Repository content filters do not work in Settings pluginManagement

High **big-guy** published **GHSA-jvmj-rh6q-x395** on Apr 9, 2021

Package

Gradle (Java)

Affected versions	Patched versions
5.1 to 6.8.3	7.0

Description

We would like to thank **@ZacSweers** from **@slackhq** for making us aware of this security issue.

Impact

Repository content filtering is a security control Gradle introduced to help users specify what repositories are used to resolve specific dependencies. This feature was introduced in the wake of the "A Confusing Dependency" blog post.

In some cases, Gradle may ignore content filters and search all repositories for dependencies. This only occurs when repository content filtering is used from within a `pluginManagement` block in a settings file. This may change how dependencies are resolved for Gradle plugins and build scripts.

For builds that are vulnerable, there are two risks:

- Information disclosure: Gradle could make dependency requests to repositories outside your organization and leak internal package identifiers.
- Dependency poisoning/Dependency confusion: Gradle could download a malicious binary from a repository outside your organization due to name squatting.

Example

These examples demonstrate vulnerable builds. You would find this usually inside of a settings.gradle or settings.gradle.kts file.

Using an exclusive content filter:

```
pluginManagement {
    repositories {
        exclusiveContent {
            forRepository {
                maven {
                    name = "JCenter"
                    setUrl("https://jcenter.bintray.com/")
                }
            }
            filter {
                includeModule("org.jetbrains.kotlinx", "kotlinx-html-jvm")
                includeGroup("org.jetbrains.dokka")
                includeModule("org.jetbrains", "markdown")
            }
        }
        mavenCentral()
        gradlePluginPortal()
    }
}
```

Using an content filter on the repository itself:

```
pluginManagement {
  repositories {
    jcenter {
      content {
        includeModule("org.jetbrains.kotlinx", "kotlinx-html-jvm")
        includeGroup("org.jetbrains.dokka")
        includeModule("org.jetbrains", "markdown")
      }
    }
    mavenCentral()
    gradlePluginPortal()
  }
}
```

In both examples, the modules "org.jetbrains.kotlinx:kotlinx-html-jvm" and "org.jetbrains:markdown" and all modules from group "org.jetbrains.dokka" should be resolved from JCenter. Due to the bug/vulnerability, Gradle will attempt to resolve the dependencies from all repositories.

What should you do?

Upgrade to Gradle 7.0

The problem has been patched and released with Gradle 7.0.
Users relying on this feature should upgrade their build as soon as possible.

Workaround for older versions

- Use a company repository which has the right rules for fetching packages from public repositories.
- Use project level repository content filtering, inside `buildscript.repositories`. This option is available since Gradle 5.1 when the feature was introduced.

References

- CWE-829: Inclusion of Functionality from Untrusted Control Sphere
- A Confusing Dependency

- Dependency Confusion

Questions?

- For security related issues, please email us at security@gradle.com.
- For non-security related issues, please open an issue on GitHub.

Severity

High 8.0 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	High
Privileges required	High
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	High

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

CVE ID

CVE-2021-29427

Weaknesses

CWE-829

Credits

 ZacSweers