Diversas bibliotecas públicas y bien conocidas, previamente desatendidas pero todavía en uso en aplicaciones Java y Android, han demostrado ser susceptibles a un novedoso método de ataque a la cadena de suministro de software denominado MavenGate.

> Oversecured señaló en un análisis publicado la semana pasada que *«el acceso a proyectos puede ser secuestrado a través de la compra de nombres de dominio y, dado que la mayoría de las configuraciones de construcción por defecto son vulnerables, sería complicado o incluso imposible detectar si se está llevando a cabo un ataque».*

Explotar con éxito estas debilidades podría permitir a actores maliciosos tomar el control de artefactos en dependencias e insertar código malicioso en la aplicación, e incluso comprometer el proceso de construcción mediante un plugin malicioso.

La empresa de seguridad móvil añadió que todas las tecnologías basadas en Maven, incluyendo Gradle, son vulnerables al ataque, y que informó a más de 200 empresas, incluyendo Google, Facebook, Signal, Amazon y otras.

Apache Maven se utiliza principalmente para construir y gestionar proyectos basados en Java, permitiendo a los usuarios descargar y gestionar dependencias (identificadas de manera única por sus groupIds), crear documentación y gestionar lanzamientos.

Aunque los repositorios que albergan tales dependencias pueden ser privados o públicos, un atacante podría apuntar a los públicos para llevar a cabo ataques de envenenamiento de la cadena de suministro mediante el aprovechamiento de bibliotecas desatendidas añadidas a repositorios conocidos.

Concretamente, esto implica la compra del dominio inverso expirado controlado por el propietario de la dependencia y obtener acceso al groupId.

> *«Un atacante puede obtener acceso a un groupId vulnerable afirmando sus derechos sobre él a través de un registro DNS TXT en un repositorio donde no exista una cuenta que gestione el groupId vulnerable»,* explicó la compañía.
>
> *«Si un groupId ya está registrado en el repositorio, un atacante puede intentar obtener acceso a ese groupId poniéndose en contacto con el equipo de soporte del repositorio».*

Para poner a prueba el escenario de ataque, Oversecured subió su propia biblioteca de prueba para Android (groupId: «com.oversecured»), que muestra el mensaje de «Hello World!» en Maven Central (versión 1.0), mientras también subía dos versiones a JitPack, donde la versión 1.0 es una réplica de la misma biblioteca publicada en Maven Central.

Pero la versión 1.1 es una copia editada *«no confiable»* que también tiene el mismo groupId, pero apunta a un repositorio de GitHub bajo su control y se reclama añadiendo un registro DNS TXT para hacer referencia al nombre de usuario de GitHub con el fin de establecer la prueba de propiedad.

El ataque funciona añadiendo tanto Maven Central como JitPack a la lista de repositorios de dependencias en el script de construcción de Gradle. Es importante señalar en esta etapa que el orden de declaración determina cómo Gradle verificará las dependencias en tiempo de ejecución.

> *«Cuando movimos el repositorio JitPack por encima de mavenCentral, la versión 1.0 se descargó desde JitPack. Cambiar la versión de la biblioteca a 1.1 resultó en el uso de la versión de JitPack independientemente de la posición de JitPack en la lista de repositorios»,* detallaron los investigadores.

Como resultado, un adversario que busca corromper la cadena de suministro de software

puede dirigirse a versiones existentes de una biblioteca publicando una versión superior o contra nuevas versiones empujando una versión que es inferior a la de su contraparte legítima.

Esto representa otra variante de un ataque de confusión de dependencias, donde un atacante publica un paquete falso en un repositorio de paquetes público con el mismo nombre que un paquete dentro del repositorio privado previsto.

> *«La mayoría de las aplicaciones no verifican la firma digital de las dependencias, y muchas bibliotecas ni siquiera la publican. Si el atacante quiere permanecer indetectado durante el mayor tiempo posible, tiene sentido lanzar una nueva versión de la biblioteca con el código malicioso incrustado y esperar a que el desarrollador la actualice»,* añadieron los investigadores.

De los 33,938 dominios totales analizados, se encontró que 6,170 (18.18%) eran vulnerables a MavenGate, lo que permite a los actores de amenazas secuestrar las dependencias e inyectar su propio código.

Sonatype, propietaria de Maven Central, dijo que la estrategia de ataque descrita *«no es factible debido a la automatización en marcha»,* pero señaló que ha *«deshabilitado todas las cuentas asociadas con dominios caducados y proyectos de GitHub»* como medida de seguridad.

También informó que abordó una *«regresión en la validación de la clave pública»* que permitía subir artefactos al repositorio con una clave no compartida públicamente. También anunció planes de colaborar con SigStore para firmar digitalmente los componentes.

> *«El desarrollador final es responsable de la seguridad no solo de las dependencias directas, sino también de las dependencias transitivas»,* dijo Oversecured.

> *«Los desarrolladores de bibliotecas deben ser responsables de las dependencias que declaran y también escribir hashes de clave pública para sus dependencias, mientras que el desarrollador final solo debe ser responsable de sus dependencias directas».*