

[New issue](#)[Jump to bottom](#)

# Insecure path traversal in Git Trigger Source can lead to arbitrary file read #1947

🔒 Closed

whynowy opened this issue on May 10 · 0 comments · Fixed by #1965

Labels

security

whynowy commented on May 10 • edited ▼

Member

A path traversal issue was found in the (g \*GitArtifactReader).Read() API. Read() calls into (g \*GitArtifactReader).readFromRepository() that opens and reads the file that contains the trigger resource definition:

<https://github.com/argoproj/argo-events/blob/master/sensors/artifacts/git.go>

```
func (g *GitArtifactReader) readFromRepository(r *git.Repository, dir string)

...

    if err := w.Pull(pullOpts); err != nil && err != git.NoErrAlreadyUpToDate {
        return nil, fmt.Errorf("failed to pull latest updates. err: %v", err)
    }

    return ioutil.ReadFile(fmt.Sprintf("%s/%s", dir, g.artifact.FilePath))
}
```

No checks are made on this file at read time, which could lead an attacker to read files anywhere on the system. This could be achieved in at least three ways:

Symbolic link in Git repository

An attacker controls a Git repository that the victim uses in a Git Trigger Source. The attacker adds a file to the Git repository that is a symbolic link to a file containing sensitive information on the victims machine.

Argo then clones the repository onto the victims machine, and the symbolic link is followed during file read on the marked line above. An attacker could now read the file containing sensitive information.

## Race condition

An attacker who has limited access to the file system may be able to read arbitrary files by leveraging a race condition. The attacker could replace the git-temp directory created by argo with a symbolic link to the directory containing the file to be read. This could be done anytime between the time it is created in (g \*GitArtifactReader).Read() and the file is read in the return statement of (g \*GitArtifactReader).readFromRepository(r \*git.Repository, dir string).

## Malicious manifest

An attacker controls a manifest for a Git Trigger Source that the victim creates.

The manifest has a filePath to a sensitive file anywhere on the victims machine, for example:

```
triggers:
  - template:
      name: workflow-trigger
      k8s:
        operation: create
        source:
          git:
            url: "git@github.com:argoproj/argo-workflows.git"
            cloneDirectory: "/git/argoproj"
            sshKeyPath: "/secret/key"
            namespace: argo-events
            filePath: "/path/to/sensitive/file"
            branch: "master"
```

## Recommendations

### Disallow symbolic links

Check whether the file at GitArtifactReader.artifact.FilePath is a symbolic link before it is opened and read in (g \*GitArtifactReader).readFromRepository(). Fail if it is.

### Sanitize GitArtifactReader.artifact.FilePath

This includes checks for unsafe path patterns, such as:

Check whether the string begins with "/".

Disallow "..", "\", "~" in path.

Other checks to ensure that only the files from the Git repository can be read

  **whynoway** added the `security` label on May 10

 This was referenced on May 10

## Security Audit #1943

✓ Closed

**fix: git artifactory arbitrary file read issue #1965**

 Merged

 **whynow** closed this as completed in [#1965](#) on May 12

---

  **jba** mentioned this issue on Jun 17

**x/vulndb: potential Go vuln in Path is unknown: CVE-2022-25856** [jba/nested-modules#353](#)

 Open

  **GoVulnBot** mentioned this issue on Jun 17

**x/vulndb: potential Go vuln in github.com/argoproj/argo-events/sensors/artifacts: CVE-2022-25856** [golang/vulndb#492](#)

 Closed

#### Assignees

No one assigned

---

#### Labels

security

---

#### Projects

None yet

---


#### Milestone

No milestone

---

#### Development

Successfully merging a pull request may close this issue.

 **fix: git artifactory arbitrary file read issue**  
[whynow/argo-events](#)

---

1 participant

