Framework de intercepción de tráfico de aplicaciones móviles en Android y iOS

Mobile application traffic interception framework on Android and iOS

Presentación: 13 y 14 setiembre de 2023

Fabián Gibellini
Universidad Tecnológica Nacional – Facultad Regional Córdoba
fabiangibellini@gmail.com

Leonardo Ciceri
Universidad Tecnológica Nacional – Facultad Regional Córdoba
leonardorciceri@gmail.com

Juliana Notreni
Universidad Tecnológica Nacional – Facultad Regional Córdoba
julinotreni@gmail.com

Germán Parisi
Universidad Tecnológica Nacional – Facultad Regional Córdoba
germannparisi@gmail.com

Analía Ruhl
Universidad Tecnológica Nacional – Facultad Regional Córdoba
analialorenaruhl@gmail.com

Milagros Zea Cárdenas
Universidad Tecnológica Nacional – Facultad Regional Córdoba
milyzc@gmail.com

Marcelo Auquer
Universidad Tecnológica Nacional – Facultad Regional Córdoba
marcelo.auquer@gmail.com

Ileana Barrionuevo
Universidad Tecnológica Nacional – Facultad Regional Córdoba
ilebarrionuevo@gmail.com

Federico Bertola
Universidad Tecnológica Nacional – Facultad Regional Córdoba
fedebertola@gmail.com

Sergio Quinteros
Universidad Tecnológica Nacional – Facultad Regional Córdoba
ser.quinteros@gmail.com

Ignacio Sánchez Balzaretti
Universidad Tecnológica Nacional – Facultad Regional Córdoba
ignaciojsb@gmail.com

Resumen

El creciente uso de los dispositivos móviles conlleva una mayor utilización de aplicaciones móviles tanto para la búsqueda de datos, uso de redes sociales como para realizar transacciones bancarias, etc. Esto ha generado un nuevo mercado para los delincuentes informáticos. La mayoría de estas aplicaciones se comunican con

servidores para enviar datos. Esta comunicación es la que se pretende analizar cuando se realizan auditorías de seguridad en una aplicación móvil. Esta comunicación depende, entre otras cosas, del sistema operativo sobre el que se ejecuta la aplicación, el lenguaje de la aplicación, las librerías implementadas, los tipos de cifrado aplicados y el protocolo de comunicación. Al analizar el tráfico de una aplicación, es necesario un marco de trabajo que agilice las tareas de configuración y unifique la diversidad de técnicas con las que se encuentra el pentester al momento de interceptar el tráfico.

Palabras clave: ciberseguridad, pentest, intercepción de tráfico

Abstract

The growing use of mobile devices leads to a greater use of mobile applications both for data search, use of social networks and for banking transactions, etc. This has created a new market for cybercriminals. Most of these applications communicate with servers to send data. This communication is the one that is intended to be analyzed when security audits are carried out in a mobile application. This communication depends, among other things, on the operating system on which the application runs, the language of the application, the implemented libraries, the type of encryption applied and the communication protocol. When analyzing the traffic of an application, a framework is necessary that speeds up the configuration tasks and unifies the diversity of techniques that the pentester encounters when intercepting the traffic.

Keywords: cybersecurity, pentest, traffic interception

Introducción

En los últimos años, se ha visto cómo los dispositivos móviles (tablets, teléfonos celulares) han cobrado mayor protagonismo en el uso cotidiano, desde una búsqueda de datos, sorteos, uso de redes sociales hasta conectarse a servidores. Estos dispositivos, ya no solo se utilizan para comunicarse con otras personas a través de llamadas o mensajería de texto, sino también, para realizar transacciones bancarias, compras, entretenimiento, etc.

El creciente uso de los dispositivos móviles es lo que lleva a utilizar un mayor número de aplicaciones móviles, lo que ha generado un nuevo mercado para los delincuentes informáticos.

Entre los tipos de *malware* móviles conocidos se pueden encontrar (Kaspersky, 2023):

- *Malware bancario*: el *malware* móvil basado en la banca va en aumento, ya que los hackers buscan comprometer a los usuarios que prefieren llevar a cabo todos sus negocios (incluidos los pagos de facturas y las transferencias de dinero) desde sus dispositivos móviles. Por mencionar uno, el malware denominado TeaBot, tiene la capacidad de robar credenciales de los usuarios, así como mensajes SMS, para allanar el camino a actividades fraudulentas contra bancos en Italia, Bélgica, España, Alemania y Holanda. Una vez que TeaBot está instalado en el dispositivo de la víctima, los atacantes pueden obtener fácilmente una transmisión en vivo de la pantalla del mismo y también pueden interactuar con él a través de los servicios de accesibilidad (DIW, 2021).
- *Ransomware móvil*: el *ransomware*, que primero se hizo popular en los ordenadores de sobremesa, "bloquea" datos importantes del usuario, como documentos, fotos y videos, mediante el cifrado de la información y luego exige el pago de un rescate a los creadores del *malware*. Si no se paga el rescate a tiempo (generalmente en *Bitcoin*), se eliminan todos los archivos o simplemente se bloquean, de manera que el usuario nunca más podrá acceder a ellos. Un ejemplo es *Doublelocker,* que se propagó a través de aplicaciones falsas que se descargaron de sitios web comprometidos. El *malware* cambió el PIN del dispositivo afectado y cifró los archivos del almacenamiento principal, renombrándolos con la extensión ".cryeye". Se exigió un rescate para descifrarlos (Lipovsky and Stefanko, 2018).
- *Spyware móvil*: el *spyware*, cargado como un programa en el dispositivo del usuario, supervisa su actividad, registra su ubicación y sustrae información crítica, como nombres de usuario y contraseñas de cuentas de correo electrónico o sitios de comercio electrónico. En muchos casos, con el *spyware* se incluye otro software aparentemente benigno que recopila datos silenciosamente en segundo plano. Incluso puede que no se note la presencia del *spyware* hasta que el rendimiento del dispositivo disminuye o hasta que se ejecute un análisis *antimalware* en el dispositivo. El caso más conocido es *Pegasus,* del grupo NSO Group. Esta herramienta espía dispositivos móviles iOS y Android. NSO Group es una compañía israelí que se dedica a vender herramientas de vigilancia de alta tecnología a agencias de todo el mundo. Este malware desató el escándalo del presunto espionaje de los Estados Unidos Mexicanos a íconos de la sociedad en 2017, entre ellos periodistas y defensores de derechos humanos. Los enlaces de explotación de Pegasus y los servidores *Command & Control*

utilizan HTTPS, que requiere que los operadores registren y mantengan los nombres de dominio. Los nombres de dominio para enlaces de explotación a veces se hacen pasar por proveedores móviles, servicios en línea, bancos y servicios gubernamentales, lo que puede hacer que los enlaces parezcan benignos a primera vista(Soltero et al.,2019)(Marczak, 2018).

- *Malware de MMS*: los creadores de *malware* también buscan maneras de aprovechar la comunicación basada en texto como una forma de distribuir *malware*. Como ejemplo, está *Stagefright*, un conjunto de vulnerabilidades en la biblioteca multimedia de Android hizo posible que los atacantes envíen un mensaje de texto incrustado con *malware* a cualquier número de teléfono móvil. Tiene asignados varios CVE ID (*Common Vulnerabilities and Exposure)*, como CVE-2015-3864 (CVE Mitre, 2015).

En 2021 se identificaron 571 vulnerabilidades relacionadas con el sistema operativo Android, en 2022, 897 vulnerabilidades y en lo que va de 2023 hasta mayo ya se encuentran identificadas 221 (Google, 2023), según las estadísticas del cvedetails.com, que es una fuente de información libre de vulnerabilidades y exposiciones comunes (CVE -*Common Vulnerabilities and Exposures*). Por otro lado, respecto a vulnerabilidades relacionadas con el sistema operativo iOS, fueron 380 las vulnerabilidades identificadas en 2021, 242 en 2022 y 37 en lo que va del año 2023 hasta mayo (Apple, 2023).

Las versiones de Android afectadas han sido la 10, 11, y 12. Entre los impactos más señalados de las vulnerabilidades identificadas hasta principios de mayo de 2023 en Android se encuentran (Android, 2022):

- Elevación de privilegios.
- Denegación de servicio.
- Exposición de información.

La pandemia Covid-19 ha presentado negocios a nivel mundial, lo que incluye desafíos de ciberseguridad, como campañas de *phishing* y oportunidades para ingeniería social. También las compañías deberían tener en cuenta buenas prácticas para proteger el router, elementos de Internet de las cosas (IoT) y las redes privadas virtuales (VPN) de sus empleados que trabajan desde sus hogares (Accenture, 2020).

Según el Reporte Anual sobre Crimen en Internet del 2019 del FBI, las pérdidas económicas de ese año fueron de 3500 miles de millones dólares (FBI, 2019).

Command and Control, también conocida como *C2* o *C&C* es el conjunto de herramientas y técnicas que usan los atacantes para mantener comunicación con los dispositivos comprometidos. Es una de las 11 tácticas definidas por la MITRE ATT&CK. Existen diferentes formas en las que un atacante puede establecer control con el dispositivo comprometido dependiendo de la estructura de red y defensas de la víctima. La táctica C&C contiene 16 técnicas (Mitre ATT&CK, 2018). Un ejemplo reciente de aplicación de C&C en computadoras personales, es el incremento de criminales que usan *Telegram*, aplicación de mensajería instantánea, para el control usando un nuevo malware llamado *ToxicEye*, descubierto recientemente (Hofman, 2021). Por otro lado, un C&C en dispositivos móviles es *FlixOnline*. Es una aplicación con un servicio falso que pretende permitir a los usuarios ver contenido de *Netflix* de todo el mundo en sus teléfonos móviles, Sin embargo, la aplicación está diseñada para monitorear las notificaciones de *WhatsApp* y enviar respuestas automáticas a los mensajes entrantes del usuario utilizando un contenido que recibe de un servidor de C&C (Hazum et al., 2021).

Otro ataque, no tan reciente, pero no por eso menos usado, es el del tipo *man-in-the-middle*. Este ataque se ha dado mucho en aplicaciones web y explota el hecho de que el servidor HTTPS envía un certificado con su clave pública al navegador web. Si este certificado no es confiable, toda la comunicación es vulnerable. Reemplaza el certificado original que autentica al servidor HTTPS servidor con un certificado modificado. El ataque tiene éxito si el usuario se niega a verificar el certificado cuando el navegador envía una notificación de advertencia. Esto ocurre con demasiada frecuencia, especialmente entre los usuarios que con frecuencia encuentran certificados autofirmados al acceder a sitios de intranet. De esta forma, el atacante puede interceptar y "escuchar" mensajes de terceros (Callegati et al., 2009) (Hubbard et al., 2014).

En las aplicaciones móviles existen técnicas como *SSL pinning* para evitar los ataques *man-in-the-middle*. Esta técnica funciona de la siguiente manera: al realizarse la negociación SSL y el servidor envía su certificado, por defecto Android (aunque ocurre igual en otras plataformas) comprueba que dicho certificado pertenezca a una autoridad certificadora de confianza y que este no está revocado o caducado. Cuando el dispositivo se encuentra en una red pública, es posible que un atacante se coloque "en medio" y se haga pasar por el servidor, haciendo de puente entre este y el dispositivo. Si esto lo hace con un certificado válido, el sistema comprueba el certificado y lo dará por válido, pudiendo este atacante hacerse con todos los datos que se intercambian con el servidor en texto plano. *SSL Pinning* se denomina al proceso de verificar además que el certificado que ha enviado el servidor sea solo el del servidor de la aplicación y no cualquiera válido. Así, si se detecta un certificado válido, pero que no es el del servidor, se puede rechazar la conexión.

Estos vectores y tácticas de ataques exponen la urgente necesidad de ejecutar pruebas de penetración a apps móviles antes de que estas salgan al mercado. Si bien estas pruebas no aseguran que las aplicaciones sean o no vulneradas eventualmente, reducen los riesgos de disponibilizar una aplicación móvil con vulnerabilidades ya conocidas.

Actualmente, la mayoría de los sistemas de información incluyen aplicaciones móviles dentro de su solución. Por lo tanto, estas también tienen que ser analizadas e incluidas dentro de las pruebas de seguridad realizadas al sistema.

Gadient et al. analizaron 160 aplicaciones Android, y si bien encontraron que el protocolo HTTPS es usado en la mayoría de ellas, también identificaron numerosos casos de SQL embebido en endpoints con los que se comunica la app. Además, en el lado del servidor identificaron mala configuración de servidores, servidores web desactualizados, interpretadores de lenguaje con vulnerabilidades conocidas, fuga de mensajes con errores internos y datos sensibles. Por último, identificaron comunicación con servidores privados sin ninguna clase de mecanismos de autenticación o autorización (Gadient et al., 2020).

Las pruebas de penetración o pentesting es un método de prueba con el foco en componentes binarios individuales o en la aplicación como un todo para determinar si las vulnerabilidades en o entre componentes pueden explotarse para comprometer la aplicación, sus datos o los recursos de su entorno (NIST, 2020). El proceso implica un análisis activo del sistema para detectar posibles vulnerabilidades, incluidas la deficiente o inadecuada configuración del sistema, fallas de hardware y software y debilidades operativas en el proceso o contramedidas técnicas (Mohanty, 2013).

Una de las actividades cuando se está realizando pruebas de penetración en una aplicación móvil es determinar con qué servidores se comunica -en el caso de que se comunique- y estudiar dicha comunicación. El estudio de la comunicación entre una aplicación web y los servidores también se lleva a cabo en actividades de auditoría o actividades forenses, para los casos que haya existido una vulneración de aplicaciones móviles, por ejemplo, los ataques mencionados anteriormente.

Desarrollo

Estado actual sobre intercepción de tráfico en aplicaciones móviles

Por lo expuesto, es importante identificar los recursos con los que tiene conexión y comunicación una aplicación móvil y analizarlos. Hasta el momento, no hay suficientes herramientas de código abierto o software libre que hagan de estas actividades una tarea sencilla tanto para sistemas operativos Android como iOS. Si bien existen herramientas para interceptar tráfico de aplicaciones, a veces existen capas de protección en la misma aplicación, que evitan que estas herramientas puedan efectivamente capturar su comunicación. Algunas de esas herramientas son:

- *Burp suite:* Es un framework para hacer análisis de seguridad. Permite interceptar peticiones HTTP o mensajes *WebSocket* entre una aplicación móvil y un servidor. Tiene una versión Community y otra versión paga. (Portswigger, 2023)
- *Wireshark:* Analizador de red, permite analizar paquete por paquete y su contenido. Wireshark corre en sistemas operativos tipo *Unix*, incluyendo *Linux, Solaris, FreeBSD, NetBSD, OpenBSD, macOS*, y también en sistemas como *Microsoft Windows, U3* y en *Portable Apps*. (Wireshark, 2023)
- *Tamper Dev:* Es una extensión de Google que intercepta peticiones HTTP. Es de código abierto. (Tamper Dev, 2021) [22]
- *Frida:* Ayuda a realizar ingeniería inversa y análisis de seguridad. Es un conjunto de herramientas de instrumentación de código dinámico. Permite inyectar fragmentos de JavaScript o una biblioteca propia en aplicaciones nativas en *Windows, macOS, GNU / Linux, iOS, Android* y *QNX* (*Unix-like real-time operating system*) (Frida, 2023).
- *Objection:* Permite realizar instrumentación a través de *Frida*, con la particularidad de que funciona en dispositivos no rooteados, facilitando las pruebas. También permite inyectar fragmentos de código que interactúen con el comportamiento de la aplicación. Es útil tanto para aplicaciones de Android como de iOS. (Objection, 2021)

Como se mencionó, estas herramientas permiten saltar ciertas capas de protección, pero ¿qué capas permiten saltar?, depende del sistema operativo sobre el que corre la aplicación móvil, el lenguaje de la aplicación, las librerías que usa para ejecutarse o los tipos de cifrado que implementa, el protocolo de comunicación con el servidor, entre otras cosas. Es por esto la necesidad de identificar y analizar todas las capas de protección de una aplicación móvil, investigar cómo se comunica con sus respectivos servidores, con qué servidores, para qué se comunica con ellos durante los análisis de seguridad, auditoría de seguridad o actividades de forensia.

Grado de avance

El proyecto en el que está inserto este trabajo busca generar un framework de automatización de análisis de tráfico de datos entre aplicaciones móviles y servidores remotos, tanto para sistemas operativos Android como iOS describiendo métodos y técnicas de análisis de tráfico entre aplicaciones móviles y servidores.
Para ello se plantean los siguientes objetivos específicos:

- Examinar y categorizar vulnerabilidades conocidas explotadas que se han usado en ataques relacionados con aplicaciones móviles, que incluyan intercepción de tráfico.
- Investigar métodos y técnicas de intercepción de tráfico entre una aplicación Android y servidores remotos.
- Investigar métodos y técnicas de intercepción de tráfico entre una aplicación iOS y servidores remotos.
- Definir alcance, con base en las técnicas investigadas, del framework de automatización de análisis de tráfico de datos entre aplicaciones móviles y servidores remotos.
- Diseñar el framework de automatización de análisis de tráfico de datos entre aplicaciones móviles y servidores remotos.
- Desarrollar e implementar componentes que integren el framework de automatización definido.
- Transferir el conocimiento obtenido a los ámbitos académicos, de investigación y a la industria a través de conferencias y talleres prácticos.

Conclusiones

Actualmente el estudio del tráfico de datos en aplicaciones web es un tema que se viene desarrollando y aplicando en distintos frameworks para lograr estudiar la seguridad de dichos sistemas, pero cuando se habla de sistemas que incluyen aplicaciones móviles son pocas las herramientas disponibles para lograr el estudio del tráfico de datos de dichas aplicaciones. Por esto el objetivo del proyecto, en el que está inserto el presente trabajo es desmitificar este tópico, generando esta línea de conocimiento e investigación de seguridad en aplicaciones móviles.

A la fecha, las herramientas existentes han incorporado funcionalidades que permiten realizar la deshabilitación de diversos métodos de SSL pinning, mayormente a través de scripts de Frida que se combinan para poder efectivizar la intercepción de tráfico.

Los puntos claves de este proyecto, que mejoran el uso de las herramientas ya existentes y las integran mediante automatización, son los siguientes:

- Detección del sistema operativo y arquitectura del dispositivo en donde se realizará la intercepción de tráfico.
- Detección del modo root / no root, para determinar qué herramienta utilizar: instalación del servidor (agente) de Frida en el dispositivo, o bien, inyección del gadget de Frida en la aplicación objetivo para que funcione en un dispositivo no rooteado. Ejemplo: selección de la herramienta *objection* para modificar una apk e inyectarle la librería necesaria para la intercepción.
- Instalación del certificado de proxy en el dispositivo móvil como certificado de sistema.
- Decompilación de apk y modificaciones de archivos de red de la aplicación para que confíe en certificados de usuario.
- Compilación de la apk y firmado de la misma, para su posterior instalación.
- Intercepción del tráfico, mediante el script de Frida, a través del propio cliente de esta herramienta o a través de *objection*.

Es decir, el proyecto avanza en la dirección de la automatización de varias de las herramientas y procesos manuales existentes que satisfacen los requerimientos de intercepción de tráfico, pero que por sí solas no satisfacen los requerimientos para gran variedad de dispositivos y arquitecturas, por lo cual, se pretende continuar avanzando con la integración de las técnicas de intercepción de manera tal de automatizar progresivamente los puntos previamente mencionados: comenzando por la detección del sistema operativo, arquitectura, modo root, instalación de certificados de proxy necesarios, etc.

Referencias

Amenazas a la seguridad móvil para Android, Kaspersky.

A new malware targeting banking apps on Android is making rounds in Europe, mayo 2021, https://www.digitalinformationworld.com/2021/05/a-new-malware-targeting-banking-apps-on.html, última visita 06/04/2022.

Lipovský, R., Štefanko, L. Android ransomware: From android defender todoublelocker. ESET.
Román Soltero, A., Bautista, B., Ramos, R., Lechuga Salais, A., Carrasco, R., Rodriíguez, N. septiembre 2019. Análisis ético de la información en el escándalo Pegasus. Revista de Investigación en Tecnologías de la Información, Número 14, Vol. 7, ISSN: 2387-0893. Editorial SEICIT. doi: https://doi.org/10.36825/RITI.07.14.003, última visita 06/04/2022.
Marczak, B., Scott-Railton, J., McKune, S., Razzak, B. A., Deibert, R. Sptiembre 2018. HIDE AND SEEK Tracking NSO Group's Pegasus Spyware to Operations in 45 Countries. Citizen Lab. https://tspace.library.utoronto.ca/bitstream/1807/95391/1/Report%23113--hide%20and%20seek.pdf, última visita 06/04/2022.
Detalles de la vulnerabilidad. CVE Mitre. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3864, última visita 06/04/2022.
Google Android: Vulnerability Statistics https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224, última visita 7/05/2023.
Apple Iphone Os : Vulnerability Statistics. https://www.cvedetails.com/product/15556/Apple-Iphone-Os.html?vendor_id=49, última visita 07/05/2023.
Android Security Bulletin, febrero de 2022, https://source.android.com/docs/security/bulletin/2023-05-01, última visita 07/05/2023.
Cyber Threatscape Report 2020. Accenture Security 2020. https://www.accenture.com/_acnmedia/PDF-136/Accenture-2020-Cyber-Threatscape-Full-Report.pdf, última visita 12/04/2022.
Federeal Boreau of Investigation (FBI), 2019 Internet Crime Annual Report, US Department of Justice, https://www.ic3.gov/Media/PDF/AnnualReport/2019_IC3Report.pdf. Última visita 12/04/2022.
Command and Control. MITRE. ATT&CK. https://attack.mitre.org/tactics/TA0011/ última visita 12/04/2022.
Hofman, O. Abril 2021. Remote access trojan exploits Telegram communications to steal data from victims and update itself to perform additional malicious activities. https://blog.checkpoint.com/2021/04/22/turning-telegram-toxic-new-toxiceye-rat-is-the-latest-to-use-telegram-for-command-control/ última visita 12/04/2022.
Hazum, A., Melnykov, B., Wenik, B. Abril 2021. Autoreply attack! New Android malware found in Google Play Store spreads via malicious auto-replies to WhatsApp messages. Check Point Software Technologies. https://blog.checkpoint.com/2021/04/07/autoreply-attack-new-android-malware-found-in-google-play-store-spreads-via-malicious-auto-replies-to-whatsapp-messages/ última visita 12/04/2022.
Callegati, Franco & Cerroni, Walter & Ramilli, Marco. (2009). Man-in-the-middle attack to the HTTPS protocol. Security & Privacy, IEEE. 7. 78 - 81. 10.1109/MSP.2009.12.
Hubbard, John & Weimer, Ken & Chen, Yu. (2014). A study of SSL Proxy attacks on Android and iOS mobile applications. 86-91. 10.1109/CCNC.2014.6866553.
Gadient, Pascal et al. Web APIs in Android through the Lens of Security. 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (2020): n. pag. Crossref. Web.
NIST, Computer Security Resource Center, https://csrc.nist.gov/glossary/term/penetration_testing, última visita 07/05/2023.
Mohanty, D. Demystifying Penetration Testing HackingSpirits.
Burp Suite Community Edition. Página oficial https://portswigger.net/burp/communitydownload.
Wireshark. Página oficial https://www.wireshark.org/.
Tamper Dev. Repositorio de código https://github.com/google/tamperchrome.
Frida. Página oficial https://frida.re/.
Objection - Runtime Mobile Exploration. Repositorio de código https://github.com/sensepost/objection.