-NAME>.json.zip` se puede encontrar en los datos adjuntos de cada versión.

1.10 Fortalecimiento para los *runners* auto-hospedados.

Los *runners* hospedados en *GitHub* ejecutan código en máquinas virtuales aisladas, limpias y efímeras, lo que significa que no hay forma de poner este entorno en riesgo de forma persistente, o de obtener acceso de otra forma a más información de la que se ha colocado en este entorno durante el proceso de arranque.

Los *runners* auto-hospedados no tienen garantías sobre ejecutar en máquinas virtuales limpias y efímeras, y pueden ponerse en riesgo de forma persistente mediante el código no fiable en un flujo de trabajo.

Para mayor información consultar:

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions#hardening-for-self-hosted-runners

1.11 Auditar eventos de *GitHub Actions.*

Es posible utilizar la bitácora de auditoría para monitorear las tareas administrativas en una organización. El registro de auditoría registra el tipo de acción, cuándo se ejecutó, y qué cuenta personal llevó a cabo la acción.

Para mayor información consultar:

- https://docs.github.com/es/actions/security-guides/security-hardening-for-github-actions#auditar-eventos-de-github-actions
- https://docs.github.com/es/organizations/keeping-your-organization-secure/managing-security-settings-for-your-organization/reviewing-the-audit-log-for-your-organization