Los delincuentes informáticos están empleando GitHub de manera cada vez más frecuente para propósitos ilícitos mediante estrategias innovadoras, como la manipulación de Gists confidenciales y la emisión de órdenes dañinas a través de mensajes de confirmación en git.

> Karlo Zanki, investigador de ReversingLabs, señaló en un informe: *«De vez en cuando, los creadores de malware dejan sus ejemplares en plataformas como Dropbox, Google Drive, OneDrive y Discord para alojar software malicioso y evitar herramientas de identificación».*

> *«Sin embargo, recientemente, hemos notado un incremento en el uso de GitHub para albergar malware».*

Se sabe que servicios públicos legítimos son cooptados por delincuentes para guardar software malicioso y actuar como intermediarios para descubrir la verdadera dirección de control y comando (C2).

A pesar de que emplear recursos públicos para C2 no los exime de posibles intervenciones, sí facilita a los atacantes configurar una infraestructura de ataque económica y robusta.

Este método es astuto, permitiendo a los delincuentes disfrazar su tráfico maligno con comunicaciones auténticas en una red comprometida, complicando su detección y respuesta adecuada. Por ello, es improbable que un dispositivo infectado que se conecte con un repositorio de GitHub sea identificado como sospechoso.

El uso indebido de gists en GitHub indica una progresión en esta táctica. Estos gists, en esencia repositorios, proporcionan una vía sencilla para que programadores compartan fragmentos de código.

Es relevante mencionar que los gists públicos aparecen en el feed discover de GitHub,

mientras que los gists secretos, aunque ocultos del descubrimiento, pueden ser compartidos por su URL.

> *«Si alguien desconocido descubre el enlace, podrá ver el gist. Para proteger tu código, sería conveniente optar por un repositorio privado»*, aclara GitHub en sus directrices.

Un detalle interesante es que los gists secretos no figuran en la página de perfil de GitHub del autor, permitiendo a los delincuentes utilizarlos como un servicio de almacenamiento temporal.

ReversingLabs identificó varios paquetes en PyPI, incluyendo httprequesthub, pyhttpproxifier, libsock, libproxy y libsocks5, que se presentaban como herramientas para gestionar proxies de red, pero que en realidad contenían una URL cifrada apuntando a un gist confidencial alojado en una cuenta de GitHub temporal y sin proyectos públicos.

Este gist incluye comandos encriptados que son procesados a través de código malicioso en el archivo setup.py de estos paquetes.

El empleo de gists confidenciales para enviar órdenes maliciosas a equipos comprometidos fue previamente destacado por Trend Micro en 2019, vinculado a una campaña que distribuía un troyano llamado SLUB.

Una técnica adicional identificada por expertos en seguridad informática implica usar características del sistema de control de versiones, aprovechando los mensajes de confirmación para ejecutar comandos en el sistema.

El paquete de PyPI, easyhttprequest, incluye código que *«clona un repositorio específico de GitHub y verifica si el 'commit' principal tiene un mensaje que inicia con una secuencia particular»,* comentó Zanki.

«Si es así, elimina ese segmento y descifra el resto del mensaje, ejecutándolo como un comando Python en un proceso nuevo». El repositorio de GitHub clonado es una bifurcación de un proyecto PySocks legítimo, sin mensajes de confirmación maliciosos.

Todos los paquetes fraudulentos han sido retirados del repositorio PyPI.

> *«Si bien usar GitHub para C2 no es nuevo, el aprovechamiento de características como Git Gists y mensajes de confirmación para distribuir comandos representa tácticas innovadoras de los ciberdelincuentes»,* concluyó Zanki.