Partial Path Traversal in com.amazonaws:aws-java-sdk-s3

High millems published GHSA-c28r-hw5m-5gv3 on Jul 15

Package

com.amazonaws:aws-java-sdk-s3 (Maven)

Affected versions	Patched versions
<= 1.12.260	>= 1.12.261

Description

Overview

A partial-path traversal issue exists within the `downloadDirectory` method in the AWS S3 TransferManager component of the AWS SDK for Java v1. Applications using the SDK control the `destinationDirectory` argument, but S3 object keys are determined by the application that uploaded the objects. The `downloadDirectory` method allows the caller to pass a filesystem object in the object key but contained an issue in the validation logic for the key name. A knowledgeable actor could bypass the validation logic by including a UNIX double-dot in the bucket key. Under certain conditions, this could permit them to retrieve a directory from their S3 bucket that is one level up in the filesystem from their working directory.

This issue's scope is limited to directories whose name prefix matches the destinationDirectory. E.g. for destination directory `/tmp/foo`, the actor can cause a download to `/tmp/foo-bar`, but not `/tmp/bar`.

Versions of the AWS Java SDK for S3 v1 before and including v1.12.260 are affected by this issue.

Impact

If `com.amazonaws.services.s3.transfer.TransferManager::downloadDirectory` is used to download an untrusted buckets contents, the contents of that bucket can be written outside of the intended destination directory.

Root Cause

The `com.amazonaws.services.s3.transfer.TransferManager::downloadDirectory` contains a partial-path traversal vulnerability.

This is due to the guard logic in `leavesRoot` containing an insufficient protection against partial-path traversal.

aws-sdk-java/aws-java-sdk-s3/src/main/java/com/amazonaws/services/s3/transfer/TransferManager.java
Lines 1513 to 1519 in `5be0807`

```
1513      private boolean leavesRoot(File localBaseDirectory, String key) {
1514          try {
1515              return !new File(localBaseDirectory, key).getCanonicalPath().startsWith(local
1516          } catch (IOException e) {
1517              throw new RuntimeException("Unable to canonicalize paths",  e);
1518          }
1519      }
```

The application controls the `localBaseDirectory` argument, but the `key` comes from the AWS bucket entry (ie. can be attacker controlled). The above bit of logic can be bypassed with the following payloads:

```
// The following will return 'false', although the attacker value will "leave" the `/usr/foo` di
leavesRoot(new File("/usr/foo"), "/../foo-bar/bar")
```

This guard is used here which should guard against path traversal, however `leavesRoot` is an insufficient guard:

aws-sdk-java/aws-java-sdk-s3/src/main/java/com/amazonaws/services/s3/transfer/TransferManager.java
Lines 1420 to 1423 in `ae88c8a`

```
1420      if ( leavesRoot(destinationDirectory, s.getKey()) ) {
1421          throw new RuntimeException("Cannot download key " + s.getKey() +
1422              ", its relative path resolves outside the parent directory.");
1423      }
```

True Root cause

If the result of `parent.getCanonicalPath()` is not slash terminated it allows for partial path traversal.

> Consider `"/usr/outnot".startsWith("/usr/out")`. The check is bypassed although `outnot` is not under the `out` directory.
> The terminating slash may be removed in various places. On Linux `println(new File("/var/"))` returns `/var`, but `println(new File("/var", "/")) - /var/`, however `println(new File("/var", "/").getCanonicalPath()) - /var`.
> - @JarLob (Jaroslav Lobačevski)

Patches

Upgrade to the AWS SDK for Java >= 1.12.261, if you are on a version < 1.12.261.

Workarounds

When calling `com.amazonaws.services.s3.transfer.TransferManager::downloadDirectory` pass a `KeyFilter` that forbids `S3ObjectSummary` objects that `getKey` method return a string containing the substring `..` .

References

Similar vulnerabilities:

- ESAPI (The OWASP Enterprise Security API) - https://nvd.nist.gov/vuln/detail/CVE-2022-23457

For more information

If you have any questions or comments about this advisory, please contact AWS's Security team.

Severity

High 7.9 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	High
Privileges required	Low
User interaction	Required
Scope	Changed
Confidentiality	High
Integrity	High
Availability	Low

CVSS:3.1/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:L

CVE ID

CVE-2022-31159

Weaknesses

CWE-22

Credits

JLLeitschuh