UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

ANÁLISIS DE MALWARE RAT (REMOTE ACCESS TROJAN) MEDIANTE TÉCNICAS DE DETECCIÓN ESTÁTICAS Y DINÁMICAS EN PLATAFORMAS ANDROID

AUTOR

CHANCAY BANCHÓN ARELYS GINGER

PROYECTO UNIDAD INTEGRACION CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

ING. LÍDICE HAZ LÓPEZ, MSI.

SANTA ELENA, ECUADOR

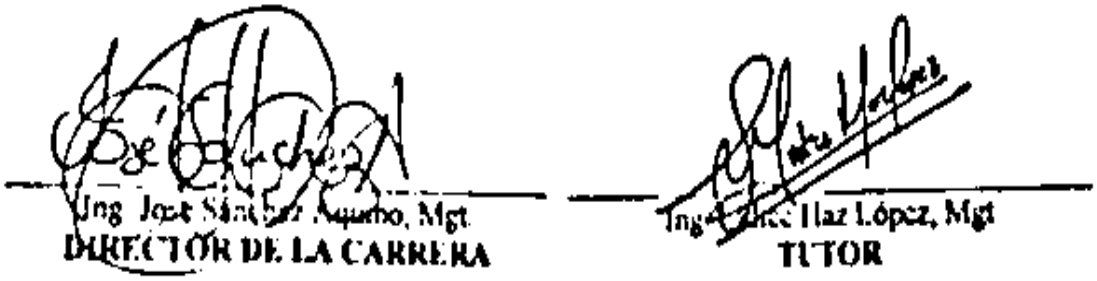
2024

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN



Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. [illegible] Díaz López, Mgt
TUTOR

Ing. Jaime Orozco Iguasnia, Mgt.
DOCENTE ESPECIALISTA

Ing. Marjone Coronel Suárez. Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **CHANCAY BANCHÓN ARELYS GINGER**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 05 días del mes de diciembre del año 2024

TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

Ing. Lídice Haz López, Msi

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Chancay Banchón Arelys Ginger**

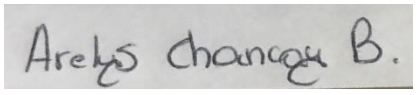
DECLARO QUE:

El trabajo de Titulación**, Análisis de Malware RAT (Remote Acces Trojan) mediante Técnicas de Detección Estáticas y Dinámicas en Plataformas Android.** previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 05 días del mes de diciembre del año 2024

EL AUTOR



Arelys Ginger Chancay Banchón

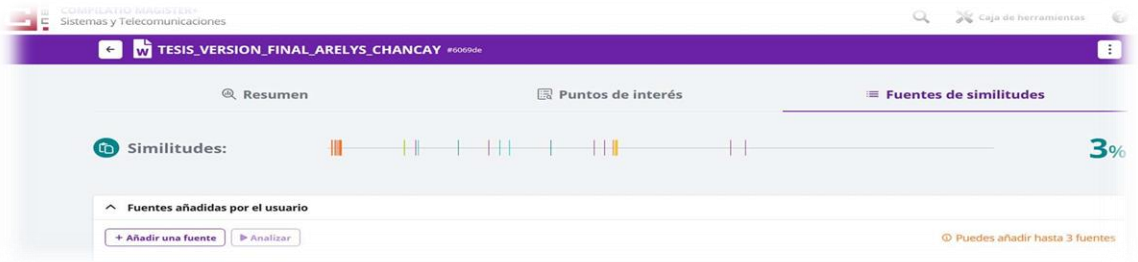
UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Análisis de Malware RAT (Remote Acces Trojan) mediante Técnicas de Detección Estáticas y Dinámicas en Plataformas Android**, presentado por la estudiante Chancay Banchón Arelys Ginger fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Firmado electrónicamente por:
LIDICE VICTORIA HAZ LOPEZ

Ing. Lídice Haz López, Msi

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

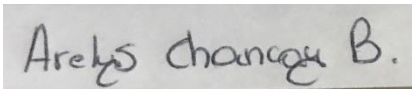
Yo, **Chancay Banchón Arelys Ginger**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 05 días del mes de diciembre del año 2024

EL AUTOR



Arelys Ginger Chancay Banchón

AGRADECIMIENTO

No puede faltar en estas líneas mi agradecimiento inmenso a mi Dios, porque él me ha brindado las ganas de luchar por mis metas, llenándome de mucha Fe y paciencia en cada situación de mi vida.

Agradecida con mi familia que ha estado conmigo apoyándome en este largo camino de formación educativa.

Culmino este agradecimiento hacia mi tutora Ing Lídice Haz por guiarme. Gracias ing por todo el aprendizaje, gracias porque pude contar con usted desde el día uno que acepto ser mi tutora.

DEDICATORIA

Dedico este proyecto de investigación a mi familia y a mi Abuelo, pues él fue mi primer profesor en casa, también de mis hermanos y mis primos, sé que no podrá leer estas líneas, sé que estás en un mejor lugar, te extraño mucho, en mi memoria abarca todos los recuerdos de mi infancia. Abrazos y miradas al cielo, gracias a Dios por ser mi abuelo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE IMÁGENES

RESUMEN

Este estudio se centra en el estudio del malware RAT (Acceso Remoto Trojan) en dispositivos Android a través de la implementación de métodos de detección estáticos y dinámicos. Estas amenazas posibilitan el control remoto no permitido, poniendo en riesgo la seguridad y privacidad de los usuarios al propiciar el hurto de información delicada, seguimiento de acciones y manejo de dispositivos. Se establecieron laboratorios virtuales utilizando herramientas especializadas como MobSF y Drozer, lo que posibilita un ambiente regulado para analizar la conducta del malware. El estudio se llevó a cabo en cinco etapas: configuración del ambiente, elección de las muestras, análisis, exámenes de laboratorio y comunicación de los descubrimientos. Los hallazgos abarcaron la detección de vectores de ataque, procesos de propagación y tácticas de persistencia, además de la creación de una guía de prevención y mitigación para los vectores de ataque.

Palabras claves : RAT (Remote Access Trojan), Malware, Android

ABSTRACT

This study focuses on the analysis of RAT (Remote Access Trojan) malware on Android devices through the implementation of static and dynamic detection methods. These threats enable unauthorized remote control, jeopardizing user security and privacy by facilitating the theft of sensitive information, activity monitoring, and device manipulation. Virtual laboratories were established using specialized tools such as MobSF and Drozer, enabling a controlled environment to analyze the behavior of the malware. The study was conducted in five stages: environment setup, sample selection, analysis, laboratory testing, and reporting of findings. The results included the identification of attack vectors, propagation processes, and persistence tactics, as well as the development of a prevention and mitigation guide for the identified attack vectors.

The keywords are: RAT (Remote Access Trojan), Malware, Android

INTRODUCCIÓN

Actualmente, el malware RAT (Trojan de Acceso A distancia) se ha establecido como una amenaza sofisticada en dispositivos Android, proporcionando a los atacantes control total a distancia para llevar a cabo tareas como la extracción de información delicada, el manejo del sistema y la realización de acciones no permitidas. Su análisis técnico fusiona métodos estáticos, como el desensamblado de código y la identificación de patrones malintencionados, con técnicas dinámicas que monitorean su actuación en tiempo real e interactúan con servicios del sistema

Etapa Capítulo I: En esta sección se incluye en el desarrollo del problema principal, entender el crecimiento que experimenta todo el tema de Malware RAT en dispositivos móviles Android, su estado actual, razones, evolución, entre otros aspectos. Además, se basa en conocer técnicas fundamentales para el análisis adecuado de estos archivos malintencionados, empleando métodos de seguridad como el análisis estático y dinámico para la evaluación.

Etapa Cpítulo II: Establece un método de cinco etapas para examinar el malware RAT en Android. Se estableció un laboratorio virtual utilizando recursos como MobSF y Drozer, se escogieron muestras pertinentes y se examinaron a través de métodos estáticos y dinámicos. En experimentos de laboratorio, se realizaron simulacros de ataques para analizar el comportamiento del malware, corroborando su efecto en la seguridad.

Etapa Capítulo III: elabora un procedimiento de cinco etapas para examinar el malware RAT en Android. Se estableció un laboratorio virtual utilizando recursos como MobSF y Drozer, se escogieron muestras pertinentes y se examinaron a través de métodos estáticos y dinámicos. En experimentos de laboratorio, se realizaron simulacros de ataques para analizar el comportamiento del malware, corroborando su efecto en la seguridad. Finalmente, se produjo un informe exhaustivo de los descubrimientos y se desarrolló una guía con sugerencias útiles para prevenir y minimizar riesgos, reforzando la ciberseguridad en dispositivos Android.

CAPÍTULO I

1 FUNDAMENTACIÓN

1.1 ANTECEDENTES

Hoy en día, el avance tecnológico en dispositivos móviles es un tema de gran relevancia que ha marcado el comienzo de lo que se conoce como la "era de la conectividad ininterrumpida". Este progreso ha transformado la manera en que las personas se comunican e interactúan con la tecnología, ofreciendo una amplia gama de servicios y funcionalidades desde la palma de su mano [1].

No obstante, este avance también ha presentado desafíos significativos en términos de seguridad cibernética. La creciente dependencia de los dispositivos móviles para realizar transferencias financieras y acceder a información confidencial ha dado lugar a la aparición de amenazas latentes y muy maliciosas, como el ransomware, el troyano bancario, el RAT (Remote Access Trojan), así como el adware. Estos programas maliciosos realizan una variedad de actividades dañinas para extraer información confidencial de manera no autorizada [2].

En 2023, Kaspersky observó un aumento constante en el número de amenazas a dispositivos móviles, que alcanzó casi 33,8 millones de ataques, lo que supone un aumento de más del 50% con respecto a las cifras del año anterior. La amenaza más frecuente para los dispositivos móviles fue el adware, que constituye el 40.8% de todas las amenazas detectadas. Este creciente aumento de los ataques informáticos en dispositivos móviles ha sido una tendencia que se ha visto impulsada por la ubicuidad de los teléfonos inteligentes y tabletas en la vida cotidiana. Los atacantes informáticos han identificado la vulnerabilidad inherente en la constante conectividad a Internet y el manejo delicado de información en estos dispositivos [3].

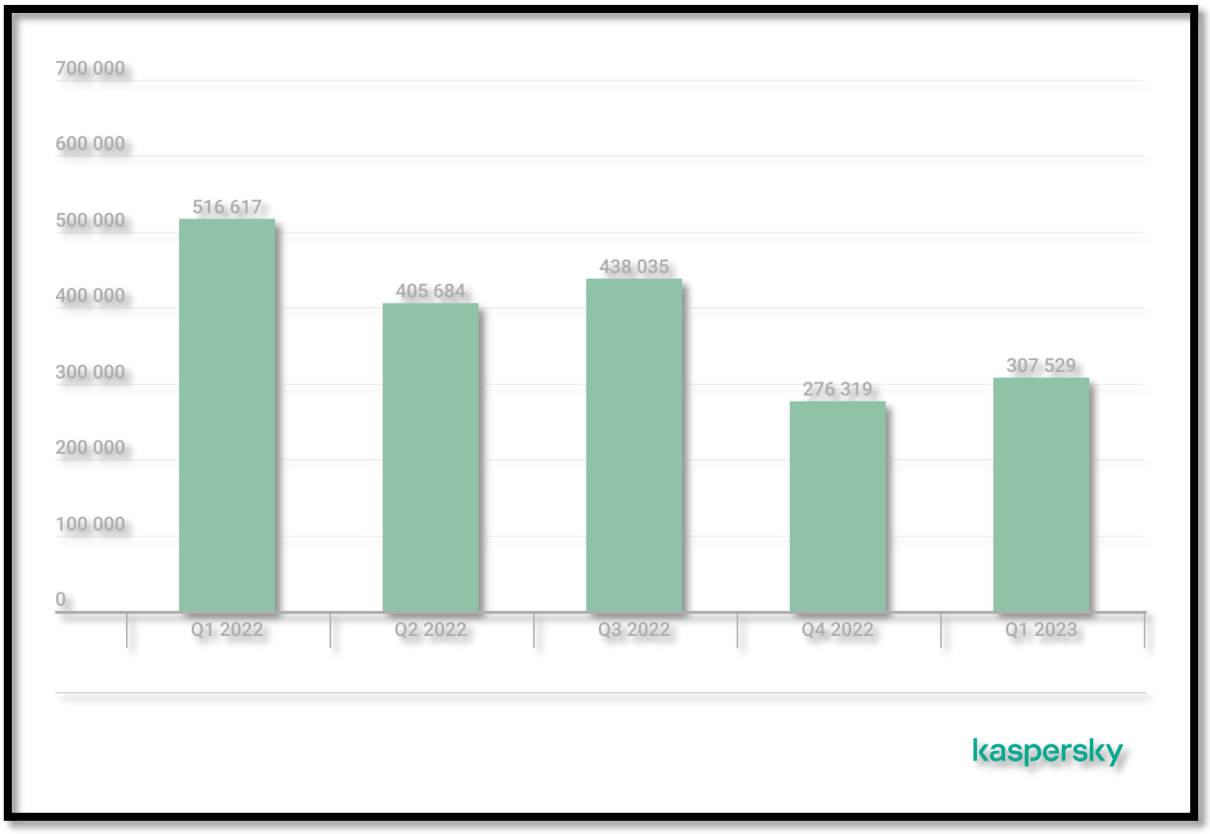


Figura 1: Estadísticas de las amenazas móviles Fuente: [4]

El gráfico proporcionado por Kaspersky Lab, que refleja las estadísticas de amenazas móviles, muestra fluctuaciones en la cantidad de amenazas detectadas a lo largo del año, destacando una disminución entre el primer y segundo trimestre de 2022; sin embargo, la tendencia general indica un aumento en las cifras de amenazas. Este patrón evidencia que, a pesar de las medidas de seguridad implementadas, los ciberdelincuentes continúan adaptando sus tácticas para comprometer dispositivos móviles. Entre las posibles razones de este crecimiento se encuentran el aumento en el uso de teléfonos inteligentes, la aparición de nuevas vulnerabilidades en sistemas operativos y aplicaciones, así como la creciente sofisticación de los ataques dirigidos [4]. Por ello, es fundamental que los usuarios comprendan la naturaleza de estas amenazas y adopten medidas de protección proactivas, como mantener sus dispositivos actualizados, descargar aplicaciones exclusivamente de

fuentes confiables e implementar soluciones de seguridad móvil, para salvaguardar la integridad de sus dispositivos en un entorno de amenazas en evolución.

Para el presente proyecto se dará iniciativa a guías de investigaciones relacionadas al tema ya sea artículos, tesis, con la finalidad de entender más al fondo sobre el comportamiento de malware en dispositivos móviles Android y a su vez la importancia de la ciberseguridad en la protección de datos personales. Por lo tanto, se mencionan tres aspectos esenciales de estudio como base indagadas de manera local, internacional y nacional.

La tesis "**Análisis del impacto de los ataques de ransomware en las organizaciones colombianas para la determinación de nuevas estrategias de protección cibernética**", de Jhon Jairo Pinzón R., subraya la importancia crítica del ransomware, que cifra datos y exige un rescate. También examina la evolución tecnológica que ha facilitado la aparición de nuevas variantes de ransomware, capaces de afectar diversos sistemas y dispositivos. Además, destaca el aumento significativo de eventos de seguridad cibernética en Colombia en los últimos años, con pérdidas financieras considerables para las organizaciones afectadas. Este estudio proporciona una visión profunda del impacto del ransomware en el entorno empresarial colombiano y ofrece una base sólida para el desarrollo de estrategias efectivas de protección cibernética [4].

La tesis "**Análisis comparativo de Malware en teléfonos inteligentes Android, prevención y mitigación de sus efectos**" de Stalyn Andrango P., menciona como el avance tecnológico ha transformado la vida diaria, especialmente en el ámbito laboral, pasando de grandes computadoras a dispositivos móviles. A pesar de su conveniencia, la seguridad en estos dispositivos suele ser descuidada, convirtiéndolos en blancos para ciberdelincuentes. Por ello, este proyecto se centra en mejorar la seguridad de los dispositivos móviles con Android, que dominan el mercado. Identificar y abordar sus vulnerabilidades críticas puede reducir significativamente los daños causados por ataques cibernéticos [5].

La tesis **"Implementación de un laboratorio virtual de análisis y comportamiento de malware para mejorar la seguridad y protección de datos en los laboratorios de la Facultad de Sistemas y Telecomunicaciones (FACSISTEL)"**, de [Nombre del autor], subraya la necesidad crítica de controlar el malware en la red de la facultad. Identifica la

falta de regulación en el tráfico de red como una causa de inseguridad, que permite la propagación de software malicioso y amenaza la integridad de los datos. Para abordar este problema, la tesis propone la implementación de un laboratorio virtual en los servidores de FACSISTEL, utilizando herramientas de código abierto para análisis estático y dinámico de malware [6].

En concordancia con las investigaciones previamente mencionadas, se resalta la esencial tarea de proteger los activos de información frente a las incesantes innovaciones tecnológicas y las nuevas modalidades de ataques que surgen. En este contexto, se enfoca en el análisis pormenorizado del malware, particularmente los RAT (Remote Access Trojans), con el fin de entender su intrusión en dispositivos Android. Este análisis proporciona una visión profunda de cómo estos programas interactúan con los sistemas, escalan privilegios y extraen información. Además, se hace hincapié en la implementación de sólidas prácticas de seguridad de la información, junto con programas de concientización sobre buenas prácticas de TI, como elementos clave para establecer una protección robusta y proactiva de los datos digitales.

1.2 DESCRIPCIÓN DEL PROYECTO

La falta de diversidad en la información sobre la importancia de la seguridad informática en dispositivos Android ha impulsado un aumento significativo de ataques cibernéticos, como es el caso de los RAT (Remote Access Trojan). Estos malware permiten acceso no autorizado, robo de información y comprometen gravemente la integridad del sistema. Por ello, se considera que este trabajo aborda la necesidad de realizar un análisis exhaustivo del malware RAT, con el objetivo de desarrollar y proponer una metodología coherente que incluya herramientas y técnicas esenciales para identificar, caracterizar y mitigar eficazmente estos ataques en dispositivos Android. Se busca garantizar así la protección integral de la seguridad y privacidad de los usuarios en entornos móviles.

Los test de penetración son cruciales para el estudio, ya que permiten simular escenarios reales y comprender de manera detallada la problemática. Se considera que, en este trabajo, dichos análisis ayudarán a examinar el accionar del malware RAT en dispositivos Android,

comprendiendo su naturaleza, los objetivos de los payloads y los activos de información comprometidos. Por ende, el estudio de malware a través de técnicas estáticas y dinámicas es clave para desarrollar buenas prácticas que permitan combatir las brechas de seguridad que estos malware ocasionan.

El presente proyecto toma como guía la Metodología Penetration Testing Execution Standard (PTES)

Fase 1: Configuración del Laboratorio Virtual

- Preparación del Laboratorio Virtual controlado
- Instalación de herramientas de análisis
- Configurar sistemas de Registro

Fase 2: Elección de la muestra

- Explorar las técnicas estáticas y dinámicas para emplear el análisis de Malware
- Emplear parámetros de agrupación de clasificación como Vector de ataque, sistemas en peligro, vulnerabilidad a provocar, entre otros.
- Categorizar el Origen del RAT
- Buscar malware´s RAT más empleados en Android

Fase 3: Análisis de Malware

- Seleccionar 3 Malware´s de la fase 2 para emplear el análisis en las herramientas a trabajar
- Obtener la información detallada del malware
- Conocer el comportamiento malicioso
- Activos de información en peligro
- Vulnerabilidades provocadas

Fase 4: Pruebas de laboratorios

- Desarrollar pruebas específicas reales como controladas
- Monitorear el comportamiento del malware
- Identificar vulnerabilidades explotadas
- Conocer el nivel de acceso al objeto victima

- Recopilar información relevante (Activos de información Usuario)

Fase 5: Reporte

- Datos de pruebas
- Objetivo
- Alcance
- Análisis e interpretación de resultados
- Observación practica
- Recomendación técnicas

1.3 OBJETIVOS DEL PROYECTO

1.3.1 OBJETIVO GENERAL

Implementar un laboratorio controlado virtual para el análisis de Malware RAT mediante técnicas de detección estáticas y dinámicas en dispositivos móviles Android.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analizar técnicas de detección estáticas y dinámicas empleadas en el Malware RAT en Android para comprender su comportamiento.
- Utilizar laboratorios de pruebas usando máquinas virtuales pre-configuradas para la inspección del Malware RAT.
- Describir los mecanismos de propagación y comportamiento del malware RAT.
- Elaborar una guía de métodos de prevención para minimizar la materialización de los riesgos y pérdida que genera el malware RAT.

1.4 JUSTIFICACIÓN DEL PROYECTO

El análisis del malware RAT (Remote Access Trojan) reviste una importancia crítica en el ámbito de la seguridad cibernética, especialmente en entornos Android. Los RAT representan una forma avanzada de software malicioso diseñado para infiltrarse en sistemas informáticos, permitiendo a los atacantes acceder y controlar remotamente los dispositivos infectados. Esta modalidad de malware puede llevar a la exfiltración de datos sensibles, el robo de

información financiera y personal, así como a la violación de la privacidad de los usuarios [7].

Es esencial comprender la amplia gama de malware existente en el entorno digital, que incluye tipos como el ransomware, troyanos bancarios, RAT y adware. El ransomware, por ejemplo, bloquea el acceso a archivos y exige un rescate para su liberación, mientras que los troyanos bancarios se centran en el robo de información financiera, como contraseñas y detalles de tarjetas de crédito, comprometiendo la seguridad de las transacciones en línea. Por otro lado, los RAT proporcionan a los atacantes acceso remoto y control sobre dispositivos comprometidos, permitiéndoles llevar a cabo una variedad de actividades maliciosas, desde la vigilancia hasta la ejecución de ataques de denegación de servicio [8].

La necesidad de salvaguardar la información y proteger la identidad digital personal se vuelve aún más evidente en la era digital actual, donde la mayoría de las actividades se llevan a cabo en línea. La pérdida o compromiso de datos sensibles puede tener consecuencias devastadoras para los individuos y las organizaciones, incluyendo el robo de identidad, la pérdida financiera y el daño a la reputación. Por lo tanto, es imperativo implementar medidas de seguridad proactivas y eficaces para mitigar los riesgos asociados con los RAT y otras formas de malware.

En este sentido, la concientización sobre buenas prácticas de TI juega un papel fundamental en la protección de la identidad digital personal y la prevención de ataques de malware. Educando a los usuarios sobre cómo reconocer y evitar las técnicas de ingeniería social, cómo mantener actualizados los sistemas operativos y las aplicaciones, y cómo utilizar herramientas de seguridad como firewalls y antivirus, se puede reducir significativamente el riesgo de infección por malware y proteger la integridad de los datos personales y empresariales. En última instancia, la combinación de análisis de malware avanzado y prácticas de seguridad sólidas es esencial para garantizar la seguridad y la privacidad en el entorno digital actual.

1.5 ALCANCE DEL PROYECTO

El enfoque del proyecto propuesto se basa en una metodología de investigación experimental, que implica el análisis detallado de las técnicas estáticas y dinámicas para la detección y mitigación del malware RAT (Remote Access Trojan) en dispositivos Android. Esta metodología permitirá la evaluación de los resultados antes y después de la investigación, centrándose en la eficacia de las técnicas utilizadas para combatir esta amenaza. La variable de medición principal se enfoca en la capacidad de detección y el impacto en la seguridad de los dispositivos Android frente a la presencia de malware RAT. Se utilizarán casos reales de pruebas y máquinas virtuales para simular escenarios prácticos y evaluar la eficacia de las medidas implementadas. Este proyecto se alinea estrechamente con el tema de la tesis, ya que busca comprender y mejorar la capacidad de protección contra el malware RAT en dispositivos Android mediante el uso de técnicas estáticas y dinámicas de análisis [9].

Fase 1: Configuración del Laboratorio Virtual

Durante esta fase, se prepara un entorno virtual controlado, instalando las herramientas de análisis necesarias y configurando los sistemas de registro para monitorear las actividades y resultados del análisis.

Fase 2: Elección de la Muestra

Se exploran técnicas estáticas y dinámicas para el análisis de malware. También se emplean parámetros de clasificación como vectores de ataque, sistemas en riesgo y vulnerabilidades potenciales. Se categoriza el origen del RAT y se identifican los malware RAT más utilizados en dispositivos Android.

Fase 3: Análisis de Malware

En esta fase, se seleccionan tres malware de la fase anterior para su análisis detallado, con el objetivo de comprender su comportamiento malicioso, identificar los activos en peligro y las vulnerabilidades explotadas.

Fase 4: Pruebas de Laboratorios

Se realizan pruebas controladas para monitorear el comportamiento del malware, identificar las vulnerabilidades explotadas, evaluar el nivel de acceso a los dispositivos afectados y recopilar información relevante sobre los activos de información comprometidos.

Fase 5: Reporte

En esta etapa final, se documentan los datos de las pruebas, el objetivo, el alcance, y se realiza un análisis e interpretación de los resultados obtenidos. Se incluyen observaciones prácticas sobre el comportamiento del malware y las vulnerabilidades detectadas.

1.6 METODOLOGÍA DEL PROYECTO

1.6.1 METODOLOGÍA DE INVESTIGACIÓN

El método exploratorio tiene como definición el comportamiento de analizar una problemática de investigación poco conocido sin ser abordado a profundidad y que carece de ideas de información [10]. El tema de investigación cumple la necesidad de abordar un sin número de información relevantes que contendrán aspectos superficiales como base sobre el análisis de malware RAT en dispositivos móviles, debido que son dispositivos que cuentan con alto espectro de información relevante es peligro cuando se trata de ataques informáticos.

Por lo tanto, se toma en cuenta la investigación cualitativa que emplea un enfoque central sobre las percepciones en experiencias y comportamientos de los usuarios en relación con la seguridad informática y la protección contra malware. Esto podría implicar entrevistas en profundidad, grupos focales o análisis de contenido de foros en línea para recopilar datos cualitativos sobre las opiniones y actitudes de los usuarios [11].

Adicional, la investigación será desarrollada mediante búsqueda bibliográfica como el conjunto de conocimientos y técnicas que el profesional o el investigador deben poseer para usar habitualmente la biblioteca y sus fuentes hacer pesquisas bibliográficas y escribir documentos científicos.

El alcance de la investigación se define como descriptivo. Esta elección se basa en el objetivo de describir detalladamente las técnicas estáticas y dinámicas utilizadas en la detección y mitigación del malware RAT en plataformas Android. En lugar de profundizar en la comprensión de cómo y por qué estas técnicas son efectivas, se busca proporcionar una descripción exhaustiva del fenómeno estudiado, definiendo sus características, aplicaciones y limitaciones. El enfoque estará en identificar las diversas técnicas utilizadas en la detección y mitigación del malware RAT, así como en explicar su funcionamiento y aplicabilidad en entornos de seguridad de dispositivos Android [12].

1.6.2 VARIABLES DEL ESTUDIO

Variable Independiente: Técnicas de detección estáticas y dinámicas.

Variable Dependiente: Resultado del análisis del malware RAT

1.6.3 HIPÓTESIS

Hipótesis: El uso combinado de técnicas de detección estáticas y dinámicas mejora la precisión y la eficacia en la detección de malware RAT en plataformas móviles Android, en comparación con el uso individual de cada técnica.

1.6.4 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

- Técnicas

  Estudio del arte, entrevistas y fuentes bibliográficas, artículos, tesis

- Instrumentos

  Para la recolección de datos a través de entrevistas, se empleará un guion de entrevista semiestructurado diseñado específicamente para explorar las percepciones, experiencias y comportamientos de los usuarios en relación con la seguridad informática y la protección contra el malware en dispositivos Android. Este guion incluirá preguntas abiertas que permitirán a los participantes expresar sus opiniones y preocupaciones de manera amplia y detallada. Además, se utilizará equipo de grabación de audio o video para registrar las entrevistas y garantizar una captura precisa de los datos. El uso de entrevistas semiestructuradas proporcionará

flexibilidad para profundizar en temas relevantes y capturar una variedad de perspectivas de los participantes.

- Población
  - Entrevista Expertos en el area de seguridad informática.
  - Conocer la forma de cómo se enfrentan cada vez a las generaciones a las tendencias de la tecnología para así resguardar sus datos personales en un dispositivo móvil.

1.6.4.1 ANÁLISIS DE ENTREVISTA

El objetivo de las entrevistas realizadas a los Especialista 1 y Especialista 2 que son expertos en Ciberseguridad con la finalidad de recopilar información sobre los problemas de la seguridad informática en dispositivos Android, con un enfoque en la detección de malware RAT (Remote Access Trojan) mediante el uso de técnicas de análisis estático y dinámico. A continuación, se presenta un análisis unificado de las ideas y experiencias aportadas por ambos expertos en Ciberseguridad.

. **Experiencia en el Análisis de Malware y RAT en Android**

Ambos ingenieros destacan su experiencia en el análisis de malware, aunque desde perspectivas ligeramente distintas. El Especialista 1 ha trabajado específicamente con técnicas tanto estáticas como dinámicas para la detección de malware en dispositivos Android, lo que le ha permitido tener una visión más integral sobre las ventajas de combinar ambas técnicas (Ver Anexo #1). En cambio, el Ing. Especialista 2, aunque se especializa más en el análisis de tráfico de red, también comparte su enfoque hacia las técnicas dinámicas como un aspecto clave en la detección de malware, subrayando la importancia de la flexibilidad y personalización en estas herramientas (Ver Anexo #2).

2. **Beneficios de la Combinación de Técnicas Estáticas y Dinámicas**

Ambos expertos coinciden en que la combinación de técnicas estáticas y dinámicas es esencial para mejorar la precisión y la eficacia en la detección de malware RAT. El Especialista 1, describe cómo la ofuscación de código puede dificultar la detección con

técnicas estáticas, mientras que las dinámicas revelan comportamientos maliciosos, como la conexión a servidores de comando y control (Ver Anexo #1). Por su parte, el Especialista 2 señala que las técnicas dinámicas permiten modificar el código y generar informes más precisos, lo que favorece la detección de malware que podría no ser identificado con técnicas estáticas por sí solas. Ambos sugieren que la combinación maximiza la capacidad de detección, proporcionando una cobertura más amplia frente a las amenazas (Ver Anexo #2).

3. **Limitaciones de las Técnicas Estáticas**

Las limitaciones de las técnicas estáticas son un tema recurrente en ambas entrevistas. El Especialista 1 menciona que estas técnicas a menudo fallan al identificar malware con técnicas de ofuscación o códigos polimórficos, lo que puede dar lugar a falsos negativos (Ver Anexo #1). El Especialista 2, también subraya que las herramientas estáticas predefinidas no permiten modificaciones, lo que las hace menos flexibles y efectivas en ciertos casos. Ambos coinciden en que el análisis estático, por sí solo, es insuficiente para detectar comportamientos maliciosos que se manifiestan durante la ejecución del malware (Ver Anexo #2).

4. **Utilidad de las Técnicas Dinámicas**

En cuanto a las técnicas dinámicas, ambos expertos están de acuerdo en su efectividad, pero destacan distintos aspectos. El Especialista 1 señala que estas técnicas revelan conexiones maliciosas y descargas de archivos que el análisis estático no puede identificar (Ver Anexo #1). Mientras tanto, el Especialista 2 prefiere las técnicas dinámicas debido a su capacidad de ser adaptadas y modificadas, lo que las hace más adecuadas para sus investigaciones sobre tráfico de red y malware (Ver Anexo #2).

5. **Desafíos en la Implementación de Técnicas Combinadas**

El mayor desafío identificado por ambos ingenieros al combinar estas técnicas es la demanda de recursos. El Especialista 1 hace énfasis en los requerimientos de hardware y la necesidad de entornos controlados para ejecutar análisis dinámicos (Ver Anexo #1), mientras que el

Especialista 2 menciona la diversidad de sistemas operativos y la complejidad de configurar simulaciones de entornos reales en redes empresariales. Ambos superaron estas limitaciones mediante el uso de servidores dedicados y laboratorios virtualizados (Ver Anexo #2).

6. **Métricas de Evaluación de Precisión y Eficacia**

Ambos entrevistados mencionan la importancia de evaluar la detección de malware a través de métricas como el tiempo de ejecución, la tasa de detección, los falsos positivos y la cobertura del comportamiento del malware. El Especialista 1 sugiere que las técnicas combinadas proporcionan una tasa de detección más alta en comparación con las técnicas individuales (Ver Anexo #1), mientras que el Especialista 2 propone el uso de algoritmos de inteligencia artificial para analizar la precisión de estas detecciones. Ambos consideran que las técnicas combinadas ofrecen una visión más completa del comportamiento del malware y reducen la probabilidad de errores en la detección (Ver Anexo #2).

7. **Recomendaciones para Profesionales de Ciberseguridad**

Finalmente, los dos ingenieros recomiendan a los investigadores y profesionales de la ciberseguridad no depender de una sola técnica de detección. El Especialista 1 sugiere invertir en hardware adecuado y utilizar una combinación de análisis estáticos y dinámicos (Ver Anexo #1), mientras que el Especialista 2 enfatiza la importancia de trabajar en seguridad por capas y desarrollar herramientas personalizadas basadas en librerías de código abierto, para garantizar una detección más efectiva y adaptada a las necesidades específicas de cada entorno (Ver Anexo #2).

En conjunto, ambos expertos coinciden en la necesidad de combinar técnicas estáticas y dinámicas para una detección efectiva de malware RAT en dispositivos Android. Aunque sus enfoques difieren ligeramente, con El Especialista 1 centrado en la detección de malware mediante técnicas establecidas y con El Especialista 2 orientado hacia el desarrollo de herramientas más flexibles, ambos reconocen la importancia de una estrategia integral y adaptativa. Las limitaciones de las técnicas estáticas, las ventajas de las dinámicas y los

desafíos técnicos en la implementación de soluciones efectivas son elementos clave que los expertos señalan para mejorar la seguridad en plataformas Android.

1.6.5 METODOLOGÍA DE DESARROLLO DEL PROYECTO

El Penetration Testing Execution Standard (PTES) es un marco de trabajo para llevar a cabo pruebas de penetración de manera estructurada y organizada. Proporciona una guía detallada sobre las diferentes fases de un proceso de pruebas de penetración, desde la planificación y la recopilación de información hasta la explotación de vulnerabilidades, el análisis de resultados y la presentación de informes [13]

Fase 1: Recopilación de Información

Se centra en reunir la mayor cantidad de información sobre el objetivo antes de realizar cualquier ataque. La clave es identificar el entorno, los sistemas, servicios, aplicaciones y usuarios asociados con el sistema objetivo. Técnicas como el escaneo de redes, el reconocimiento pasivo (recoger información pública) y el activo (escaneos directos) son fundamentales para entender la infraestructura y los puntos débiles potenciales.

Fase 2: Análisis de Vulnerabilidades

Se espera identificar y analizar posibles vulnerabilidades en el sistema. Se utilizan herramientas automáticas y técnicas manuales para buscar fallos de seguridad que podrían ser explotados, como configuraciones incorrectas, software desactualizado o errores de código.

Fase 3: Explotación

En esta fase se utilizan las vulnerabilidades descubiertas para obtener acceso no autorizado a los sistemas o redes del objetivo. Aquí se lanzan ataques con el objetivo de comprometer el sistema, simular cómo un atacante real aprovecharía estas debilidades para obtener acceso, ejecutar comandos maliciosos o tomar control de recursos importantes. La intención no es dañar, sino demostrar la viabilidad del ataque.

Fase 4: Post-explotación

Después de comprometer al sistema, esta fase se enfoca en evaluar el impacto del acceso conseguido. El tester determina hasta qué punto puede moverse lateralmente dentro del sistema, extraer información sensible, obtener credenciales adicionales o comprometer más recursos. Se evalúan las posibles consecuencias de una intrusión y el nivel de control que un atacante podría obtener**.**

Fase 5: Informe

Aquí implica la creación de un informe detallado que documente todo el proceso de la prueba de penetración. Se incluye los análisis de las vulnerabilidades encontradas, cómo fueron explotadas, el impacto potencial de dichas vulnerabilidades y se presentan recomendaciones para mitigar o solucionar los problemas de seguridad identificados. El informe suele incluir un resumen ejecutivo para la alta dirección y un análisis técnico detallado para los equipos de seguridad.

Figura 2: Metodología Penetration Testing Execution Standard(PTES)

CAPITULO II

2 MARCO REFERENCIAL

2.1 MARCO CONTEXTUAL

Los centros de investigaciones en el campo de seguridad informática aparecen como un aporte a la solución de los problemas que se generan en las empresas, organizaciones e instituciones, por lo que este texto constituirá en un verdadero aporte a nivel de estudiantes y profesionales.

Actualmente la auditoria informática comprende un componente importante dentro de la evaluación, control y seguridad de programas, aplicaciones y tecnología que permiten a las Instituciones llevar a cabo de manera eficaz y eficiente sus operaciones. Los datos personales, financieros y corporativos son el blanco principal de los ciberdelincuentes. Un ataque cibernético puede ir desde la intrusión a sistemas privados hasta el robo de identidad, pasando por la difusión de malware que puede paralizar infraestructuras enteras [14].

La ciberseguridad en el mundo digital ha evolucionado drásticamente debido a los avances tecnológicos que en la actual se presenta, así mismo, como aparecen nuevos enfoques informáticos, también ocurre nuevas amenazas cibernéticas que son cruciales para comprometer un sistema en común de una empresa u organización. El objetivo principal de la ciberseguridad no comprende en solo proteger los sistemas informáticos que contenga información confidencial, sino también garantizar la continuidad del negocio y de la confianza del cliente. Es importante destacar que las violaciones de seguridad informática a los sistemas que las entidades cuenta permite en su mayoría ser usados para incumplir las leyes legales como la protección de datos digitales, entre otros. [15]

La reputación de las entidades organizacionales esta en margen de ser vulneradas gracias a la filtración de información que los ciberdelicuentes adquieren luego de elaborar ataques informáticos a los sistemas informáticos, lo que permite afectar negativamente la relación con los clientes y lograr dar un declive en los procesos de negocios perdiendo grandes cantidades de suma de dinero que en un futuro sino es corregido la brecha de seguridad provocaría la quiebra en su totalidad de la enditad. Por lo tanto, es necesario invertir en

ciberseguridad ya que es esencial proteger los activos digitales y mantener a flote todo lo referente a los procesos de negocios y no perder la confianza con los clientes [16].

El malware tiene un impacto económico y social significativo, afectando a individuos, empresas y gobiernos en todo el mundo. Económicamente, genera pérdidas millonarias por la interrupción de operaciones, robos de datos y el costo de medidas de recuperación y ciberseguridad, estimándose que el costo global del cibercrimen alcanzará los 10 billones de dólares para 2025. Socialmente, compromete la privacidad, afecta servicios críticos como la salud y crea desconfianza en las tecnologías digitales. Un caso relevante es el ataque de ransomware WannaCry en 2017, que afectó a más de 200,000 computadoras en 150 países, interrumpiendo servicios de salud y causando daños valorados en miles de millones de dólares [17].

2.2 MARCO TEORICO

2.2.1 METODOLOGÍA PARA EL ANÁLISIS DE MALWARE EN UN AMBIENTE CONTROLADO

Este trabajo tiene como objetivo establecer un laboratorio seguro para realizar un análisis automatizado de muestras de malware, buscando reducir el tiempo de análisis mediante el uso de Cuckoo Sandbox. Se llevarán a cabo pruebas para evaluar el ciclo de vida del malware, sus características de propagación, los métodos de infección y la recolección de datos, así como su impacto en los sistemas comprometidos. Además, se pretende comprender las motivaciones y objetivos detrás de los ataques, con el fin de desarrollar políticas de seguridad que ayuden a las empresas a asegurar la confidencialidad, integridad y disponibilidad (CID) de sus datos. Nuestro enfoque se centra en el análisis dinámico de malware, que permite la detección en función del comportamiento del malware, lo que implica la ejecución de muestras y la observación de sus acciones en tiempo real [18].

2.2.2 MISTIC TFM: SEGURIDAD EN ANDROID – ANÁLISIS DE VULNERABILIDADES Y MALWARE.

El uso extensivo de dispositivos móviles los convierte en objetivos primordiales para ataques cibernéticos, dado que suelen almacenar información confidencial, personal y crítica. Esta

realidad ha incrementado la preocupación por la seguridad entre empresas, administraciones públicas y usuarios individuales. Este proyecto se enfoca en delinear diversos métodos y técnicas para realizar análisis de seguridad en dispositivos Android™, aplicándolos a un dispositivo específico. Se presenta como una guía práctica destinada a organizaciones que buscan garantizar la seguridad de sus dispositivos móviles y prevenir la infección por malware. En la primera sección, se describe la arquitectura de seguridad de Android y se identifican los tipos de malware comunes en esta plataforma. También se detallan los pasos para configurar un laboratorio con las herramientas necesarias para analizar dispositivos móviles y su malware [19]

2.2.3 ESTUDIO DE ATAQUES RAT EN DISPOSITIVOS MOVILES ANDROID A FUNCIONARIOS DE INSTITUCIONES PÚBLICAS QUE MANEJAN INFORMACIÓN SENSIBLE.

El teléfono móvil se ha convertido en el dispositivo más utilizado para la comunicación y el entretenimiento, ofreciendo una amplia variedad de aplicaciones. Sin embargo, también es un objetivo atractivo para los ciberdelincuentes, quienes utilizan técnicas como el phishing para engañar a los usuarios. A través de malware, como las herramientas de Administración Remota (RAT), los atacantes pueden infiltrarse en los teléfonos Android sin autorización, robando información valiosa de las víctimas o de las entidades donde trabajan. Este acceso no autorizado plantea un riesgo significativo para la seguridad personal e institucional. Nuestro estudio se centra en cómo el malware RAT permite el control remoto de los dispositivos, lo que facilita el acceso a datos sensibles. Analizamos un caso de una institución pública que maneja información clasificada, resaltando la gravedad de estos ataques [20] .

2.3 MARCO CONCEPTUAL

2.3.1 CIBERSEGURIDAD

La ciberseguridad se refiere a la protección de dispositivos, redes, aplicaciones de software, sistemas críticos y datos de amenazas digitales. Las empresas están obligadas a mantener la información confidencial para mantener la confianza de sus clientes y cumplir con las

normas legales. Para lograrlo, utilizan herramientas y técnicas de ciberseguridad creadas para evitar que personas no autorizadas accedan a datos confidenciales y reducir el riesgo de interrupciones en sus operaciones causadas por actividades maliciosas en la red. La ciberseguridad se logra mediante la optimización de personas, procesos y tecnologías que fortalecen las defensas digitales [21].

Los ciberataques pueden interrumpir, dañar o destruir empresas, comunidades y vidas, generando robos de identidad, extorsión, pérdida de datos confidenciales, interrupciones operativas y la pérdida de clientes o negocios, e incluso el cierre de empresas. Por lo tanto, la ciberseguridad es crucial. El impacto económico de estos ataques es creciente, y a medida que los atacantes se vuelven más sofisticados, se estima que para 2025 la ciberdelincuencia costará 10,5 billones de dólares anuales a nivel global [22].

2.3.2 CIBERCRIMINALIDAD

El cibercrimen es una actividad delictiva que involucra el uso de computadoras, redes informáticas o dispositivos conectados, ya sea como objetivo o como medio para cometer el delito. La mayoría de los ciberdelitos son cometidos por ciberdelincuentes o piratas informáticos con el objetivo de obtener beneficios económicos, aunque en algunos casos el objetivo es dañar sistemas por motivos personales o políticos. Tanto individuos como organizaciones pueden estar involucrados en el cibercrimen, con algunos delincuentes altamente organizados y técnicamente capacitados y otros hackers con habilidades limitadas [23].

La cibercriminalidad es un fenómeno complejo que va más allá de las fronteras físicas y el entorno virtual y representa la convergencia entre la tecnología, el crimen y la sociedad actual. El robo de datos, las estafas en línea, el ciberespionaje y el sabotaje de infraestructuras vitales son ejemplos de delitos cometidos a través de la tecnología de la información. La naturaleza transnacional de los ciberdelincuentes y su capacidad para evolucionar rápidamente frente a las medidas de seguridad la hacen particularmente difícil de combatir,

ya que aprovechan la anonimidad y el alcance global de Internet para actuar desde cualquier lugar [24].

2.3.2.1 TIPOS DE CIBERCRIMINALES

- **HACKER ÉTICOS**

  Aunque el término "hacker" con frecuencia se asocia con delitos informáticos, también hay expertos en hacking ético, quienes conocen a fondo los ciberataques y las técnicas utilizadas en ellos. Siempre con permiso previo, estos profesionales se involucran en la seguridad de sistemas para identificar fallas y ayudar a corregirlas. El hacking ético, también conocido como piratería ética, implica realizar pruebas de intrusión en los sistemas de una organización con el consentimiento de la organización para descubrir vulnerabilidades que puedan ser solucionadas antes de que los actores maliciosos puedan explotarlas [25].

- **HACKER MALICIOSOS**

  Un ciberdelincuente malicioso es un ciberdelincuente que explota fallas de seguridad o escapa de las barreras de seguridad para acceder de manera no autorizada a computadoras o redes con el fin de robar información. El hacking de este tipo tiene intenciones perjudiciales y ha llevado a una amplia economía de cibercrimen, en la que los delincuentes obtienen ganancias mediante el lanzamiento de ataques cibernéticos, la venta de malware o el comercio de datos robados. En realidad, se calcula que este mercado ilegal es de tal magnitud que ocupa el tercer lugar en tamaño en todo el mundo, siendo solo superado por las economías de Estados Unidos y China. [26]

- **HACKTIVISTAS**

  Los hacktivistas son personas o grupos que utilizan ciberataques como una forma de protesta con fines políticos, sociales o económicos. Como medio de expresión de sus causas, utilizan el hacktivismo para llevar a cabo acciones que alteran el funcionamiento de sitios web o interrumpen la accesibilidad a servicios online. Los organismos de ciberseguridad suelen distinguir entre protestas pacíficas en línea, como campañas informativas o recolección de firmas, y la intrusión no autorizada en

sistemas o sitios web, que es lo que caracteriza a las actividades hacktivistas, aunque no hay una definición universalmente aceptada. Estos movimientos pueden estar motivados por una variedad de ideologías [27].

- **CRACKERS**

  Los crackers son figuras delictivas en el ámbito tecnológico, conocidos por romper sistemas de seguridad con fines maliciosos o de beneficio personal. A diferencia de los hackers éticos, que operan dentro de la legalidad, los crackers actúan de manera clandestina, explotando vulnerabilidades para robar datos, distribuir malware o extorsionar a individuos y organizaciones. Su falta de ética y sus acciones representan una seria amenaza para la ciberseguridad global, poniendo en riesgo tanto la integridad de los sistemas como la confianza de los usuarios en la tecnología. Debido a esto, la lucha contra los crackers es una prioridad constante para las empresas y entidades de seguridad digital en todo el mundo [28].

- **PHISHERS**

  Los phishers son delincuentes cibernéticos que engañan a las personas enviando correos electrónicos falsos que parecen provenir de fuentes confiables, como bancos o empresas. Su objetivo es persuadir a los receptores para que revelen datos confidenciales como contraseñas o datos financieros. Para evitar estos engaños, escriba directamente la dirección de un sitio web en lugar de hacer clic en enlaces sospechosos [29].

- **SPAMMERS**

  Los spammers son individuos o entidades que envían comunicaciones no solicitadas de manera masiva, generalmente con fines publicitarios o promocionales. Aunque el spam se asocia comúnmente con correos electrónicos no deseados, los spammers también utilizan otros canales como mensajes instantáneos, SMS, redes sociales e incluso mensajes de voz. Estas prácticas, conocidas como "spamming," son ilegales en muchas jurisdicciones debido a su naturaleza invasiva y su impacto negativo en los usuarios y las plataformas de comunicación [30].

- **SCRIPT KIDDIES**

  Los aficionados a la ciberseguridad llamados Script Kiddies intentan realizar ciberataques utilizando herramientas y scripts creados por hackers más experimentados sin tener mucho conocimiento. A menudo recurren a ataques de denegación de servicio (DoS) para desestabilizar sitios web o servicios en línea, ya que su principal motivación es ganar notoriedad o causar interrupciones. A pesar de sus limitaciones, sus acciones todavía pueden causar daños significativos en el mundo digital [31].

- **ESTAFADORES EN LINEA**

  Los delincuentes que utilizan dispositivos digitales y plataformas de internet para llevar a cabo actividades fraudulentas se conocen como estafadores en línea. Estos engaños pueden incluir una variedad de engaños, como phishing, suplantación de identidad o la venta de productos falsos o inexistentes. Aprovechan la confianza de los usuarios en los servicios en línea para obtener beneficios personales, afectando a millones de personas en todo el mundo. Es fundamental comprender los diferentes tipos de estafas y cómo protegerse para evitar ser víctima de ellas [32].

2.3.3 VULNERABILIDAD

Una infracción de seguridad se refiere a un suceso que facilita el acceso no permitido a datos, aplicaciones, redes o equipos de computación, y generalmente sucede cuando un intruso consigue eludir las medidas de seguridad establecidas. Esta circunstancia se diferencia de una infracción de datos, que ocurre cuando los ciberdelincuentes adquieren datos sensibles; por ejemplo, una infracción de seguridad podría equipararse a la intrusión en un inmueble, mientras que una violación de datos sería similar al hurto de una cartera o un ordenador portátil. La información sensible, frecuentemente vendida en la web oscura, posee un valor considerable, y las infracciones de seguridad pueden generar un costo medio de casi 4 millones de dólares para las grandes compañías [34].

Por otro lado, una vulnerabilidad de seguridad se define como una falla o debilidad en la estructura, funcionalidad o implementación de una red o activo en red que los hackers pueden explotar para realizar ciberataques, acceder sin autorización a sistemas o datos, o dañar a una organización. Ejemplos comunes incluyen configuraciones incorrectas en firewalls que permiten la entrada de malware y errores no corregidos en el protocolo de escritorio remoto que podrían facilitar el control de un dispositivo por parte de piratas informáticos. Dada la alta distribución de las redes empresariales y el descubrimiento constante de nuevas vulnerabilidades, la gestión manual de estas resulta casi inviable, por lo que los equipos de ciberseguridad a menudo recurren a soluciones automatizadas para la gestión de vulnerabilidades [35].

2.3.4 AMENAZA INFORMÁTICA

Las amenazas digitales aluden a cualquier acción dañina o potencialmente dañina dirigida a sistemas de computación, redes o datos, y pueden surgir de diferentes fuentes, como hackers, trabajadores insatisfechos o fallos involuntarios. Con el progreso tecnológico, estas amenazas han experimentado una evolución y se han vuelto más complejas; por ejemplo, la inteligencia artificial generativa y el Internet de las Cosas (IoT) han generado nuevas posibilidades para los ataques cibernéticos. Esto aumenta las amenazas a la ciberseguridad y subraya la importancia de disponer de recursos de seguridad informática totalmente fiables [36].

Las amenazas a los recursos de información en una entidad son múltiples, y identificarlas y asignarles prioridad puede resultar complicado. De acuerdo con la norma ISO/IEC 27001, una amenaza se define como un suceso que provoca un incidente, causando perjuicios materiales o pérdidas intangibles en los recursos de información. INCIBE también caracteriza la amenaza como toda acción que explote una vulnerabilidad y pone en riesgo la seguridad del sistema. En conclusión, cualquier elemento que pueda impactar en el funcionamiento principal de una organización debe ser visto como una amenaza, cuya categorización se basará en la actividad principal del negocio y en un inventario minucioso de activos [37].

2.3.5 RIESGO TECNOLÓGICO

El riesgo tecnológico alude a una diversidad de problemas que surgen del empleo de la tecnología, tales como averías de hardware, fallos en el software, obsolescencia y problemas de compatibilidad. No todos estos peligros surgen de acciones malintencionadas; muchos de ellos son resultado de fallos humanos, errores en los sistemas, o una ausencia de mantenimiento y renovación. Este peligro se enfoca en el efecto que los inconvenientes técnicos pueden causar en la disponibilidad y desempeño de sistemas y servicios, considerando elementos vitales como la continuidad de la empresa y la recuperación de desastres [38].

Es esencial la administración de riesgos tecnológicos para salvaguardar la innovación y garantizar la continuidad del negocio en la época digital. Esto conlleva reconocer de manera proactiva amenazas tecnológicas, desde ataques cibernéticos hasta interrupciones de sistemas, para reaccionar de manera eficaz. Además, es necesario desarrollar una sólida resiliencia ante las amenazas cibernéticas, facilitando una pronta recuperación frente a incidentes. La adopción de tecnologías emergentes conlleva tanto oportunidades como riesgos, por lo que numerosas compañías de Latinoamérica intentan balancear dicha adopción con una correcta administración de riesgos, garantizando que se encuentre en sintonía con las metas empresariales [39].

2.3.6 ANDROID

Android es un sistema operativo creado originalmente para smartphones, parecido a iOS, Symbian y Blackberry OS, y se caracteriza por su base en Linux, un núcleo de código abierto, libre y multiplataforma. Facilita la creación de aplicaciones mediante una versión de Java denominada Dalvik, ofreciendo todas las interfaces requeridas para que los programadores puedan utilizar las funciones del dispositivo, tales como el GPS, las llamadas y la agenda, de forma fácil usando un lenguaje de programación ampliamente reconocido [37].

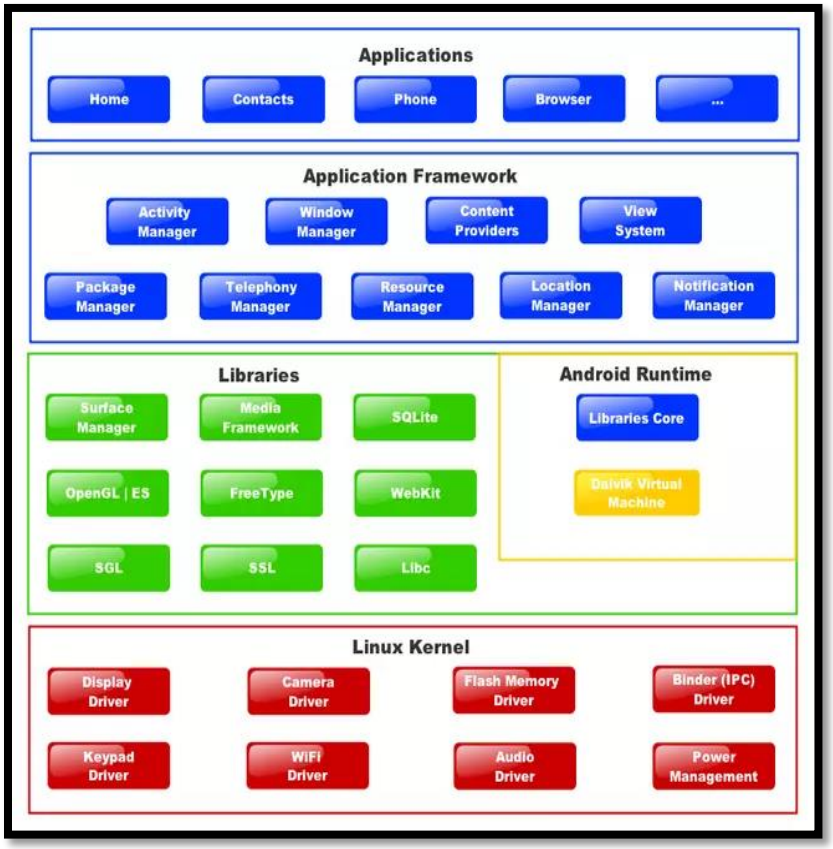


Figura 3: Arquitectura del Sistema Operativo Android Fuente: [37]

2.3.7 CIBERATAQUES

Un ciberataque se refiere a cualquier esfuerzo intencional de robar, revelar, alterar, desactivar o aniquilar datos, aplicaciones u otros recursos a través del acceso no permitido a redes, sistemas de computación o dispositivos digitales. Los responsables de amenazas llevan a cabo ciberataques por múltiples motivos, que oscilan entre hurtos menores hasta conflictos bélicos, empleando estrategias como la infiltración de malware, estafas de ingeniería social y el hurto de credenciales para conseguir acceso a sus metas [38].

Un ciberataque es una acción llevada a cabo por ciberdelincuentes con el objetivo de desactivar sistemas, sustraer datos o utilizar un sistema vulnerable a ataques futuros. El incremento en la complejidad de estos ataques en años recientes ha hecho que su prevención sea esencial para personas y organizaciones. El triunfo del cibercrimen se basa en el uso de vulnerabilidades en los sistemas; mientras que los sistemas de seguridad deben resguardar cada punto de acceso, los atacantes solo requieren explotar una debilidad concreta. Esta

desigualdad favorece a los atacantes y reta incluso a grandes entidades a evitar el ingreso no permitido a sus redes [42].

2.3.8 ¿QUÉ ES EL MALWARE?

El malware se refiere a programas maliciosos creados con la intención deliberada de causar daño a dispositivos y sistemas informáticos, mientras que los errores de software son fallos no intencionados que pueden ocasionar daños. A menudo, hay confusión sobre la distinción entre un virus y el malware en general. En realidad, el malware es un término amplio que abarca diversas amenazas en línea, como virus, spyware, adware, ransomware y otros tipos de software malicioso, mientras que un virus informático es simplemente una de las muchas formas de malware existentes [33].

El malware puede provocar daños significativos al penetrar sistemas con contraseñas débiles, propagarse por redes y interrumpir operaciones empresariales. Además de bloquear archivos y enviar correos no deseados, el malware es responsable de ciberataques generalizados, incluidos robos de identidad y fraudes. Los ataques de ransom

ware, por ejemplo, causan pérdidas millonarias y son perpetrados por hackers contra usuarios, empresas y gobiernos [34].

2.3.8.1 TIPOS DE MALWARE

- **PHISHING**

  El phishing ocurre a través de correos electrónicos, llamadas o mensajes de texto, en los que los estafadores se hacen pasar por entidades legítimas para engañar a las personas enviando mensajes que parecen urgentes y auténticos para que compartan información confidencial como contraseñas o números de tarjetas de crédito. Estos ataques suelen dirigir a las víctimas a sitios web falsos que parecen reales, donde se les pide que ingresen información confidencial. Los estafadores pueden obtener información valiosa a través de una fachada bien elaborada y la presión para actuar rápidamente, lo que deja a las víctimas vulnerables al robo de identidad y pérdidas financieras [35].

- **SPYWARE**

  El spyware se caracteriza por su invisibilidad, lo que lo convierte en una amenaza importante, ya que mientras más tiempo pasa sin ser detectado, mayor es el daño que puede causar al recopilar datos personales del usuario. A pesar de que se usa principalmente con fines maliciosos, hay usos legítimos. Las empresas, por ejemplo, pueden utilizar software de monitoreo para monitorear el uso de dispositivos por parte de sus empleados, para proteger información confidencial o para controlar la productividad. De manera similar, los controles parentales que limitan el uso del dispositivo y bloquean contenido inapropiado también se consideran una forma de spyware [36].

- **ADWARE**

  El adware es un tipo de software malicioso que envía publicidad no deseada a la computadora del usuario. No solo es invasivo, sino que también recopila datos personales y registra los sitios web visitados para mostrar anuncios personalizados. Los ciberdelincuentes obtienen ganancias cuando los usuarios hacen clic en anuncios falsos y también pueden vender los datos de navegación a terceros para obtener ingresos adicionales, por lo que su objetivo principal es económico [37].

- **RANSOMWARE**

  El ransomware es un tipo de ciberataque en el que los delincuentes bloquean el acceso a los datos o dispositivos de una víctima hasta que se pague un rescate. Según el IBM Security X-Force Threat Intelligence Index 2023, en 2022, los ataques de ransomware representaron el 17 % de todos los ciberataques. Originalmente, estos ataques simplemente exigían un pago a cambio de una clave de cifrado, pero en los últimos años han evolucionado hacia técnicas más sofisticadas, como la doble extorsión, en la que se amenaza con divulgar los datos robados, y la triple extorsión, que involucra a los clientes o socios comerciales de la víctima para aumentar la presión [38].

- **TROYANOS**

  Un caballo de Troya, también conocido como troyano, es un tipo de malware que se disfraza de software real para engañar a los usuarios y lograr que se instale en sus

sistemas. Los ciberdelincuentes utilizan troyanos para obtener acceso no autorizado a los dispositivos y, con frecuencia, emplean técnicas de ingeniería social para convencer a los usuarios de ejecutar el programa malicioso. El troyano, una vez que se activa, permite a los atacantes espiar al usuario, robar datos confidenciales y abrir una puerta trasera que permite el acceso continuo al sistema comprometido [39].

- **GUSANOS**

  Los gusanos informáticos son particularmente peligrosos porque pueden propagarse por sí mismos. A diferencia de los troyanos, que necesitan ser activados, pueden expandirse rápidamente a través de una red una vez que infectan una máquina anfitriona sin necesidad de intervención del usuario. Estos gusanos se infiltran y operan sin el conocimiento del usuario aprovechando las vulnerabilidades del sistema operativo. Los hackers los crean con el fin de infiltrarse en el sistema objetivo y llevar a cabo actividades maliciosas sin ser detectados [40].

- **VIRUS**

  Un tipo de malware llamado virus informáticos tiene una amplia gama de formas, métodos de propagación y efectos. Estos virus pertenecen a dos grupos: el primero infecta y replica inmediatamente al ingresar en un sistema, mientras que el segundo permanece inactivo hasta que el usuario ejecuta involuntariamente el código malicioso, iniciando su comportamiento dañino [41].

- **KEYLOGGER**

  Un tipo de malware que registra y almacena todas las pulsaciones de teclas realizadas en un dispositivo se conoce como keylogger. Se considera una forma de spyware porque espia a los usuarios de forma encubierta. Es extremadamente invasivo registrar todo lo que se escribe. Los registradores de teclas son de dos tipos: de software, que es el tipo más utilizado, y de hardware, que requiere acceso físico al dispositivo para funcionar [42].

- **BOTNETS**

  Un botnet es una red de computadoras comprometidas que los ciberdelincuentes controlan para realizar fraudes y ataques cibernéticos. El término "botnet" proviene de las palabras "robot" y "red". Estos dispositivos infectados, también conocidos

como bots, tienen la capacidad de automatizar ataques masivos como el robo de datos, la sobrecarga de servidores y la propagación de malware sin el consentimiento del propietario del dispositivo. La creación de un botnet es normalmente solo la primera etapa de un ataque cibernético más complejo [43].

- **MALWARE DE PUP**

  Un programa potencialmente no deseado, también conocido como PUP, es un tipo de software que suele instalarse junto con otros programas descargados y no ofrece ventajas evidentes para el usuario. Debido a que pueden actuar como adware o spyware, con frecuencia se consideran aplicaciones no deseadas, y en algunos casos también se les conoce como PUA, o aplicaciones potencialmente no deseadas [44].

- **MALWARE SIN ARCHIVOS**

  El malware sin archivos es un tipo de software malicioso que opera directamente desde la memoria del sistema en lugar de almacenarse en el disco, lo que lo hace más difícil de detectar y eliminar. Debido a la complejidad que presenta para las soluciones de seguridad, este tipo de malware ha ganado popularidad desde 2017. El malware sin archivos se diferencia de las amenazas convencionales mediante el uso de métodos sofisticados, como el almacenamiento de scripts maliciosos en Windows Management Instrumentation (WMI) o su ejecución directa en PowerShell. También puede ejecutarse en la memoria o ocultarse en el registro del sistema operativo [45].

2.3.9 DETECCIÓN DE MALWARE ESTÁTICO

La detección estática de malware implica analizar el código o archivo malicioso sin ejecutarlo, utilizando técnicas como la inspección del encabezado del archivo, los metadatos, el hash, la firma o el contenido del código. Este enfoque es rápido y eficaz para identificar variantes de malware conocidas o similares. Sin embargo, puede ser fácilmente eludido por malware que emplea técnicas como cifrado, compresión, empaquetado o polimorfismo para modificar su apariencia. Además, la detección estática no proporciona información sobre el comportamiento real del malware ni su impacto en el sistema o la red [46].

2.3.10 DETECCIÓN DE MALWARE DINÁMICO

El análisis dinámico del malware implica ejecutar el software malicioso en un entorno controlado, como una máquina virtual, para observar su comportamiento en tiempo real. Este enfoque proporciona una visión detallada de las acciones del malware, lo que permite identificar técnicas de evasión y funcionalidades ocultas. Es esencial para comprender completamente las capacidades del malware y desarrollar medidas efectivas de detección y mitigación.

2.3.11 RAT (REMOTE ACCESS TROJAN)

Las RAT, o troyanos de acceso remoto, fueron inicialmente concebidas con la loable finalidad de simplificar la gestión de ajustes y aplicaciones a distancia, particularmente útiles para el personal de soporte técnico. Sin embargo, en manos de delincuentes cibernéticos, estas herramientas se transforman en armas altamente destructivas. Al permitir la instalación de un troyano en tu dispositivo móvil, estás concediendo acceso remoto a un individuo desconocido, equiparable a entregar las llaves de tu hogar a un extraño, lo que puede tener consecuencias devastadoras para la seguridad y privacidad de tus datos personales y empresariales [47].

2.4 HERRAMIENTAS

MobSof: Es una herramienta de análisis de aplicaciones móviles que permite examinar y evaluar la seguridad de aplicaciones para dispositivos móviles. MobSof puede utilizarse para identificar vulnerabilidades, analizar el comportamiento de las aplicaciones y detectar posibles amenazas de seguridad [48].

Android Studio: Es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Proporciona herramientas avanzadas para la creación, depuración y optimización de aplicaciones móviles para la plataforma Android. Android Studio incluye un editor de código, herramientas de diseño de interfaz de usuario, emuladores de dispositivos Android y muchas otras características útiles para desarrolladores de aplicaciones móviles [49].

BlueStacks: Es un emulador de Android que permite ejecutar aplicaciones y juegos Android en computadoras con sistemas operativos Windows y macOS. BlueStacks proporciona una

experiencia similar a la de un dispositivo Android real, lo que permite a los usuarios probar y utilizar aplicaciones móviles en sus computadoras personales [50].

Cuckoo: Es un sistema de análisis de malware automatizado que permite ejecutar archivos sospechosos en un entorno controlado y monitorear su comportamiento para detectar actividades maliciosas. Cuckoo es ampliamente utilizado en la industria de la seguridad cibernética para analizar y clasificar muestras de malware [51].

Drozer: Es una herramienta de pruebas de penetración para dispositivos Android que permite identificar y explotar vulnerabilidades de seguridad en aplicaciones y sistemas operativos Android. Drozer proporciona una serie de módulos y funciones que facilitan la evaluación de la seguridad de dispositivos Android y la realización de pruebas de penetración [52].

Rennux: Es una distribución de Linux diseñada específicamente para pruebas de penetración y análisis forense. Rennux incluye una amplia gama de herramientas de seguridad cibernética y análisis forense que permiten a los profesionales de la seguridad realizar evaluaciones exhaustivas de la seguridad de sistemas y redes [53].

2.5 MARCO LEGAL

CÓDIGO ORGANICO INTEGRAL PENAL

El código Organizo Integral Penal(COIP) manifiesta su creación con el objetivo de imponer la necesidad de unificar en un solo texto la legislación de carácter punitivo, con el afán de dispersar una seguridad jurídica en el ordenamiento jurídico ecuatoriano.

CAPÍTULO TERCERO – MEDIOS DE PRUEBA

El Art. 500.- Contenido digital. [54] Es todo acto informático que representa hechos, información o conceptos de la realidad, almacenados, procesados o transmitidos por cualquier medio tecnológico que se preste a tratamiento informático, incluidos los programas diseñados para un equipo tecnológico aislado, interconectado o relacionados entre sí. En la investigación se seguirán las siguientes reglas.

LEY DEL SISTEMA NACIONAL DE REGISTRO DE DATOS PÚBLICOS

La ley del Sistema Nacional de Registro de Datos Públicos tiene como afán de ley garantizar la seguridad jurídica, organizar, regular, sistematizar e interconectar la información, como así mismo, la eficacia y eficiencia en el manejo, su publicidad, transparencia, acceso e implementaciones de nuevos enfoques tecnológicos [55].

CAPÍTULO III

NORMAS GENERALES APLICADAS A LOS REGISTROS PUBLICOS

El Art. 12.- Medios Tecnológicos. El Estado [56] a través del ministerio sectorial con competencia en las telecomunicaciones y en la sociedad de la información, definirá las políticas y principios para la organización y coordinación de las acciones de intercambio de información y de bases de datos entre los organismos e instancias de registro de datos públicos, cuya ejecución y seguimiento estará a cargo de la Dirección Nacional de Registro de Datos Públicos. La actividad de registro se desarrollará utilizando medios tecnológicos normados y estandarizados, de conformidad con las políticas emanadas por el ministerio sectorial de las telecomunicaciones y de la sociedad de la Información.

LEY DE PROTECCION DE DATOS PERSONALES

La ley de protección de datos personales, es conocido como una norma jurídica que tiene como objetivo garantizar y proteger el tratamiento de datos personales, libertades públicas y derechos fundamentales de las personas físicas, especialmente en su honor e intimidad personal y familiar [57].

CAPÍTULO 1: ÁMBITO DE APLICACIÓN INTEGRAL

Art 1.- Objetivo y finalidad: Establece que el propósito de la ley es asegurar el derecho a la protección de los datos personales, lo que incluye el acceso, control y resguardo de dicha información. Para ello, regula y desarrolla principios, derechos, obligaciones y mecanismos de protección adecuados [57].

Art 5.- Integrantes del sistema de protección de datos personales: Se define que los integrantes del sistema de protección de datos personales son: el titular, el responsable del tratamiento, el encargado del tratamiento, el destinatario, la autoridad de protección de datos y el delegado de protección de datos personales [57].

Art 25.- Categorías especiales de datos personales: Clasifica como categorías especiales de datos personales a los datos sensibles, los datos de menores de edad, los datos relacionados con la salud y los de personas con discapacidad o sus representantes, en relación con dicha condición [57].

Art 37.- Seguridad de datos personales: Señala que quienes gestionen o procesen datos personales deberán adherirse al principio de seguridad de los mismos, tomando en cuenta factores como el tipo y cantidad de datos, el estado de la tecnología, las mejores prácticas de seguridad y los costos de implementación, evaluando los riesgos posibles [57].

Art 38.- Medidas de seguridad en el ámbito del sector público: El sistema de protección de la información establecido por el gobierno debe contener las acciones requeridas para administrar el manejo de la información personal, con el objetivo de lidiar con cualquier peligro, amenaza, vulnerabilidad, accesos no permitidos, pérdidas, alteraciones, aniquilación o divulgación accidental o ilícita de los datos, en conformidad con el principio de seguridad de la información personal [57].

CAPÍUTLO III

3 PROPUESTA

3.1.1 DESARROLLO

3.1.1.1 FASE I: CONFIGURACIÓB DEL LABORATORIO VIRTUAL

Para el proyecto de análisis de malware de tipo RAT en plataformas Android, se empleará una máquina virtual denominada “Prueba Hacking”, que utiliza una distribución de Linux optimizada para este tipo de actividades. Esta configuración permitirá realizar tanto análisis estático como dinámico del malware en un entorno controlado y seguro, garantizando una experiencia coherente y comprender el arte que emana todo lo referente de malware.

La instalación del laboratorio comprende en la configuración de una máquina virtual correspondiente a la versión Linux accesible para emplear una imagen iso de Ubuntu y establecer la configuración pertinente para la creación del entorno donde se desarrolla la instalación de las herramientas y descargas de malware como pruebas de seguridad. Una vez creado la máquina virtual se procede a instalar la herramienta MobSf a través del repositorio Github, se elabora la clonación de la herramienta y se instala los requerimientos necesarios de los archivos de instalación y se eleva el servicio a través del navegador para preparar el análisis de malware. (Ver Anexo III: Entorno Mobsf– Estático)

La instalación del laboratorio Drozer implica la configuración pertinente de Python versión 2.7.18 para efectuar la instalación de librerías necesarias para la interaccion desde el computador y dispositivo móviles con la finalidad de ejercer la evaluación dinámica o análisis de aplicaciones Android, como a su vez la instalación de herramienta como ADB & fastboots para la escucha y movimientos de acciones en tiempo real en el sistema operativo Android con la finalidad de encontrar brechas, comportamiento de las apk mediante una análisis dinámico (Ver Anexo III: Entorno Drozer – Dinamico).

3.1.1.2 FASE II: ELECCIÓN DE LA MUESTRA

En la presente fase corresponde en la exploración de las técnicas estática y dinámicas para el análisis de Malware permitiendo emplear parámetros de agrupación de clasificación como es el vector de ataque, sistemas en peligro, entre otros. Además, también cuenta con la categorización de los Malware´s más comunes en dispositivo móviles y elaborar una categorización de su Origen RAT, y conocer al fondo sobre su arte.

A continuación, se presenta un cuadro correspondiente de las técnicas estáticas y dinámicas, como a su vez el cuadro de los Malware´s para su arte en el área informática

CUADRO DESCRIPTIVO DE TÉCNICAS ESTÁTICAS PARA MALWARE

	TÉCNICA	VECTOR DE ATAQUE	SISTEMAS EN PELIGRO	VULNERABILIDADES A PROVOCAR	NIVEL DE CRITICIDAD
ANÁLISIS ESTÁTICO	**Desensamblado**	Archivos ejecutables	Sistemas operativos, aplicaciones	Ejecución de código malicioso	Alta
	Ingeniería Inversa	Códigos fuente, binarios	Sistemas operativos, aplicaciones	Identificación de funciones vulnerables	Alta
	Hashing	Archivos ejecutables	Sistemas de archivo, antivirus	Evasión de detección por firma	Media
	Firmas YARA	Archivos ejecutables, documentos	Sistemas operativos, aplicaciones	Modificación de archivos para evasión	Media
	Análisis de Dependencias	Archivos ejecutables, bibliotecas dinámicas	Sistemas operativos, aplicaciones	Ejecución de código en bibliotecas compartidas	Media
	Análisis de Strings	Archivos ejecutables, documentos	Sistemas operativos, aplicaciones	Revelación de información sensible	Baja

Tabla 1: Cuadro descriptivo de técnicas de Análisis Estático para Malware

CUADRO DESCRIPTIVO DE TÉCNICAS DINÁMICAS PARA MALWARE

	TÉCNICA	VECTOR DE ATAQUE	SISTEMAS EN PELIGRO	VULNERABILIDADES A PROVOCAR	NIVEL DE CRITICIDAD
ANÁLISIS DINÁMICO	**Sandboxing**	Archivos ejecutables, scripts	Entornos virtuales, máquinas físicas	Ejecución de comandos maliciosos	Alta
	Monitorización de Red	Tráfico de red, comunicaciones	Infraestructura de red, servidores	Exfiltración de datos, ataques DDoS	Alta
	Debugging	Archivos ejecutables	Sistemas operativos, aplicaciones	Modificación de flujo de ejecución	Media
	Emulación	Archivos ejecutables	Entornos virtuales	Análisis sin afectar el entorno real	Media
	Análisis de Comportamiento	Archivos ejecutables, scripts	Sistemas operativos, aplicaciones	Actividades maliciosas en tiempo real	Alta
	Monitorización de Sistema	Actividad del sistema, llamadas API	Sistemas operativos, aplicaciones	Modificación de registros, archivos	Alta

Tabla 2: Cuadro descriptivo de técnicas dinámica de análisis para Malware

CUADRO DESCRIPTIVO DE MALWARE MÁS COMUNES EN S.O ANDROID MÓVIL

TIPO DE MALWARE	MÉTODO DE PROPAGACIÓN	SÍNTOMAS	IMPACTO	VULNERABILIDADES EXPLOTADAS	NIVEL DE CRITICIDAD
Troyano (Trojan)	Aplicaciones falsas, descargas	Reducción de rendimiento, aplicaciones desconocidas	Robo de datos, control remoto	Permisos excesivos, aplicaciones de terceros	Alta
Spyware	Aplicaciones disfrazadas, enlaces maliciosos	Consumo de batería, uso de datos	Robo de información personal y financiera	Acceso a datos personales, cámaras, micrófono	Alta
Adware	Aplicaciones gratuitas, anuncios	Anuncios intrusivos, redirección de navegador	Reducción de rendimiento, molestias al usuario	Permisos de publicidad, aplicaciones gratuitas	Media

Ransomware	Aplicaciones fraudulentas, correos electrónicos	Bloqueo de pantalla, mensajes de rescate	Pérdida de acceso a datos, demanda de pago	Vulnerabilidades de seguridad, descargas no seguras	Alta
Rootkit	Aplicaciones maliciosas, scripts	Difícil de detectar, alteración de configuraciones del sistema	Control total del dispositivo, ocultación de actividades	Escalada de privilegios, vulnerabilidades del sistema	Alta
Keylogger	Aplicaciones de teclado, enlaces maliciosos	Registro de teclas, ralentización	Robo de contraseñas, información confidencial	Acceso a la entrada de teclado, permisos de aplicaciones	Alta
Botnet	Aplicaciones maliciosas, exploits	Ralentización, consumo de datos	Control remoto del dispositivo, ataques DDoS	Vulnerabilidades de red, escalada de privilegios	Alta
Worm (Gusano)	Mensajes de texto, enlaces maliciosos	Reducción de rendimiento, propagación automática	Consumo de recursos, propagación a otros dispositivos	Explotación de vulnerabilidades de red, aplicaciones de mensajería	Media
Phishing	Enlaces maliciosos, correos electrónicos	Solicitudes de información personal	Robo de identidad, fraude financiero	Ingeniería social, enlaces maliciosos	Alta

Tabla 3: Cuadro Descriptivo de Malware comunes en Dispositivos Móviles Android

3.1.1.3 FASE III: ANÁLISIS DE MALWARE

El cuadro muestra un análisis estático detallado de tres aplicaciones Android (prueba.apk, magis-video.apk y AndroidKeylogger.apk), destacando aspectos clave relacionados con la seguridad, permisos, componentes exportados, configuraciones de SDK y vulnerabilidades detectadas. Este análisis proporciona una visión general de las características técnicas y riesgos asociados con cada aplicación, lo que es fundamental para evaluar su comportamiento y posibles impactos en la seguridad del dispositivo.

CUADRO DESCRIPTIVO DE TÉCNIA ESTÁTICO SOBRE MALWARE RAT ESTUDIO

ASPECTO TÉCNICO	**prueba.apk**	**magis-video.apk**	**AndroidKeylogger.apk**
HASH MD5	f38bf1f6fa356fc849d4ee67c4790ac3	2227d9653b5591b275991ed8aa089397	cc6fd60cb1c9d65bdd3d5fa1cc3d242e
PUNTUACIÓN DE SEGURIDAD	49/100	48/100	33/100
PERMISOS PELIGROSOS	Acceso a geolocalización, llamadas, cámara, contactos	Obtener tareas, leer/modificar almacenamiento externo, instalar paquetes adicionales	Acceso a red e Internet
PERMISOS ABUSIVOS	ACCESS_FINE_LOCATION, CALL_PHONE, CAMERA, READ_CONTACTS	GET_TASKS, READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE, REQUEST_INSTALL_PACKAGES	ACCESS_NETWORK_STATE, INTERNET
COMPONENTES EXPORTADOS	Actividad: .MainActivity Servicio: .MainService Receptor: .MainBroadcastReceiver	Actividades: 89 (1 exportada) Servicios: 25 (0 exportados) Receptores: 8 (1 exportado) Proveedores: 3 (0 exportados)	Actividad: .MainActivity Servicio: .Keylogger (1 exportado)
VERSIÓN MÍNIMA DE SDK	10	19	15

VERSIÓN OBJETIVO DE SDK	No especificada	28	30
CERTIFICADO Y FIRMA	Firma de depuración (RSA con SHA-1, vulnerable a colisiones)	Certificado SHA-256 con vulnerabilidad Janus en dispositivos antiguos	Firma RSA con SHA-1, expuesta a colisiones y Janus
VULNERABILIDADES DE MANIFIESTO	• Permite instalación en Android sin parches • Recepción de transmisiones no segura	• Instalación permitida en versiones sin soporte • Expone receptores y servicios sin protección adecuada	• Instalable en Android sin soporte de seguridad • Depuración habilitada
VULNERABILIDADES DE CÓDIGO	• Uso de SSL con fijación de certificado • Hash SHA-1 para integridad (inseguro) • Generador de números no seguro	Ningún problema específico detectado	• Registra información sensible • Modo de depuración activo
API Y OPERACIONES SENSIBLES	- Carga de clases y DEX dinámicos - Conexiones HTTPS - Operaciones locales de E/S	- IPC - Carga de código nativo - Operaciones locales de E/S	- Ejecución de comandos OS - IPC - Operaciones locales de E/S
PROTECCIONES EN BINARIOS	Ninguna	- NX activado - Canarios de pila habilitados - Relocación de código parcial	No se encontraron librerías compartidas

CONEXIÓN DE RED	No se especifican	Conexiones a servidores ubicados en China y Hong Kong	Comunicación con servidor ubicado en EE.UU. (cs460-android-keylogger.appspot.com)
	APK PRUEBA	APK MAGISTV	APK KEYLOGGER

Tabla 4: Cuadro descriptivo de Malware – Análisis Estático

El cuadro descriptivo presentado resume el análisis dinámico realizado sobre tres aplicaciones Android utilizando **Drozer**, una herramienta de seguridad enfocada en detectar y explotar vulnerabilidades en aplicaciones móviles. Cada fila detalla los elementos analizados, como broadcast receivers, actividades exportadas y permisos solicitados, mostrando los comandos utilizados, resultados

CUADRO DESCRIPTIVO ANÁLISIS DINÁMICO DE MALWARE´S DE ESTUDIO

ELEMENTO ANALIZADO	BROADCAST	ACTIVIDADES EXPORTADAS	PERMISOS SOLICITADOS	INICIO DE ACTIVIDADES	MANIFEST DEL PAQUETE	EXPLOTACION DE CONTENIDOS	DETALLE
App 1: com.metasploit.stage	com.metasploit.stage.MainBroadcastReceiver Permiso: null	com.metasploit.stage.MainActivity	Error en ejecución. Se recomienda explorar permisos en AndroidManifest.xml.	Comportamiento esperado, pero requiere parámetros específicos.	Archivo AndroidManifest.xml analizado. Información completa de permisos y configuraciones.	Dependiendo del URI, la app podría exponer datos sensibles.	PRUEBA.APK

App 2: com.msandroid.mobile	com.google.firebase.iid.FirebaseInstanceIdReceiver Permission: com.google.android.c2dm.permission.SEND androidx.work.impl.diagnostics.DiagnosticsReceiver Permission: android.permission.DUMP	com.mobile.brasiltv.activity.SplashAty com.mobile.brasiltv.activity.MainAty com.umeng.message.UMessageNotifyActivity Target Activity: com.umeng.message.notify.UPushMessageNotifyActivity	Los permisos solicitados de la aplicacion se presenta al efectuar la informacion del package, mostrando un total de 21 permisos que rwquiere para su funcionamiento, que a su vez define acciones de extracción de datos	Comportamiento esperado, pero requiere parámetros específicos.	El archivo en cuestión de AndroidManifestxml analizado presenta sin duda alguna todas las configuraciones como permisos correspondiente a la aplicación al momento de su ejecución y movimiento	No se encontraron URIs de contenido accesibles. Los proveedores detectados (Content Providers) no están exportados o requieren permisos adicionales para acceder, lo cual es una configuración segura."	MAGISTV.APK
App 3: com.googl.speed	No cuentan con Broadcast receivers	com.bshu2.androidkeylogger.MainActivity	Los permisos que manifiesta el package son dos que es el del internet y el Access_network_state, adicional cuenta con una direccion path de donde se encuentra una base.apk	Comportamiento esperado, pero requiere parámetros específicos.	Se analizó el archivo AndroidManifest.xml, obteniendo detalles completos sobre los permisos y configuraciones de la aplicación.	No accessible content URIs found.	ANDROIDKEYLLOGER.APK

Tabla 5: Cuadro descriptivo de Malware – Análisis Dinámico

3.1.1.4 FASE IV: PRUEBAS DE LABORATORIOS

En la siguiente fase de pruebas específicas y monitoreo del comportamiento del malware, se analizaron el payload de msfvenom y LokiBoard en un entorno controlado. Estas pruebas permitieron evaluar sus métodos de propagación y capacidades maliciosas:

- **Msfvenom - Payload** logró infiltrarse en el dispositivo, exfiltrar datos sensibles (contactos, SMS, registros de llamadas y archivos multimedia) y establecer control remoto mediante conexión inversa.
- **LokiBoard** capturó pulsaciones de teclado, números telefónicos y mensajes enviados, además de automatizar el envío de mensajes a un número específico.

El cuadro sintetiza estos resultados, proporcionando una visión clara de los mecanismos utilizados y los datos comprometidos.

CUADRO DESCRIPTIVO DE ESCENARIOS DE PRUEBAS

ASPECTO	PAYLOAD DE MSFVENOM	LOKIBOARD
DESCRIPCIÓN	Generador de payloads maliciosos diseñados para explotar vulnerabilidades o ejecutar acciones maliciosas en dispositivos comprometidos.	Malware que se disfraza como un teclado legítimo, diseñado para capturar información y mantener persistencia en el dispositivo.
MÉTODOS DE PROPAGACIÓN	- **Ingeniería social**: Envío de archivos o enlaces maliciosos. - **Explotación de vulnerabilidades**: Aprovecha	- **Distribución en tiendas de apps**: Publicado como una app funcional.

	fallos en apps o sistemas. - **Inyección**: Modificación de APKs legítimas.	- **Loaders**: Instalado mediante otras aplicaciones. - **Privilegios elevados**: Solicita permisos excesivos.
EJECUCIÓN INICIAL	- Establece una conexión inversa con el atacante (reverse shell).	- Se ejecuta como un teclado funcional para ganar la confianza del usuario.
PERSISTENCIA	- Puede ser reactivado mediante scripts adicionales o tareas programadas. - Usualmente temporal si no se combina con otros métodos de persistencia.	- Modifica configuraciones para iniciarse automáticamente. - Mantiene permisos elevados para operar sin restricciones.
TÉCNICAS DE EVASIÓN	- Ofuscación del payload para evitar detección por antivirus. - Cifrado de la carga maliciosa. - Uso de empaquetadores personalizados.	- Disfrazado como una app legítima. - Solicitud de permisos extensivos para minimizar sospechas.
CAPACIDADES	- Robo de información. - Control remoto del dispositivo. - Movimiento lateral en redes comprometidas. - Captura de audio y video.	- Registro de pulsaciones de teclado (keylogging). - Robo de credenciales sensibles. - Transmisión de datos a servidores controlados por atacantes.

RESULTADOS DEL ESCENARIO DE PRUEBA	**Información obtenida:** - Contactos del dispositivo. - Mensajes SMS recibidos y enviados. - Historial de llamadas. - Archivos multimedia como fotos y videos.	**Información obtenida:** - Contactos del dispositivo. - Mensajes SMS recibidos y enviados. - Historial de llamadas. - Archivos multimedia como fotos y videos.
DETALLE	Penetración_Prueba	Penetración_Keylogger

Tabla 6: Cuadro descriptivo del arte de malware en escenarios de pruebas

3.1.1.5 FASE V: REPORTE

En esta fase comprende en el desarrollo de reporte de las pruebas de escenarios realizada en la fase anterior con la finalidad de presentar un resume de los hallazgos obtenidos en las pruebas de seguridad, describiendo procedimientos, resultados y recomendaciones. Su objetivo es detallar vulnerabilidades detectadas y proponer medidas para fortalecer la seguridad del sistema.

- El Reporte 1: Prueba.apk evaluó la vulnerabilidad Reverse_TCP, demostrando cómo un atacante puede acceder al dispositivo Android y extraer datos sensibles. Se destacó la necesidad de controles estrictos de instalación y actualizaciones de seguridad para mitigar estos riesgos (Ver Reporte1: Prueba,apkj).
- El Reporte 2: Lokiboard.apk analizó un keylogger que registraba pulsaciones de teclas sin detección, comprometiendo contraseñas y datos personales. Se recomendó reforzar las medidas de detección de malware y monitoreo de configuraciones críticas del sistema (Ver Reporte2: Lokiboard.apk).

3.1.2 PROPUESTA GUÍA

GUÍA TÉCNICA DE PREVENCIÓN CONTRA MALWARE RAT EN DISPOSITIVOS ANDROID

El malware RAT (Trojan de Acceso A distancia) en equipos Android es una amenaza sofisticada que facilita a los intrusos tener control total a distancia sobre los dispositivos impactados. Esto abarca el acceso a información delicada, la activación de cámaras y micrófonos, la supervisión en tiempo real y la implementación de órdenes malintencionadas que ponen en riesgo la privacidad y protección del usuario. Adicionalmente, este tipo de malware puede emplear métodos de persistencia y evasión, lo que complica su identificación y suprimir. Esta guía ofrece una perspectiva técnica minuciosa, incluyendo acciones preventivas, supervisión especializada y tácticas de mitigación fundamentadas en herramientas sofisticadas y mejores prácticas para asegurar la protección de los dispositivos Android.

SEGURIDAD PROACTIVA EN DISPOSITIVOS ANDROID

1. **Actualización del sistema operativo y aplicaciones:**
    - **Descripción:** Mantener actualizado Android es esencial para corregir vulnerabilidades.
    - **Técnica:** Activar las actualizaciones automáticas y revisar periódicamente la instalación de parches de seguridad.
    - **Herramienta recomendada: OEM Update Checkers** de fabricantes como Samsung o Xiaomi para confirmar versiones seguras.
2. **Gestión granular de permisos:**
    - **Descripción:** Restringir permisos no esenciales otorgados a las aplicaciones.
    - **Técnica:** Desde **Ajustes > Aplicaciones**, revisar los accesos otorgados (como ubicación o almacenamiento).
    - **Herramienta recomendada: Bouncer**, para supervisar permisos temporalmente.

3. **Protección antimalware:**
    - **Descripción:** Prevenir infecciones con herramientas de seguridad que detecten RATs.
    - **Técnica:** Realizar escaneos programados y en tiempo real.
    - **Herramientas recomendadas:**
        - **Malwarebytes Mobile Security**: Análisis continuo y eliminación de malware.
        - **Avira Antivirus Security**: Bloqueo de amenazas de red y de dispositivos.
4. **Desactivación de depuración USB:**
    - **Descripción:** Evitar que atacantes exploten el puerto ADB activado en dispositivos Android.
    - **Técnica:** Desde **Opciones de desarrollador**, desactivar la depuración USB tras su uso.
    - **Consejo técnico:** Usar comandos como adb kill-server en entornos corporativos.
5. **Cifrado de datos:**
    - **Descripción:** Proteger información almacenada en el dispositivo con cifrado avanzado.
    - **Técnica:** Activar cifrado desde **Ajustes > Seguridad > Cifrar Teléfono**.
    - **Compatibilidad:** Funcionalidad predeterminada en dispositivos con Android 10 o superior.

MONITOREO Y DETECCIÓN DE ACTIVIDAD SOSPECHOSA

1. **Análisis del tráfico saliente:**
    - **Descripción**: Detectar conexiones no autorizadas hacia servidores controlados por atacantes.
    - **Técnica:** Usar herramientas como NetGuard para monitorear el tráfico de aplicaciones.
    - **Práctica recomendada**: Configurar reglas específicas para bloquear solicitudes a dominios sospechosos.

2. **Monitoreo de procesos en ejecución:**
    - **Descripción**: Identificar procesos ocultos o con comportamiento anómalo.
    - **Técnica:** Utilizar Task Manager Apps o comandos de terminal (ps aux) para listar procesos activos.
    - **Herramienta recomendada:** Activity Monitor para analizar patrones de CPU y RAM en tiempo real.
3. **Registro y análisis de eventos del sistema:**
    - **Descripción:** Registrar eventos críticos para identificar actividades maliciosas.
    - **Técnica:** Usar Logcat desde ADB para obtener detalles sobre errores o accesos anómalos.
    - **Comando ejemplo:** adb logcat > registro_eventos.txt para guardar los logs.
4. **Revisión de aplicaciones instaladas:**
    - **Descripción:** Identificar APKs maliciosas instaladas en el dispositivo.
    - **Técnica:** Ejecutar adb shell pm list packages y contrastar con bases de datos de malware como VirusTotal.
    - **Herramienta recomendada:** APK Analyzer para descompilar y examinar permisos de aplicaciones sospechosas.
5. **Alertas en tiempo real:**
    - **Descripción:** Configurar herramientas que envíen notificaciones sobre comportamientos extraños.
    - **Técnica:** Implementar apps con detección heurística, como Sophos Intercept X.

CONFIGURACIÓN SEGURA DE REDES Y ENTORNOS EN ANDROID

1. **Redes privadas y VPN:**
    - **Descripción:** Cifrar conexiones para evitar intercepciones de datos en redes públicas.
    - **Técnica:** Configurar VPN confiables como NordVPN o ProtonVPN.
    - **Consejo adicional:** Desactivar el uso compartido de red en dispositivos Android.

2. **Aislamiento de dispositivos vulnerables:**
    - **Descripción:** Separar dispositivos con configuraciones sospechosas en subredes dedicadas.
    - **Técnica:** Configurar VLANs en redes corporativas.
    - **Herramienta recomendada:** RouterOS para gestionar segmentación.
3. **Cifrado de red:**
    - **Descripción:** Proteger el tráfico de red con estándares seguros.
    - **Técnica:** Configurar routers domésticos con WPA3 y restringir accesos por dirección MAC.
4. **Control de puertos y conexiones:**
    - **Descripción:** Bloquear puertos abiertos que podrían ser explotados por RATs.
    - **Técnica:** Usar firewalls como AFWall+ para administrar conexiones en Android.
    - **Puerto crítico:** Bloquear el 5555, utilizado por ADB en modo remoto.
5. **Auditorías de conectividad:**
    - **Descripción:** Realizar revisiones periódicas de las configuraciones de red y sus logs.
    - **Técnica:** Usar scripts de análisis como Nmap para identificar puntos débiles en redes asociadas al dispositivo.

RESPUESTA Y MITIGACIÓN EN CASO DE INFECCIÓN

1. **Desconexión inmediata del dispositivo:**
    - **Descripción:** Aislar el dispositivo infectado para limitar el alcance del ataque.
    - **Técnica:** Apagar conexiones de red y datos móviles desde ajustes rápidos.
2. **Escaneo y eliminación del malware:**
    - **Descripción:** Analizar el sistema con herramientas avanzadas de eliminación.
    - **Herramienta recomendada:** Avast Mobile Security con análisis en tiempo real.

3. **Restauración a valores de fábrica:**
    - **Descripción**: Reiniciar el dispositivo desde Ajustes > Sistema > Restablecer.
    - **Consejo técnico**: Realizar un análisis previo de respaldos para evitar reinfecciones.
4. **Análisis forense del APK malicioso:**
    - **Descripción:** Examinar el archivo malicioso para comprender su comportamiento.
    - **Técnica:** Usar herramientas como MobSF para análisis estáticos y dinámicos.
5. **Revisión de configuraciones persistentes:**
    - **Descripción:** Identificar scripts de inicio automático o configuraciones maliciosas.
    - **Técnica:** Examinar archivos en /data/data/<paquete> y eliminar datos sospechosos.

POLÍTICAS Y BUENAS PRÁCTICAS PARA ANDROID

1. **Gestión centralizada con MDM:**
    - **Descripción:** Implementar soluciones como Microsoft Intune para aplicar políticas de seguridad en dispositivos empresariales.
2. **Política de acceso limitado:**
    - **Descripción:** Restringir permisos según roles específicos de los usuarios.
3. **Auditorías de seguridad:**
    - **Descripción:** Realizar pruebas de penetración regulares para evaluar vulnerabilidades.
4. **Capacitación continua:**
    - **Descripción:** Enseñar a los usuarios sobre riesgos comunes como phishing y APKs fraudulentas.
5. **Copias de seguridad regulares:**
    - **Descripción:** Configurar respaldos automáticos en entornos seguros como Google Drive.

CONCLUSIONES

- Se llevó a cabo un estudio comparativo de métodos estáticos y dinámicos del Malware RAT en dispositivos Android, brindando una comprensión completa del comportamiento del Malware. Las comparaciones de ambas técnicas se llevaron a cabo en base a patrones de análisis ideales, mientras que para las técnicas estáticas se dispone de la identificación de código, firmas características, desensamblado, entre otros aspectos. En cambio, las técnicas dinámicas incluyen la observación del comportamiento en tiempo real, que abarca interacciones con el sistema, comunicaciones de los servicios, actividades, entre otros aspectos. Por lo tanto, el estudio de estas técnicas destacó la relevancia de emplear este tipo de investigación para ofrecer enfoques totalmente precisos en detección y evaluación.

- Los laboratorios de pruebas basados en máquinas virtuales evidenciaron ser un ambiente seguro y regulado para llevar a cabo análisis exhaustivos del malware RAT. Estos incluyen herramientas sofisticadas como MobSF para análisis estático, que posibilita desensamblar el código, detectar vulnerabilidades y identificar firmas maliciosas, y Drozer para análisis dinámico, lo que permite el examen del comportamiento del malware en tiempo real y su interacción con servicios susceptibles. Se crearon dos escenarios de pruebas de penetración: en el primero, un paquete malintencionado consiguió extraer datos delicados, como contactos, mensajes de texto y archivos multimedia; mientras que, en el segundo, se concentró en recuperar pulsaciones de texto, consiguiendo así credenciales, números de teléfono y mensajes privados.

- La evolución del malware RAT para Android, que emplea el payload malicioso generado con msfvenom y el keylogger Lokiboard Keyboard, demuestra una habilidad sofisticada para poner en riesgo la seguridad de los dispositivos. El payload posibilita el acceso total a distancia, simplificando la obtención de datos personales, imágenes, contactos y mensajes. Por otro lado, la tecla Lokiboard registra las

pulsaciones de teclado y las guarda en un archivo, facilitando así la recuperación de credenciales, números de teléfono y correos. Los dos elementos colaboran para garantizar la permanencia en el dispositivo y eludir la detección, lo que resalta la importancia de establecer estrategias de seguridad sólidas para salvaguardar la información delicada en Android.

- La guía elaborada incluye tácticas técnicas y operativas que incluyen acciones de prevención, reacción y seguimiento para reducir los riesgos vinculados al malware RAT. Incorpora sugerencias concretas como la renovación constante de sistemas, la utilización de instrumentos de análisis sofisticados, la configuración de redes seguras y la formación de los usuarios. Estas acciones refuerzan la posición de seguridad en aparatos Android, disminuyendo la posibilidad de infección y atenuando el efecto si se produce una intrusión.

RECOMENDACIONES

- Es fundamental fortalecer el análisis de malware utilizando herramientas avanzadas como MobSF para el análisis estático, permitiendo desensamblar el código y detectar vulnerabilidades, y Drozer para el análisis dinámico, que evalúa el comportamiento del malware en tiempo real. Además, se recomienda capacitar a los especialistas en técnicas avanzadas de detección para mejorar la precisión en la identificación de amenazas y patrones maliciosos.

- La configuración de entornos seguros para pruebas es esencial para realizar análisis controlados del malware. Esto incluye el uso de máquinas virtuales como VirtualBox y Android Studio Emulator, configuraciones de redes aisladas para evitar la fuga de datos y el diseño de escenarios específicos que permitan evaluar la capacidad del malware para comprometer dispositivos y extraer datos sensibles.

- Para proteger los dispositivos Android frente a RATs, se recomienda mantener actualizados tanto el sistema operativo como las aplicaciones, cerrar vulnerabilidades conocidas mediante parches de seguridad y restringir los permisos de aplicaciones a lo estrictamente necesario. También es crucial utilizar software de seguridad confiable que permita detectar y eliminar amenazas como payloads maliciosos y keyloggers.
- La implementación de estrategias de prevención y mitigación incluye monitorear de manera proactiva el comportamiento de los dispositivos, identificando patrones sospechosos en el tráfico de datos y la interacción con el sistema. Además, la educación de los usuarios sobre buenas prácticas, como instalar aplicaciones solo desde fuentes confiables, y la realización de copias de seguridad periódicas, son medidas clave para minimizar los riesgos.

- Es imprescindible establecer políticas de seguridad tecnológica robustas. Estas deben incluir planes de respuesta rápida ante incidentes de malware, cifrado de datos sensibles para proteger la información extraída y la realización de auditorías periódicas que permitan identificar y mitigar posibles vulnerabilidades. Estas acciones refuerzan la seguridad de los dispositivos Android frente a amenazas avanzadas como el malware RAT.

BIBLIOGRAFÍAS

[1] E. &. B. L. G. Velasteguí López, «El avance en la tecnología móvil y su impacto en la sociedad,» *Explorador Digital,* vol. 2, nº 4, pp. 5-19, 2019.

[2] Victor Ruiz, «Linkedin - El Análisis de Malware en la Era de la Ciberseguridad,» 10 Octubre 2023. [En línea]. Available: https://es.linkedin.com/pulse/el-an%C3%A1lisis-de-malware-en-la-era-ciberseguridad-victor-ruiz-rjmac. [Último acceso: 08 Abril 2024].

[3] Karpersky, «Los ataques a dispositivos móviles aumentaron más del 50% en 2023,» Karpersky, 26 Febrero 2024. [En línea]. Available: https://latam.kaspersky.com/about/press-releases/2024_los-ataques-a-dispositivos-moviles-aumentaron-mas-del-50-en-2023. [Último acceso: 08 Abril 2024].

[4] Anton Kivva, «Evolución de las amenazas informáticas en el primer trimestre de 2023. Estadísticas de amenazas móviles,» 07 Junio 2023. [En línea]. Available: https://securelist.lat/it-threat-evolution-q1-2023-mobile-statistics/97916/. [Último acceso: 25 Octubre 2024].

[5] Jhon Pinzón R. , «Análisis del impacto de los ataques de ransomware en las organizaciones colombianas para la determinación de nuevas estrategias de protección cibernética,» Universidad Nacional Abierta y a Distancia - UNAD, Bógota - Colombia, 2021.

[6] Stalyn Andrango P. , «Análisis comparativo de Malware en telefonos inteligentes Android, prevencíón y mitigacíon de sus efectos,» Universidad Central del Ecuador , Quito - Ecuador , 2023.

[7] Linda Briones L. , «Laboratorio virtual de analisis y comportamiento de malware basados en técnicas y métodos de seguridad informáticas para los laboratorios en la facultad de sistemas y telecomunicaciones,» Universidad Estatal Peninsula de Santa Elena, La Libertad - Ecuador , 2020.

[8] Tatiana Jumbo T., «Metodología para el Analisis de Malware en un ambiente controlado,» Universidad Politécnica Salesiana Sede Cuenca, Cuenca - Ecuador, 2017.

[9] Ivan Belcic, «¿Qué es el malware y cómo protegerse de los ataques?,» Avast - Malware, 19 Enero 2023. [En línea]. Available: https://www.avast.com/es-es/c-malware. [Último acceso: 08 Abril 2024].

[10] S. Campbell, DISEÑOS EXPERIMENTALES Y CUASIEXPERIMENTALES EN LA INVESTIGACION SOCIAL, AMMORRORTU, Ed., BUENOS AIRES, 2002.

[11] Robert Hernandez S. ; Carlos Fernandez C. ; Pilar Bapista L. , «Metodología de la investigación - Explotatorios,» McGRAQ-Hill/interamericana editores S.A de C.V, Mexico, 2010.

[12] Fernández Pita, «INVESTIGACIÓN CUANTITATIVA Y CUALITATIVA,» ESPAÑA, CAD ATEN PRIMARIA, 2002, pp. 9:76-78.

[13] Sampieri, Metodología de la Investigación, México: McGRAW-HILL, 2010.

[14] T.P. Team, «Ptes- Standard,» 2017. [En línea]. Available: https://pentest-standard.readthedocs.io/en/latest/. [Último acceso: 09 Abril 2024].

[15] Alberto R. , William Y. , Johana M. , «Auditoría Informática,» La Caracola Editores, Riobamba - Ecuador, 2022.

[16] CIS Informática , «seguro, El panorama en constante evolución de la ciberseguridad: Desvelando los secretos de un mundo digital,» 09 Abril 2024. [En línea]. Available: https://www.cisinformatica.cat/es/evolucion-de-la-ciberseguridad/. [Último acceso: 24 Octubre 2024].

[17] Javier Candau, «Ciberseguridad: Evolución y tendencias,» Ieee.es, 2021.

[18] Nica Latto, «¿Qué es WannaCry?,» 27 Febrero 2020. [En línea]. Available: https://www.avast.com/es-es/c-wannacry. [Último acceso: 24 Octubre 2024].

[19] Tatiana Jumbo T., «Metología para el análisis de Malware en un ambiente controlado,» Universidad Politécnica Salesiana - Sede Cuenca, Cuenca - Ecuador, 2017.

[20] Antoni Sánchez M., «MISTIC TFM: Seguridad en Android - Análisis de vulnerabilidades y malware,» Instituto Nacional de Ciberseguridad - Incibe_, 2017.

[21] Gustavo Yánez A. , Christian Germán S. Paulina Ramírez P. , Andrea Balseca R. , Danny Sayay R. & Wernher Gómez T., «Estudio de ataques RAT en dispositivos móviles Android a funcionarios de instituciones,» UIDE - Arizona Satate University, Quito - Ecuador, 2022.

[22] A. W. Service, «¿Qué es la ciberseguridad?,» [En línea]. Available: https://aws.amazon.com/es/what-is/cybersecurity/. [Último acceso: 09 Octubre 2024].

[23] Gregg Lindemulder ; Matt Kosinki, «¿Qué es la ciberseguridad?,» 12 Agosto 2024. [En línea]. Available: https://www.ibm.com/es-es/topics/cybersecurity. [Último acceso: 09 Octubre 2024].

[24] Kaspersky, «¿Qué es el cibercrimen?,» 2024. [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/what-is-cybercrime. [Último acceso: 09 Octubre 2024].

[25] Nazly Borrero V., «¿Qué es la Cibercriminalidad? Tipos de Cibercriminales,» 30 Enero 2024. [En línea]. Available: https://es.linkedin.com/pulse/qu%C3%A9-es-la-cibercriminalidad-tipos-de-cibercriminales-borrero-v%C3%A1squez-y4ykc. [Último acceso: 09 Octubre 2024].

[26] UNIR - La Universidad en Internet, «¿Qué es el hacking ético y cuál es su importancia en la actualidad?,» 22 Noviembre 2022. [En línea]. Available: https://ecuador.unir.net/actualidad-unir/hacking-etico/. [Último acceso: 09 Octubre 2024].

[27] IBM, «¿Qué es el pirateo cibernético?,» [En línea]. Available: https://www.ibm.com/mx-es/topics/cyber-hacking#:~:text=Los%20hackers%20maliciosos%20(a%20veces,modalidades%20de%20ransomware%20como%20servicio).. [Último acceso: 09 Octubre 2024].

[28] UNIR - La universidad en Internet, «¿Qué es el hacktivismo y qué delitos implica?,» 13 Diciembre 2021. [En línea]. Available: https://www.unir.net/revista/derecho/hacktivismo/. [Último acceso: 09 Octubre 2024].

[29] Luiza Pires, «Hacker vs Cracker: Entiende sus diferencias,» 14 Marzo 2024. [En línea]. Available: https://www.welivesecurity.com/es/otros-temas/hacker-vs-cracker-entiende-sus-diferencias/. [Último acceso: 09 Octubre 2024].

[30] Panda a WatchGuard Brand, «Phishing,» [En línea]. Available: https://www.pandasecurity.com/es/security-info/phishing/. [Último acceso: 09 Octubre 2024].

[31] ESET Progress Protected, «¿Cuál es la definición de "spam"?,» [En línea]. Available: https://www.eset.com/es/caracteristicas/spam/. [Último acceso: 09 Octubre 2024].

[32] Luiza Pires, «¿Qué tipos de hacker existen y qué los diferencia?,» 15 Noviembre 2023. [En línea]. Available: https://www.welivesecurity.com/es/otros-temas/tipos-hacker-diferencias/. [Último acceso: 09 Octubre 2024].

[33] UNIR - La universidad en Internet, «Fraudes por internet: ¿Qué tipos de estafas son más habituales?,» 08 Abril 2024. [En línea]. Available: https://www.unir.net/revista/derecho/fraudes-internet/. [Último acceso: 09 Octubre 2024].

[34] KASPERSKY, «¿Qué es una vulneración de seguridad?,» [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/what-is-a-security-breach. [Último acceso: 25 Octubre 2024].

[35] IBM, «¿Qué es la gestión de vulnerabilidades?,» [En línea]. Available: https://www.ibm.com/es-es/topics/vulnerability-management. [Último acceso: 25 Octubre 2024].

[36] GoDaddy, «¿Qué es una amenaza informática?,» 29 Julio 2024. [En línea]. Available: https://www.godaddy.com/resources/es/seguridad/7-tipos-de-amenazas-informaticas-que-toda-pyme-debe-saber. [Último acceso: 25 Octubre 2024].

[37] José Cruz B., «¿Cómo definir las amenazas de seguridad en tus activos informáticos?,» 12 Julio 2024. [En línea]. Available: https://es.linkedin.com/pulse/c%C3%B3mo-definir-las-

amenazas-de-seguridad-en-tus-activos-cruz-bringas-sxvoc. [Último acceso: 12 Octubre 2024].

[38] Eric Ruiz C., «Riesgo Tecnológico,» 06 Julio 2024. [En línea]. Available: https://es.linkedin.com/pulse/riesgo-de-ciberseguridad-eric-ruiz-ch%C3%A1vez-9qbke#:~:text=El%20riesgo%20de%20ciberseguridad%20se,a%20una%20organizaci%C3%B3n%20o%20individuo.. [Último acceso: 25 Octubre 2024].

[39] Ztech , «Gestión de Riesgos Tecnológicos en Empresas Latinoamericanas,» 13 Marzo 2024. [En línea]. Available: https://es.linkedin.com/pulse/gesti%C3%B3n-de-riesgos-tecnol%C3%B3gicos-en-empresas-qg0qf. [Último acceso: 25 Octubre 2024].

[40] Alejandro Gonzalez, «¿Qué es Android?,» 09 Febrero 2011. [En línea]. Available: https://www.xatakandroid.com/sistema-operativo/que-es-android. [Último acceso: 25 Octubre 2024].

[41] IBM, «¿Qué es un ciberataque?,» [En línea]. Available: https://www.ibm.com/es-es/topics/cyber-attack#:~:text=Un%20ciberataque%20es%20cualquier%20esfuerzo,sistema%20inform%C3%A1tico%20o%20dispositivo%20digital.. [Último acceso: 25 Octubre 2024].

[42] KASPERSKY, «Cómo prevenir ciberataques,» [En línea]. Available: https://latam.kaspersky.com/resource-center/preemptive-safety/how-to-prevent-cyberattacks. [Último acceso: 25 Octubre 2024].

[43] Kaspersky, «¿Cuáles son los diferentes tipos de malware?,» Kaspersky, 2024. [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/types-of-malware. [Último acceso: 09 Abril 2024].

[44] Joseph Regan & Ivan Belcic, «¿Qué puede hacer el malware y qué peligro tiene?,» AVG, 14 Febrero 2022. [En línea]. Available: https://www.avg.com/es/signal/what-is-malware. [Último acceso: 09 Abril 2024].

[45] MalwareBytes, «¿Cómo funciona el phishing?,» [En línea]. Available: https://www.malwarebytes.com/es/phishing. [Último acceso: 2024 Octubre 2024].

[46] Patrick Seguin, «Spyware: detección, prevención y eliminación,» 20 Febrero 2020. [En línea]. Available: https://www.avast.com/es-es/c-spyware. [Último acceso: 12 Octubre 2024].

[47] Redacción Banco Pichincha, «Adware: aprende a protegerte de la publicidad maliciosa,» 05 Julio 2022. [En línea]. Available: https://www.pichincha.com/blog/que-es-el-adware. [Último acceso: 12 Octubre 2024].

[48] IBM, «¿Qué es el ransomware?,» [En línea]. Available: https://www.ibm.com/es-es/topics/ransomware. [Último acceso: 12 Octubre 2024].

[49] Kaspersky, «Definición de troyano,» [En línea]. Available: https://www.kaspersky.es/resource-center/threats/trojans. [Último acceso: 12 Octubre 2024].

[50] I. Belcic, «¿Cómo funcionan los gusanos informáticos?,» 21 Enero 2016. [En línea]. Available: https://www.avast.com/es-es/c-computer-worm. [Último acceso: 12 Octubre 2024].

[51] Nica Latto, «¿Cómo funcionan los virus informáticos?,» 12 Febrero 2020. [En línea]. Available: https://www.avast.com/es-es/c-computer-virus. [Último acceso: 12 Octubre 2024].

[52] Nica Latto, «¿Qué es un keylogger?,» 11 Agosto 2016. [En línea]. Available: https://www.avast.com/es-es/c-keylogger. [Último acceso: 13 Octubre 2024].

[53] Kaspersky, «¿Qué es un botnet?,» [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/botnet-attacks. [Último acceso: 13 Octubre 2024].

[54] AVG, «Qué es un PUP... y cómo se elimina,» 14 Agosto 2021. [En línea]. Available: https://www.avg.com/es/signal/what-is-a-pup. [Último acceso: 13 Octubre 2024].

[55] Kaspersky, «Protección de amenazas sin archivos,» [En línea]. Available: https://www.kaspersky.es/enterprise-security/wiki-section/products/fileless-threats-protection. [Último acceso: 13 Octubre 2024].

[56] LinkedIn, «LinkedIn - ¿Cómo se pueden comparar las técnicas de detección de malware estático y dinámico,» Lik, 13 Octubre 2023. [En línea]. Available: https://es.linkedin.com/advice/1/how-can-you-compare-static-dynamic-malware?lang=es. [Último acceso: 09 Abril 2024].

[57] Ilija Shatilin, «Parte 4: El malware móvil y dónde podemos encontrar estos ciberataques,» Kaspersky Daily, 23 Octubre 2018. [En línea]. Available: https://www.kaspersky.es/blog/mobile-malware-part-4/17232/. [Último acceso: 09 Abril 2024].

[58] Github, «Mobile Security Framework (MobSF),» Github, [En línea]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF. [Último acceso: 09 Abril 2024].

[59] Dvelopers Android, «Android Studio,» Developers, [En línea]. Available: https://developer.android.com/studio. [Último acceso: 10 Abril 2024].

[60] BlueStacks, «BlueStacks,» [En línea]. Available: https://www.bluestacks.com/es/index.html. [Último acceso: 09 Abril 2024].

[61] LinkedIn, «Cuckoo Sandbox,» 21 Noviembre 2023. [En línea]. Available: https://es.linkedin.com/advice/0/how-can-you-use-cuckoo-sandbox-analyze-malware-2ewvf?lang=es. [Último acceso: 09 Abril 2024].

[62] Github, «Drozer,» Github, [En línea]. Available: https://github.com/WithSecureLabs/drozer. [Último acceso: 09 Abril 2024].

[63] WeliveSecurity, «Remux V6: Explorando la ultima version de la suite para ejecutable malicioso,» WeliveSecurity, 18 Junio 2015. [En línea]. Available: https://www.welivesecurity.com/la-es/2015/06/18/remnux-v6-explorando-suite-analisis-malware/. [Último acceso: 09 Abril 2024].

[64] Coip_art, 2021.

[65] Ministerio de Telecomunicaciones , «LEY DEL SISTEMA NACIONAL DE REGISTRO DE DATOS PÚBLICOS,» Gobierno Nacional del Ecuador , 2012.

[66] Ministerio de Telecomunicaciones, «LEY DEL SISTEMA NACIONAL DE REGISTRO DE DATOS PÚBLICOS,» 2012.

[67] Corporación Nacional de Finanzas Populares y Solidarias - CONAFIPS, «Ley orgánica de protección de datos personales,» Gorbierno Nacional del Ecuador, 2021.

ANEXOS

ANEXOS #1: Entrevista al experto en Hacking Ético de la Universidad Península de Santa Elena, Especialista 1

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA **FACULTAD DE SISTEMAS Y TELECOMUNICACIONES** **TECNOLOGÍAS DE LA INFORMACIÓN**
Objetivo: Este cuestionario está diseñado para la recopilar información sobre los problemas de la seguridad informática en dispositivos Android, centrándose en las vulnerabilidades y métodos de detección de malware RAT hacia expertos de Ciberseguridad.
¿Cuál es su experiencia en el análisis de malware, especialmente en el contexto de RAT en plataformas Android?
Desde su experiencia se ha realizado en la universidad algunas pruebas en laboratorios controlados, ha dirigido un par de tesis de análisis de malware el cual pues en su momento analizo lo que es el ransoware en este caso hizo análisis automático con técnicas manuales y temas controlados en ambiente de laboratorios, el tema del Rat piensa que es un tema muy importante, es un malware que en algún punto toma control remoto del dispositivo que está infectando, entorno ah esto en los trabajos anteriores ah usado las técnicas estáticas como dinámicas.
En su opinión, ¿qué beneficios específicos ofrece la combinación de técnicas de detección estáticas y dinámicas en el análisis de malware RAT en comparación con el uso de una técnica de forma aislada?
Si en algún punto se utiliza estas por separadas las técnicas tanto estáticas como dinámicas puede ser que se vayan ciertos temas importantes o que se escape un comportamiento que tiene el malware, el uso o la combinación de estas técnicas el beneficio es que maximiza la capacidad d detección. x ejemplo: En el análisis estático: hay malware que en algún punto utiliza técnicas de ofuscación que puede ser q no se detecten. En el análisis dinámico: se va a ver ciertas conexiones o ciertos comportamientos que reflejan q si es un malware. La combinación lo que hace es maximizar la capacidad de la detección.
¿Ha observado mejoras en la precisión y eficacia de la detección de RAT al aplicar técnicas de detección combinadas en su experiencia? Si es así, ¿puede proporcionar ejemplos específicos?
En algún momento al usar técnicas combinadas se detectó un malware, que en la técnica estática había utilizado una ofuscación la carga útil el payload entonces había utilizado esta ofuscación y el análisis estático no se lo detecto paso como un código o archivo original, legitimo pero al ejecutar el análisis dinámico se observó como el malware intentaba conectarse a un servidor de comando y control, un servidor de comando y control es un servidor que está esperando que el malware sea ejecutado en algún dispositivo en alguna parte del mundo y este malware se conecta a este servidor descarga ciertas cosas adicionales y le da el control total a una persona remotamente del dispositivo entonces en el estático paso x alto mientras q en el dinámico se percató que quería conectarse a un servidor de comando y control eso es lo que sucedió en la precisión y eficacia de mejora en el uso de las técnicas combinadas.

¿Cuáles son las técnicas de detección estática que considera más efectivas para identificar RAT en plataformas Android, y por qué?
En la parte estática lo que se va a analizar de cierta forma es el código, se desarma ese malware, el programa, el código y se comienza a analizar lo que básicamente se busca ahí son permisos que solicita el malware se busca patrones de códigos y firmas, por lo general al hablar de permisos, este tipo de malware te pide permisos excesivos en un dispositivo te pide permisos para tener acceso a los sms, permiso a cámara. permiso a micrófono, permiso a los contacto, si un software en este caso un apk destinado para cierta operación o función en el teléfono pero se visualiza que está pidiendo demasiados permisos a otras cosas que no tiene porque pedirlas entonces se puede tratar de algo en el análisis estático. Se puede utilizar también el apk tools es una aplicación donde permite revisar código fuentes y buscar a la final patrones conocidos de malware. Las técnicas entre estática y dinámicas, escoge las dos porque la estática a veces no detecta, en esta igual iría a revisar patrones, firmas, códigos y unas de las herramientas seria el apk tools.
Desde su perspectiva, ¿qué limitaciones tienen las técnicas de detección estática cuando se utilizan solas en el análisis de RAT?
La limitación seria es que no siempre se podrían identificar códigos q vienen ofuscados o códigos polimórficos ósea códigos que vienen de alguna forma lleno de parámetros y basura entre comillas, que hacen que sea imposible la detección sencilla y que un antivirus a la final lo pueda detectar la limitación es esa no siempre se podría identificar malwares que vienen ofuscados o en un lenguaje polimórfico, no vas a poder capturar comportamientos así es el estático, no capturas que te crea un archivo no capturas q se conecte a una red pública entonces básicamente el análisis estático x si solo puede fallar, muy limitado muy simple, es que ya existen técnicas d evasión de ese tipo de análisis o los famosos encoders que existen
¿Qué técnicas de detección dinámica ha encontrado más útiles para la identificación de RAT, y cómo complementan las técnicas de detección estática?
En esta pregunta se configura el entorno donde se va a realizar el análisis, en las técnicas dinámicas dijo que son muchos más útiles pero se hacen en entornos controlados se lo hacen en sambox, en donde tienes q instalar casi un entorno real un android, herramienta d análisis kali, o cain q sirve para ver temas d forense, tal vez un servidor que simule internet o de salida a internet entonces ese sambox o ese laboratorio es el q va a permitir que de alguna forma ejecutar una técnica dinámica entonces esto complementa mucho al análisis estático porque en si ya da una vision más real d interacción entre el dispositivo y la red porque en algún punto cuando se ejecuta el malware va a buscar la re, va a buscar conexiones hacia afuera, va a crear tal vez archivos o va a descargar archivos y es ahí donde se va a complementar mucho lo que se realizó estáticamente o tal vez si estáticamente no se detectó nada se va a poder visualizar que si es un malware porque está comportándose como tal entonces las técnicas que se podrían utilizar es sambox que son estos laboratorios que hay q implementar en algún momento para complementar con el estático.
¿Podría describir algún caso en el que la combinación de técnicas de detección haya resultado en la identificación de RAT que no se habría detectado utilizando solo una de las técnicas?

Claro si hay una caso de un rat que utilizaba ofuscación avanzada o encoder avanzado, encoder que no estaban detectadas las firmas y eran desconocidas y en ese caso se utilizaba esta ofuscación avanzada para esconder el payload para esconder el código malicioso y paso el análisis estático ósea se fue como un archivo legitimo cm un programa o un apk legitimo básicamente el análisis estático revelo que era un código inofensivo pero al ejecutar la muestra, ejecutar el código en el entorno controlado en el sambox se pudo ver efectivamente quería conectarse a servidores afuera y comenzó a descargar archivos entonces es un caso que sucedió que en algún punto te lleva a ver la importancia de ejecutar los dos análisis.
En términos de implementación, ¿qué desafíos ha enfrentado al combinar técnicas de detección estáticas y dinámicas, y cómo los ha superado?
El mayor desafío esta tal vez en el recurso del hardware y se puede presentar en algún momento necesitaras alta demanda de recursos porque vas no solamente a implementar un servidor si no q se va a implementar tal vez 3 o 2 servidores para hacer el análisis dinámico entonces te va a demanda recursos en cuanto hardware y de paso también tiempo en la implementación del laboratorio en la implementación del sambox. En el análisis dinámico por otro lado puede ser lento y difícil de configurar, especialmente porque se tiene que simular entornos reales, seria instalar una máquina virtual con un android y una máquina que simule un cliente u otros teléfonos para ver si el comportamiento no es que se quiere pasar d uno a otro, tal vez una máquina que te esté censando que archivos quiere crear. El tema es el recurso el hardware y el tiempo como nosotros lo hemos superado es el uso de servidores, nosotros no lo hemos hechos en máquinas laptops o máquinas de escritorios normales si no que se le ha ejecutado en servidores con un entorno virtualizado se creó una vlan totalmente separadas de las vlans de producción se encerró el laboratorio y sobre eso se creó máquinas virtuales para cada una d esto, La otra opción como superar también al ejecutar en algún momento fue con una máquina de escritorio pero que tenía buenas características en hardware tenía una tarjeta gráfica de 8 o 16 de ram tenía la ram propia de la máquina que era 32 o 40 no recuerda bien pero q así ha sido superado el tema del hardware porque si se quiere ejecutar en una laptop de un usuario d una laptop sencilla a la final te va a llevar mucho en cuestión el tiempo y se presentaran más dificultades.
¿Qué métricas considera más relevantes para evaluar la precisión y eficacia en la detección de RAT al usar técnicas combinadas en comparación con técnicas individuales?
Porque en algún punto es como decir q una métrica se va a medir el tiempo de ejecución no sería una métrica tan real. La técnica combinada te va a llevar más tiempo que la técnica individual.

1 El tiempo de ejecución es una métrica

Usar una técnica por separado te va a llevar menos tiempo que al realizarla las dos técnicas.

2 La tasa de detección seria las segunda métrica porque si se usa una técnica x separado a la final no se podría detectar x ejemplo si analizas 10 muestras esa técnica separada te va a dar una métrica de 5 archivos detectados, pero en cambio si se utiliza la técnicas combinadas esas muestras que se lanzó en la anterior de esas 10 puede ser que detecte 8.

La tasa de detección viene de la mano con la métrica de falsos positivos, falsos positivos: es que una te diga tal vez que si es y la otra que te diga q no, es un código limpio.

3 La cobertura del comportamiento: tal vez una técnica no de todo el comportamiento real x ejemplo la técnica estática está pidiendo permisos de ciertas cosas.

La dinámica o la combinación de ambas va a dar una métrica diferente en cuanto a comportamiento está creando archivos está bajando esto.

¿Qué recomendaciones daría a los investigadores y profesionales de la ciberseguridad sobre el uso de técnicas de detección combinadas para mejorar la detección de malware en dispositivos Android?

Las recomendaciones:

1 Utilizar el hardware apropiado para q facilite el trabajo y d alguna forma disminuya el tiempo de análisis

2 No depender de una sola técnica porque si dependes d una sola se pueden pasar por alto algunas cosas cm ya se ha venido mencionando en las anteriores respuestas porque el análisis estático y dinámico ofrecen una mayor cobertura y reducen la posibilidad de que un malware pase por alto.

3 Invertir en herramientas aunque ahora existen diversidades de herramientas de softwares libres, además que existen sambox preinstalados o prediseñados para este tipo de cosas, si es que hay la posibilidad d invertir en estas herramientas pues que se realice.

4 Mantener actualizado porque d alguna manera el tema de ofuscación o de evasión o el tema de encoders siempre está evolucionando los atacantes siempre están tratando de mejorar estas técnicas de ofuscación de evasión estos encoders para poder pasarse los antivirus y poder pasarse este trabajo que realizan la gente de seguridad.

ANEXOS #2: Entrevista al experto en Seguridad Informática de la Universidad Península de Santa Elena, Especialista 2

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA **FACULTAD DE SISTEMAS Y TELECOMUNICACIONES** **TECNOLOGÍAS DE LA INFORMACIÓN**
Objetivo: Este cuestionario está diseñado para la recopilar información sobre los problemas de la seguridad informática en dispositivos Android, centrándose en las vulnerabilidades y métodos de detección de malware RAT hacia expertos de Ciberseguridad.
¿Cuál es su experiencia en el análisis de malware, especialmente en el contexto de RAT en plataformas Android?
En su experiencia especifica que se ha enfocado al análisis de tráfico. Eh incluso se encuentra trabajando en procesos de investigación en análisis de tráfico de red de manera general: tráfico de red, trafico web, tráfico de servicios, vpn, lo que me pide un celular, el tráfico que haga un celular de descarga en toda esa cuestión coge este tráfico lo limpia y luego se le ubica al malware que haga su trabajo para poder buscar ciertos warning, ciertas advertencias que me digan son malwares. Hablando del tema que le pregunte respondió que no ha tenido la experiencia como tal de haber trabajado en el ámbito de tráfico de dispositivo móviles, tal vez a futuro se darán ciertos rasgos que van a dar la opción de entrar en esa parte.
En su opinión, ¿qué beneficios específicos ofrece la combinación de técnicas de detección estáticas y dinámicas en el análisis de malware RAT en comparación con el uso de una técnica de forma aislada?.
En las estáticas comenta casi que no es partidario por lo general son líneas agregadas a herramientas predefinidas o establecidas sin opción de modificar o cambiar, si no lo que le vota el resultado de esta, eso es. Es más partidario de las técnicas dinámicas, en su defecto se pueda para poder modificar cierto código, modificar estructuras o mejorar los reportes que se tiene. En esto está trabajando con ciertos estudiantes haciendo danswors o armando estructuras a través de algoritmos que den buenos resultados para detección de tráfico, en la actualidad un poco orientado a peticiones web y esas peticiones pueden ser de una maquina o de un celular
¿Ha observado mejoras en la precisión y eficacia de la detección de RAT al aplicar técnicas de detección combinadas en su experiencia? Si es así, ¿puede proporcionar ejemplos específicos?
Indica que las estáticas realmente sólo aplican una estructura o una herramienta y vota un solo resultado en la forma de modificar va a cambiar en la dinámica si tiene así q se inclina hacia la detección dinámica, pero su orientación es mas a desarrollo a crearlas no

específicamente a comprarlas, porque no comprarlas, porque la mayor parte de estas herramientas vienen dentro d un paquete que se llama o soap, que son paquetes q orquestan toda la seguridad x capa pero su costo es elevado de 20 mil a 50 mil... que es lo más óptimo pues crear un grupo que desarrolle este tipo aplicaciones desde cero, que armen su propia arquitectura y su propia estructura de desarrollo de Ciberseguridad desde su punto de vista es su recomendación, tal vez se apoyen con Apis x ejemplo un proyecto de trabajo con un estudiante donde desarrollo una estructura para capturar paquetes pero el Apis de virus total o de apisboy que son plataformas que están en la nube son los que me devolvían si eran virus o no, cogía el paquete que era recolectado por el tráfico de red y eso le enviaba al apivoy y el apisvoy me devolvía este es virus, este no, pero ya la base de datos estaba ya en estas herramientas. Un estudiante ya desarrollo esta estructura.
¿Cuáles son las técnicas de detección estática que considera más efectivas para identificar RAT en plataformas Android, y por qué?
Ya viene con plataformas como x ejemplo chatboitng, fortigate ya viene con licencia pero no puedo modificar. Esta es una pregunta cerrada y muy técnica, no me podría ayudar con esta pregunta. El tema de nist pide mucho requisitos para implementarla, se podría decir es la esencia para tener Ciberseguridad. Otras entidades utilizan owasp como también se ha estudiado en clases.
Desde su perspectiva, ¿qué limitaciones tienen las técnicas de detección estática cuando se utilizan solas en el análisis de RAT?
Una desventaja clara es el que una persona de Ciberseguridad confía mucho en una aplicación cerrada es como si que si tengo un antivirus confié en el y se q no me va a pasar nada entre risas pero se sabe que esto no es real. Hace hincapié que se debe tener más d una alternativa. Se guía más en su línea de trabajar en tema de Ciberseguridad por cable y se habla de una analogía de trabajar como el de capas de cebolla.
¿Qué técnicas de detección dinámica ha encontrado más útiles para la identificación de RAT, y cómo complementan las técnicas de detección estática?
Las técnicas que desarrollan las mismas personas utilizando librerías libres de Python porque las que se desarrollan porque estas son más amigables y uno busca las capacidades de adiestrarlo a su gusto y paciencia como decir ah no me gusta eso y lo cambio. Se complementan al momento de combinarlas y no realizarlas de manera hibrida que es aquí donde interviene la seguridad por capas.
¿Podría describir algún caso en el que la combinación de técnicas de detección haya resultado en la identificación de RAT que no se habría detectado utilizando solo una de las técnicas?
El caso de un estudiante que realizo una herramienta de testeo y se conectó con un servicio en la nube lo usaba en la empresa y con eso empezaba a detectar malwares y virus de todo

tipo que no eran detectados por su antivirus, entonces el corría la aplicación, la aplicación solo funcionaba en el segmento de red todas las maquinas que estaban prendidas en ese momento comenzaba a testear, si una de las maquinas estaba mandando algún tipo de mensaje de tipo que tiene algún patrón de tipo malware o de tipo virus, el virus entonces enseguida le enviaba la alerta y el estudiante iba a la máquina para ver qué es lo que estaba pasando ósea ya le daba un alerta x eso sugiere las herramientas de desarrollo aunque no descarta las herramientas pagadas pero el tema es q son caras y eso dificulta poder probar o si no tener versiones demo de pocos días que da y eso no favorece al estudio.
En términos de implementación, ¿qué desafíos ha enfrentado al combinar técnicas de detección estáticas y dinámicas, y cómo los ha superado
El termino indica que es el sistema operativo aunque existen diversidad de estos, cuando se va a realizar la implementación a las empresas pues te topas desde Windows 10, hasta xp y Windows 7 pues en estos casos no sabe si va a funcionar con estos la herramienta o de pronto hay entidades que se va a implementar con Linux, Ubuntu, unos tienen debían otros fedora tienen diversidad d Linux. Sugiere tener una tercera maquina tener una simulación de un ids o d un firewall adicional a las dos maquinas Una tercera máquina que tenga pfsense y te haga algún tipo de detección.
¿Qué métricas considera más relevantes para evaluar la precisión y eficacia en la detección de RAT al usar técnicas combinadas en comparación con técnicas individuales?
Por ahora lo que se ha hecho es poder valorarlo con algoritmo de la inteligencia artificial y bueno hay las métricas que varían de acuerdo al algoritmo escort avengers. Precisión hay varias métricas que se pueden evaluar de acuerdo al algoritmo, dentro de las herramientas físicas se podría decir evaluar un checkpoint vs un fortinet, cuales son las bondades del uno y el otro.
¿Qué recomendaciones daría a los investigadores y profesionales de la Ciberseguridad sobre el uso de técnicas de detección combinadas para mejorar la detección de malware en dispositivos Android?
Su recomendación es : Trabajen en estructuras de seguridad por capas. Uso de las metodologías hibridas tanto las estáticas como dinámicas. No confiar de las aplicaciones con licencia si no tal vez tener más aplicaciones que generen niveles de seguridad. Crear cultura de Ciberseguridad.

ANEXO #3
FASE I: CONFIGURACIÓN DEL LABORATORIO VIRTUAL

INSTALACIÓN DE HERRAMIENTA MOBSF

1. Visitar el repositorio oficial de mobsf en github para copiar el repositorio

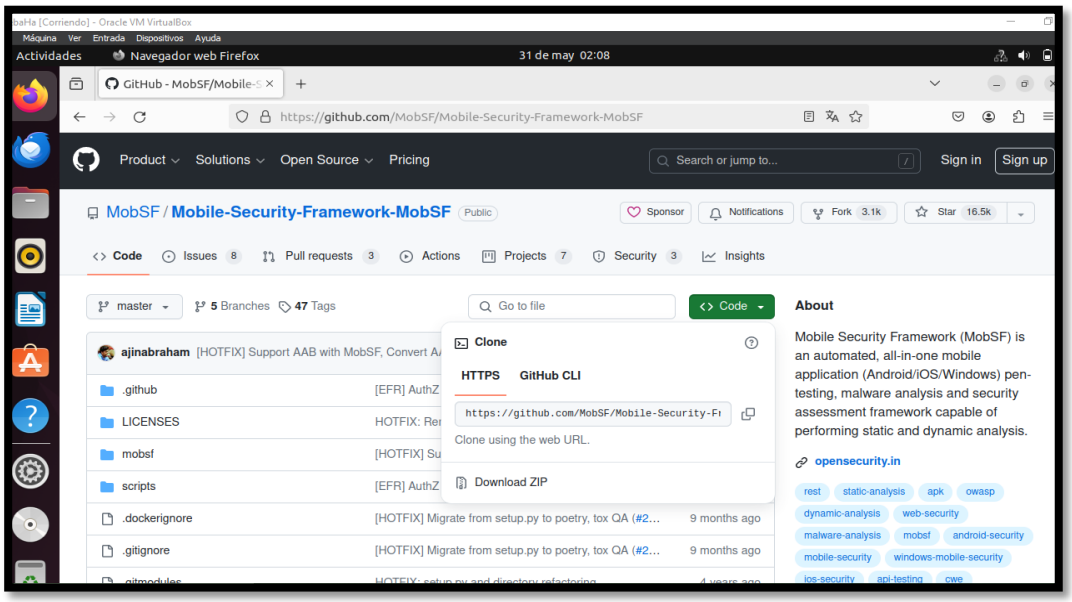


Imagen 1: Descargar el repositorio de mobsf desde github

2. Clonación del repositorio

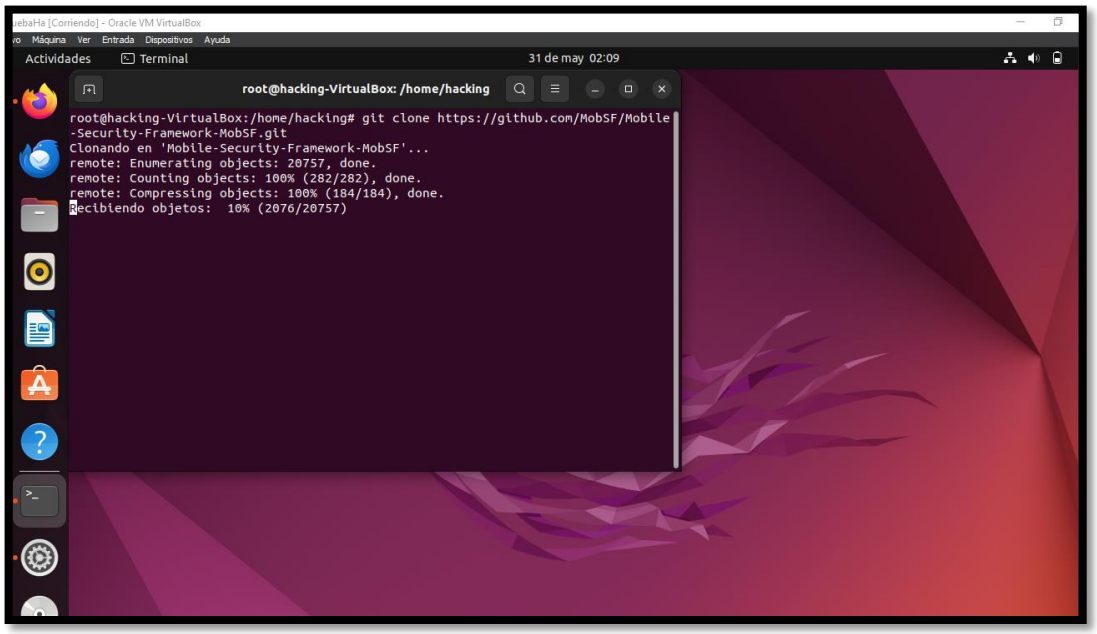


Imagen 2: Clonar el repositorio Mobsf en máquina virtual Ubuntu

3. Dirigir a la carpeta de MobSF

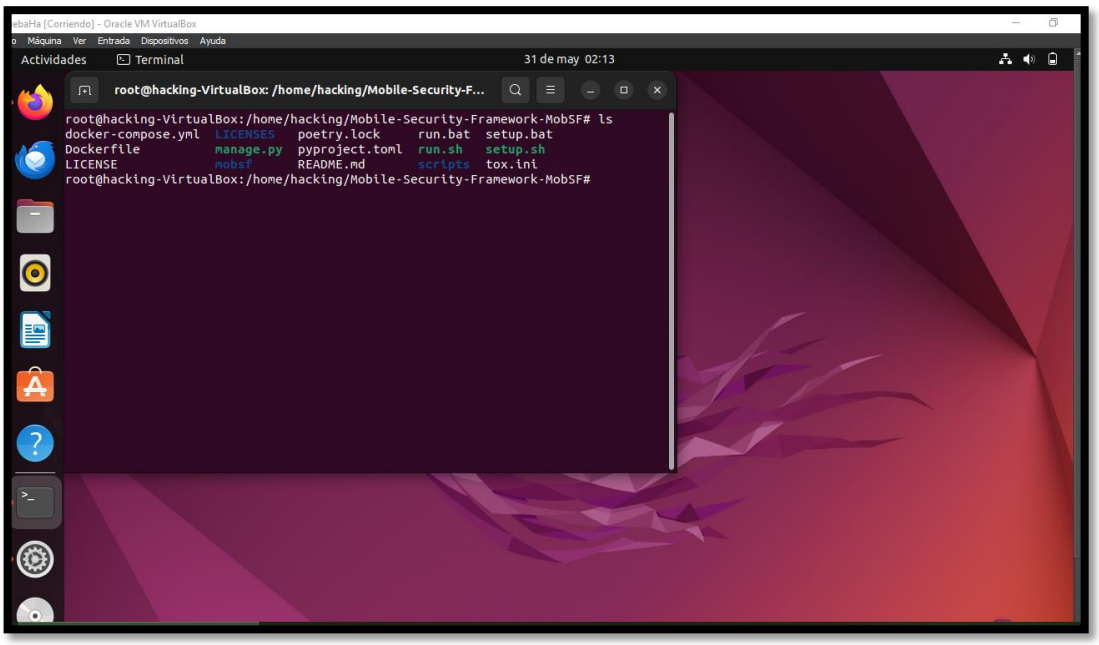


Imagen 3: Ir a la carpeta Mobsf para seguir con la configuración

4. Con el comando ls – la permite listar los archivos en la carpeta para hallar el setup de instalación, con el comando ./setup empieza la instalación de las dependencias para la herramienta

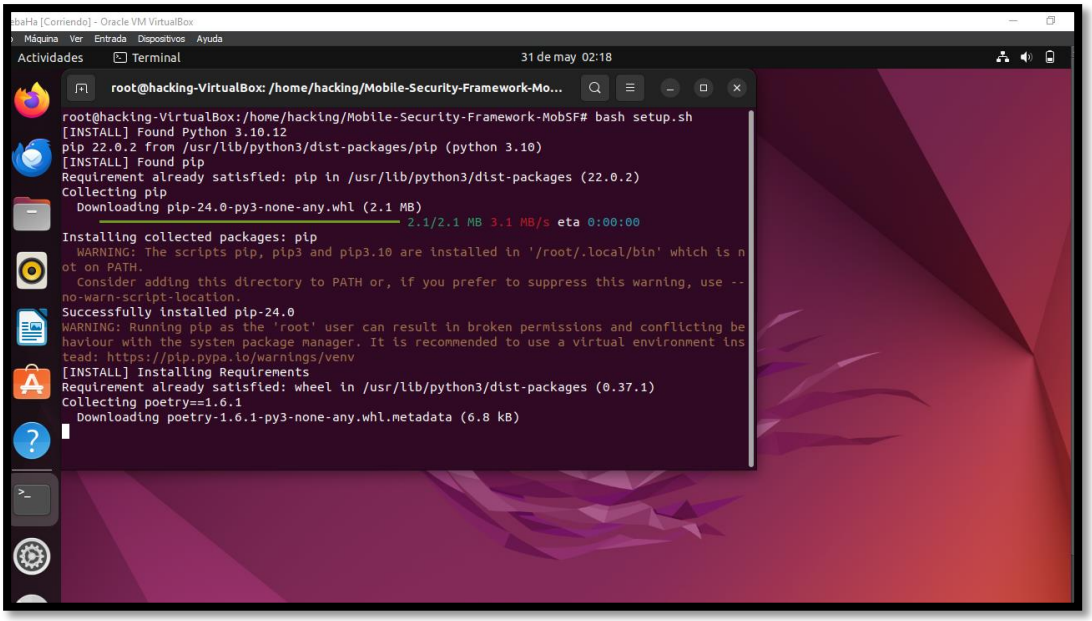


Imagen 4: Ejecutar el archivo setup para instalar dependencias

5. Proceso de instalación

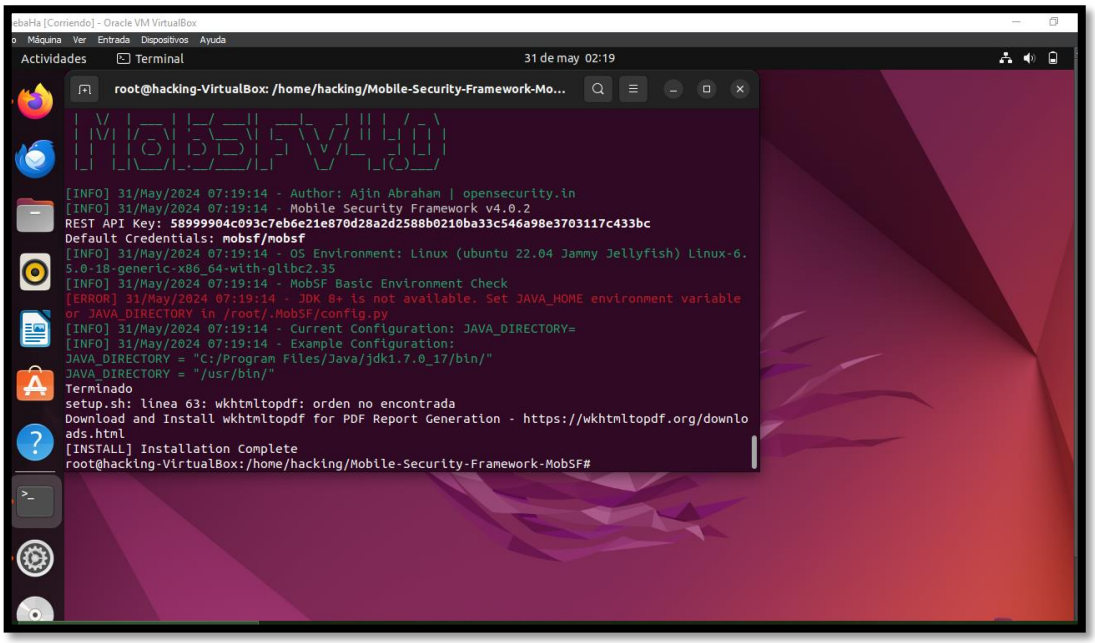


Imagen 5: Instalación de mobsf completa

6. Ejecutar el archivo run.sh

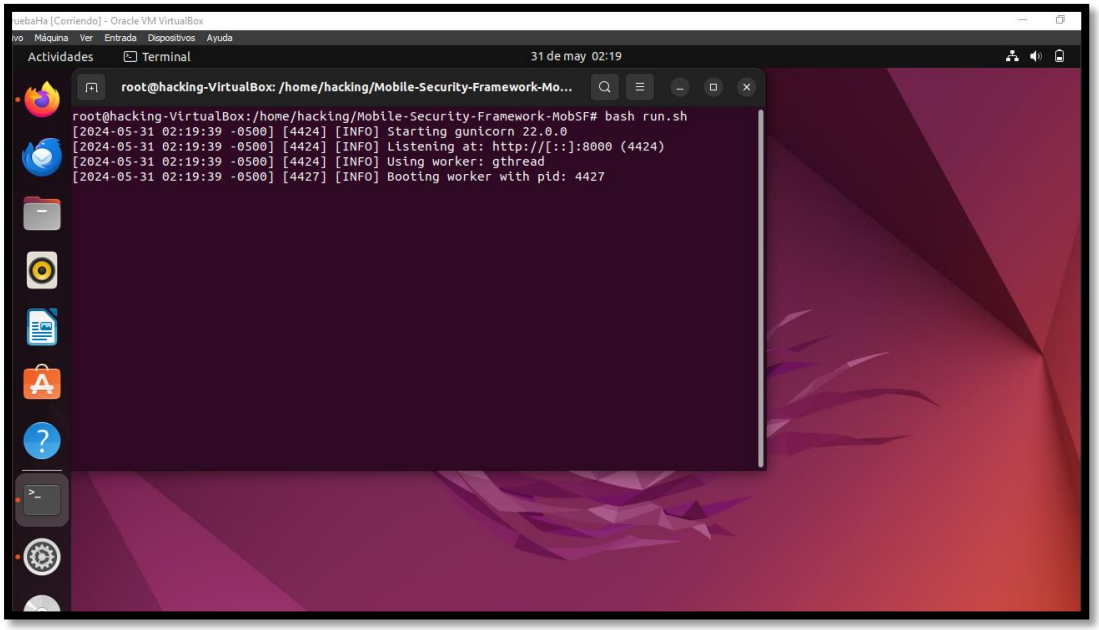


Imagen 6: Ejecución de archivo run.sh para empezar Mobsf

7. Portal de Inicio

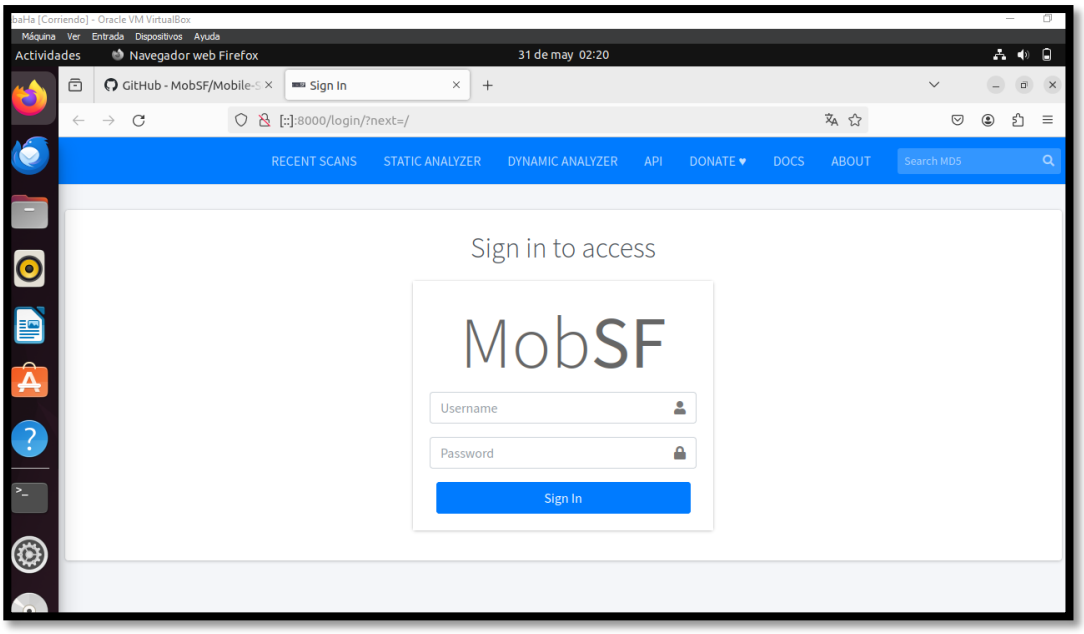


Imagen 7: Portal de Mobsf

INSTALACION DE LA HERRAMIENTA DROZER

8. Descargar Python 2.7.18 para ejercer la instalación correcta de la versión de drozer para el análisis de malware de forma dinámica

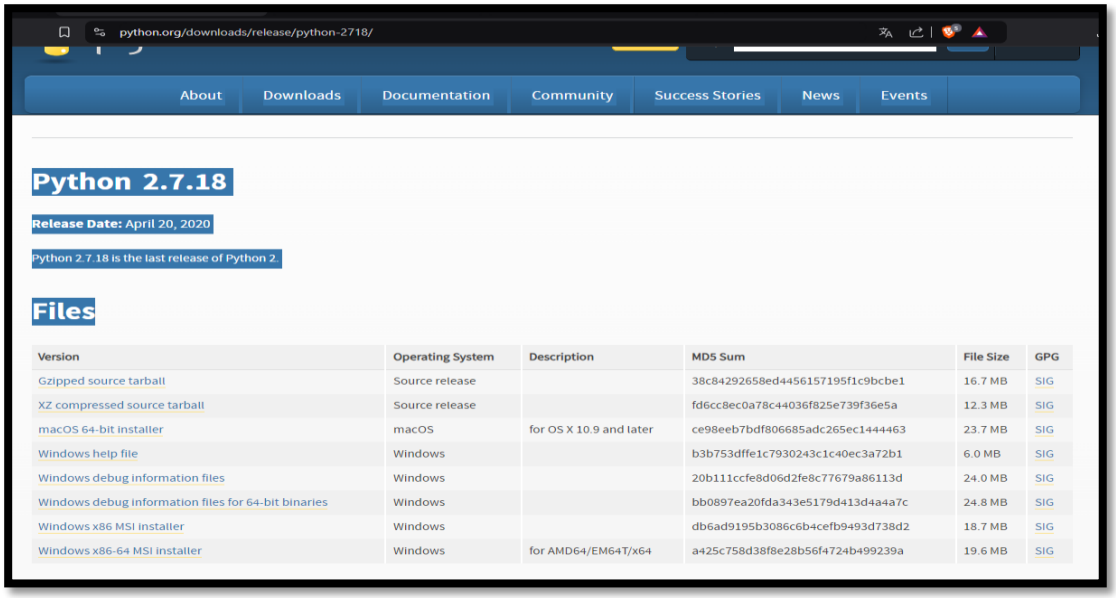


Imagen 8: Instalación de Python 2.7.18 en Windows

9. Abrir la terminal una vez que se haya instalado Python para efectuar la instalación de algunas librerías esenciales para drozer

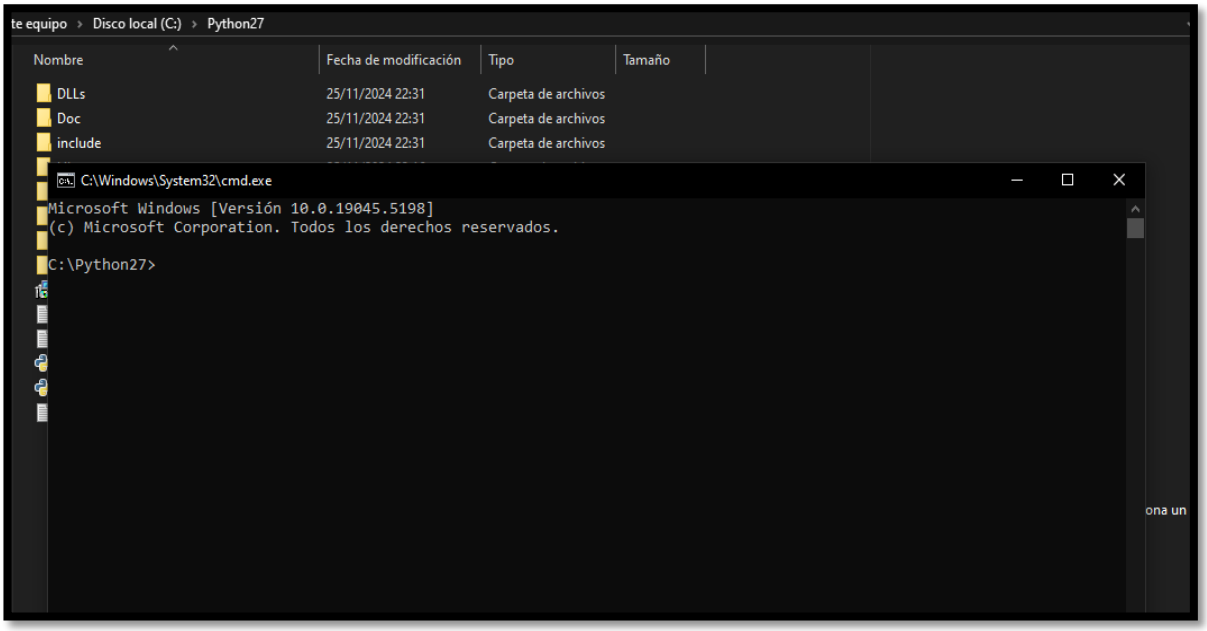


Imagen 9: Instalación completa de Python – Instalar dependencias para Drozer

10. Con el comando Python.exe –m pip install protobuf se instala protobuf, se utiliza para serializar y deserializar datos estructurados de manera eficiente. Esto es especialmente útil en aplicaciones que transmite datos en sistemas o almacenarlos de forma compacta.

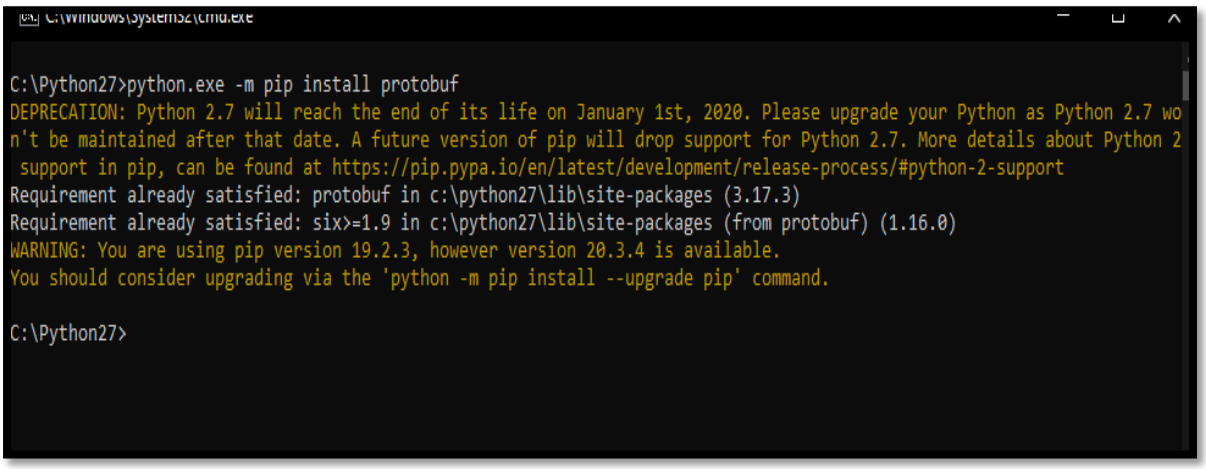


Imagen 10: Instalación de protobuf

11. Con el comando “Python.exe –m pip install pyopenssl”, sirve para emplear operaciones relacionadas con la seguridad, como por ejemplo manejar certificados SSL/TLS, establecer conexiones seguras y realizar tareas de criptografía general.

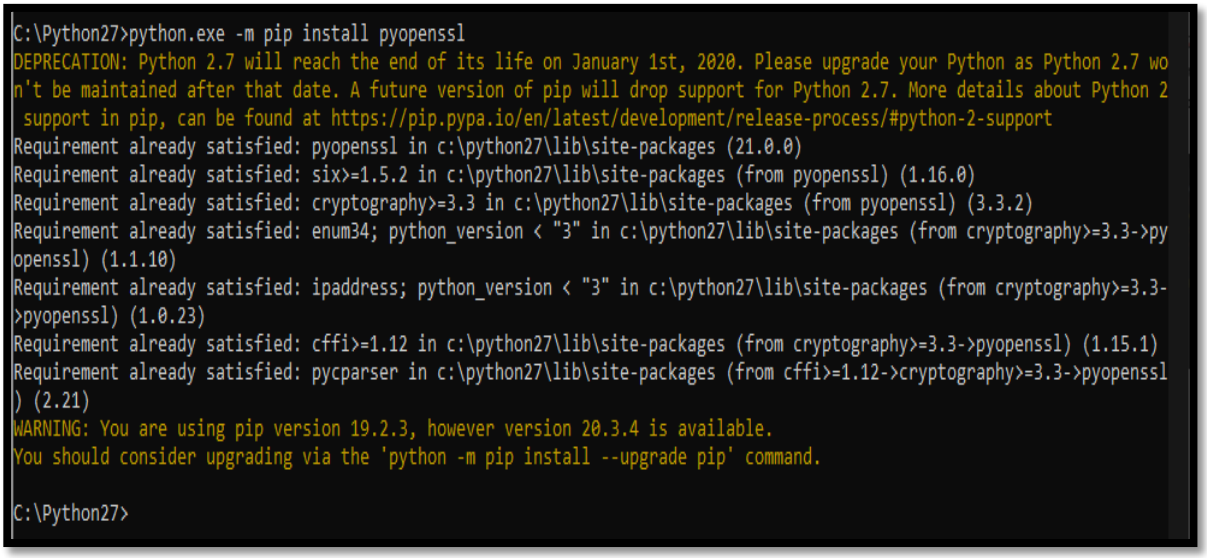


Imagen 11: Instalación de pyOpenssl

12. Con el comando “Python.exe –m pip install twisted”, está diseñado para construir aplicaciones que necesitan manejar conexiones de red, como servidores, clientes, con gran escalabilidad y eficiencia.

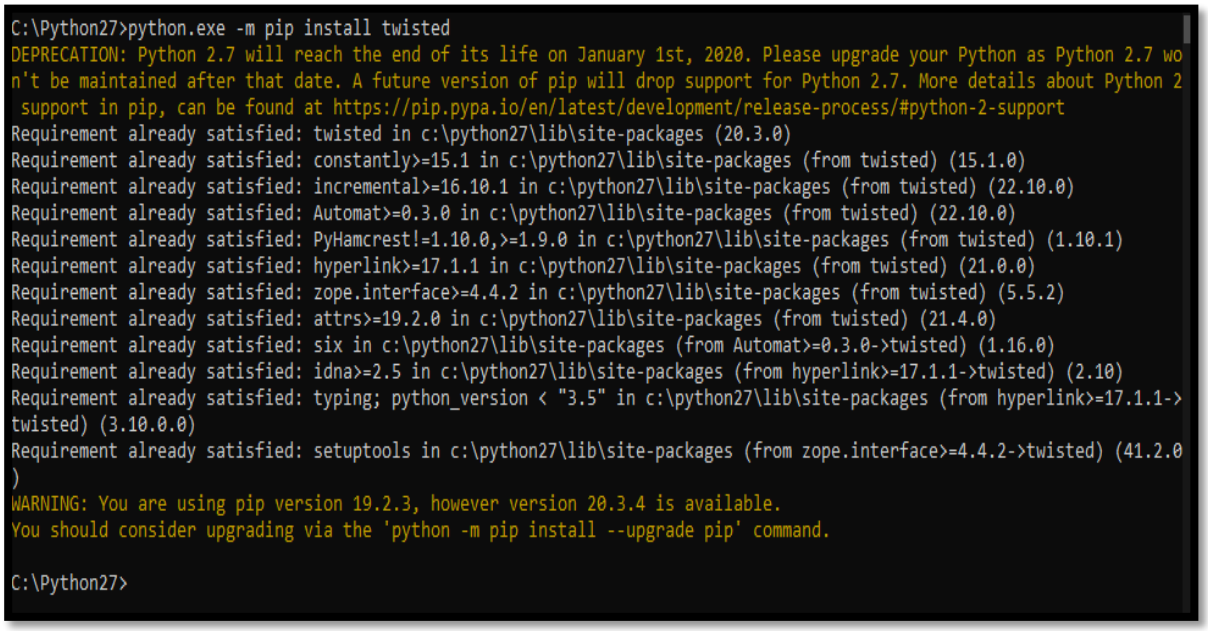


Imagen 12: Instalación de Twisted

13. Una vez instalado las librerías necesarias, descargar el client de drozer desde drozer withsecure labs la versión de drozer 2.4.4 en win32.msi

Imagen 13: descargar el client Drozer para Windows

14. Para poder descargar el archivo se necesita desactivar el antivirus y el análisis de archivos maliciosos en tiempo real, como así mismo el análisis de red.

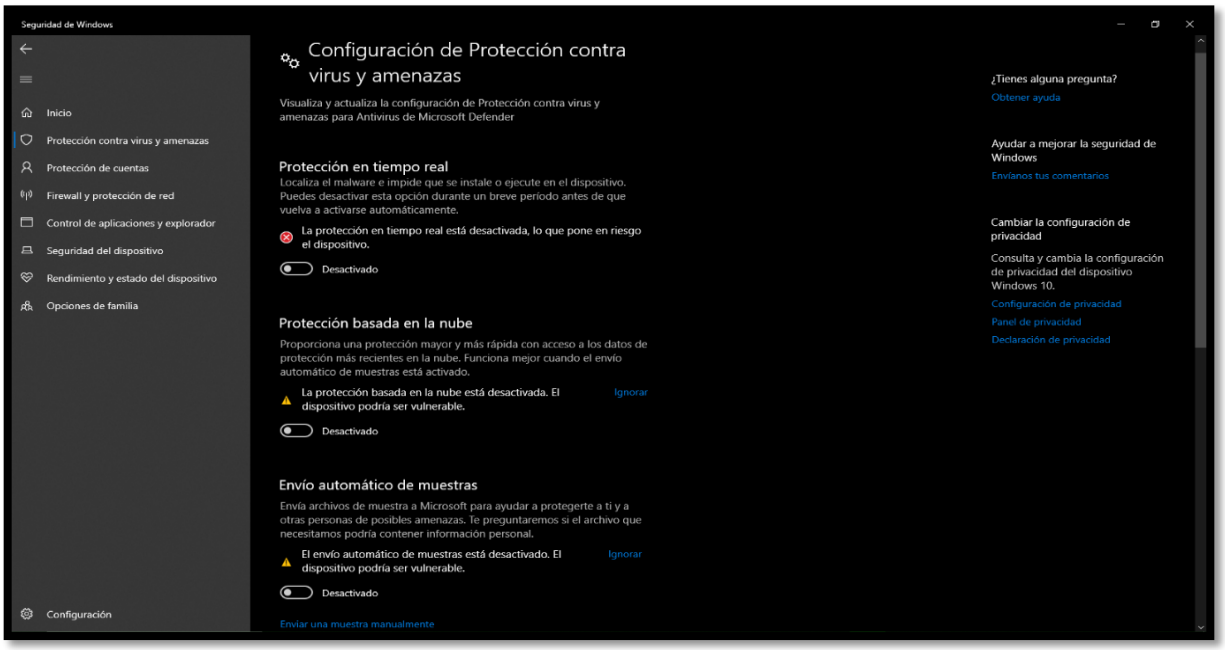


Imagen 14: Desactivar la protección contra virus y amenazas para descargar el client Drozer

15. El archivo una vez que las configuraciones de seguridad de Windows defender estén off, el archivo es descargable para que así sea guardado desde python2.7 para su instalación y dependencia desde ese repositorio

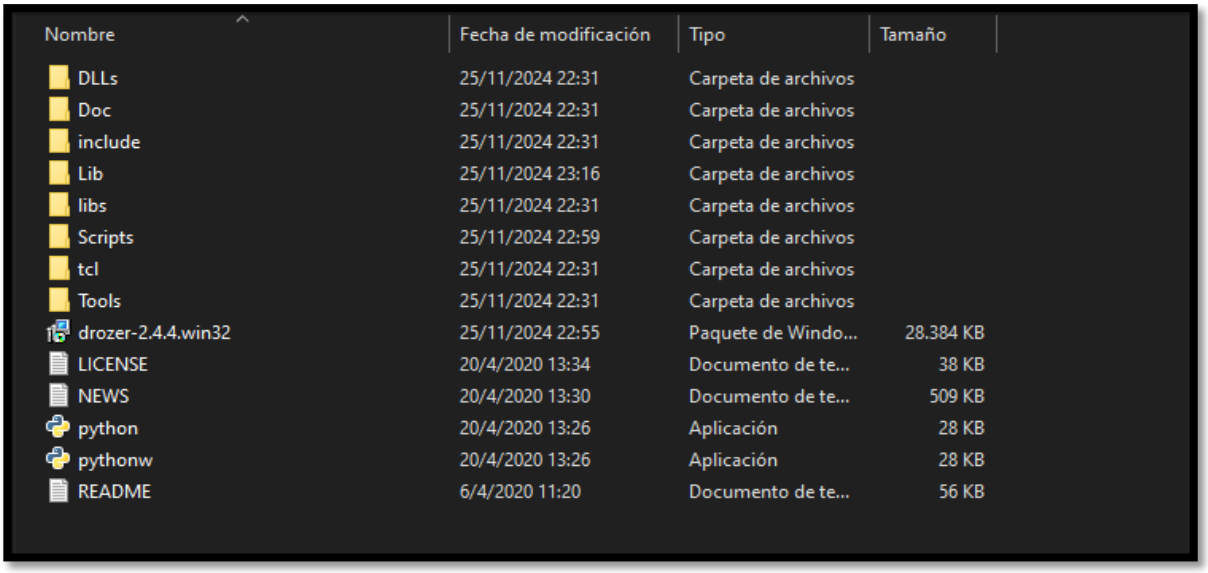


Imagen 15: Copiar el archivo Drozer en la carpeta de Python 2.7.18

16. Se ejecuta la instalación de drozer desde Windows con siguiente, siguiente, y siguiente

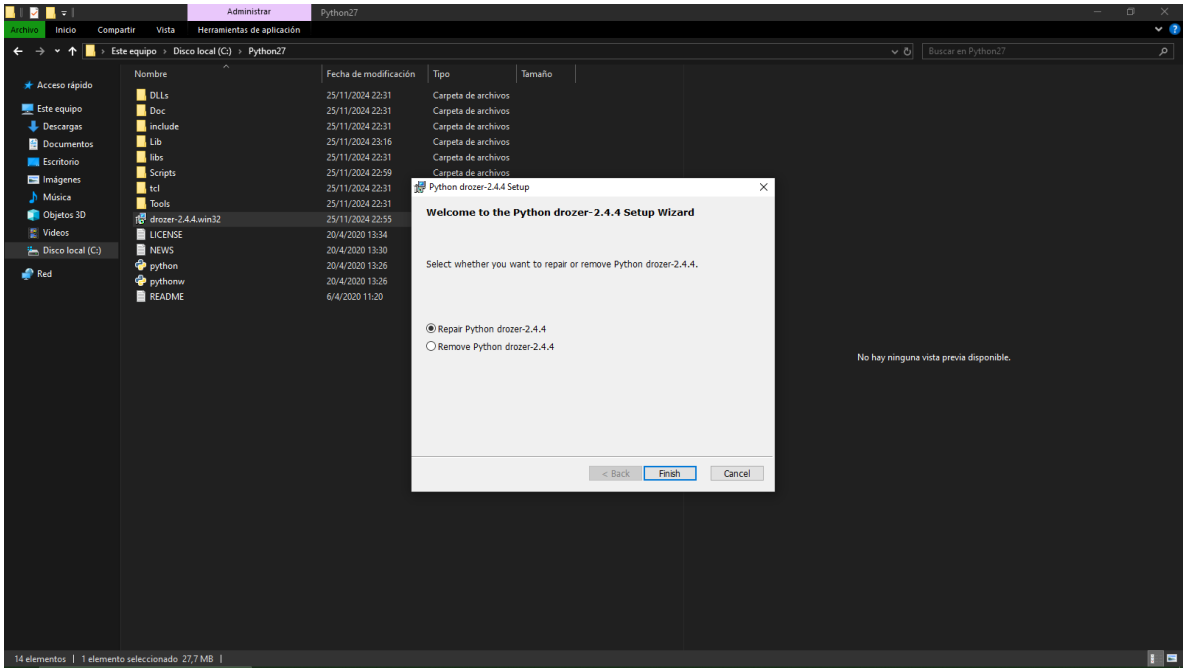


Imagen 16: Ejecutar el ejecutable para la instalación de Drozer

17. Para comprobar la instalación de drozer correctamente se ejecuta el comando “drozer”

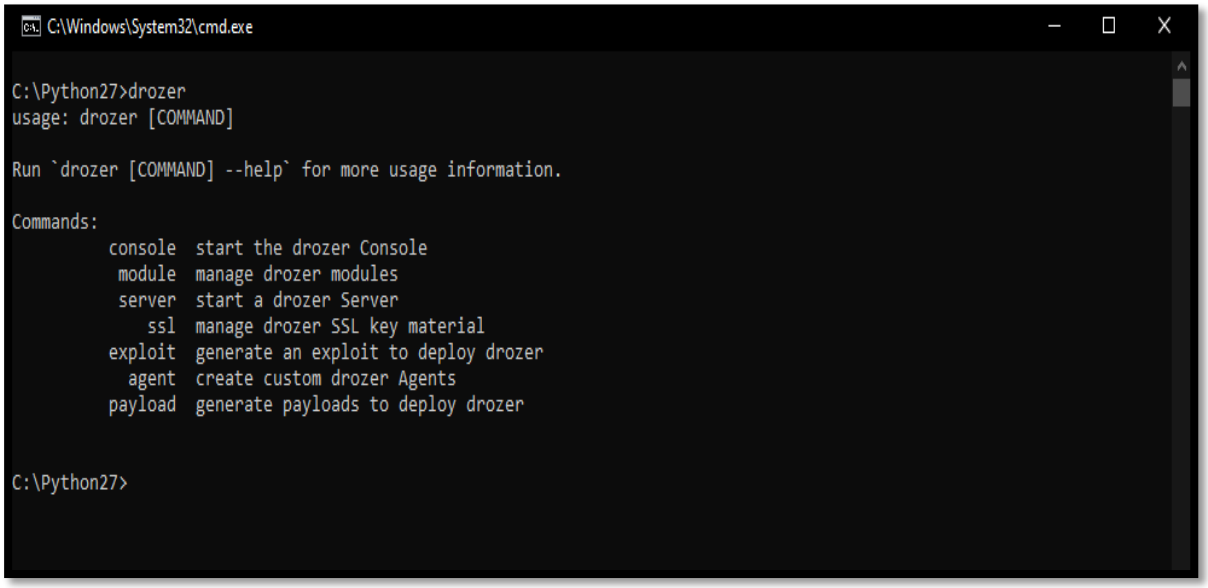


Imagen 17: Finalizado la instalación de Drozer

18. Ejecutar el comando drozer console connect para verificar la interfaz de drozer y saber que debug se hallan antes de probar

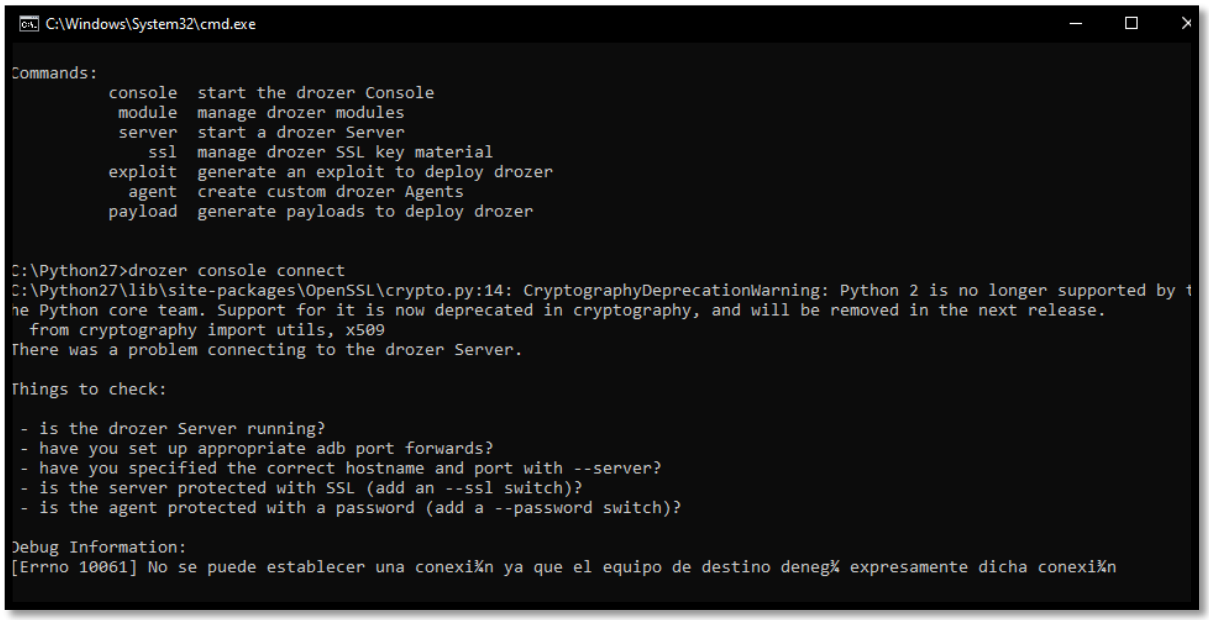


Imagen 18: Verificación de la interacción de Drozer

19. Con el comando “Python.exe –m pip install service_identity”, sirve para identificar y validar correctamente los certificados SSL/TLS durante la conexión de un servidor.

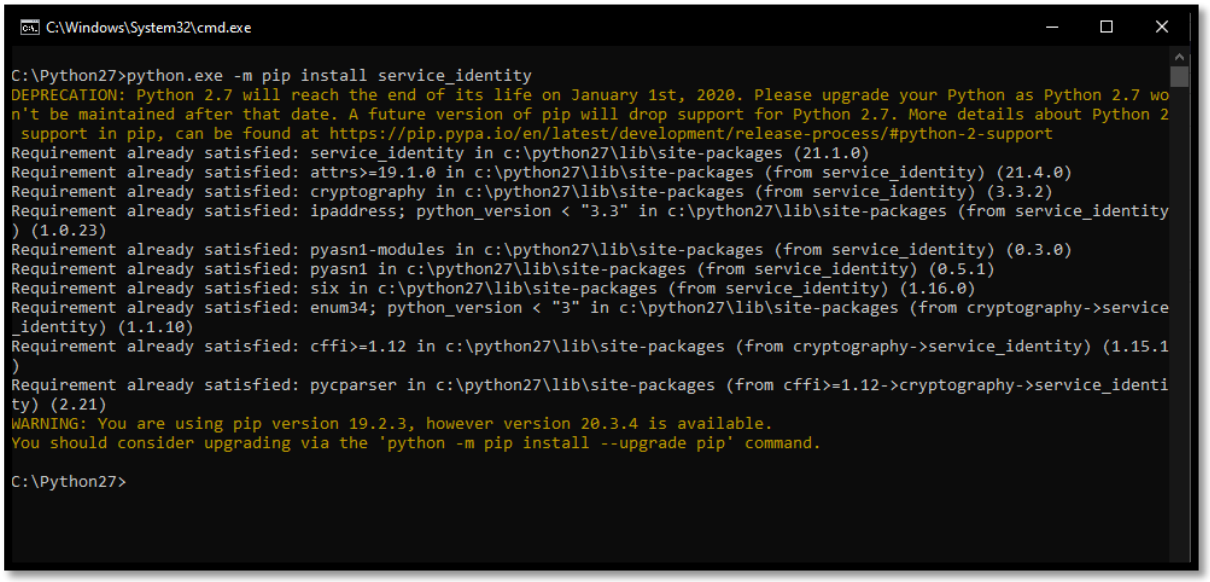


Imagen 19: Instalacion de service_identity

DROZER AGENT – ANDROID

20. Se ejerce la descarga de la herramienta Drozer Agent para lograr la interacción entre el client drozer y el agent . Se descargar el agent Drozer 2.5.0

Imagen 20: Descargar el agent Drozer para Android

21. Archivo descargado exitosamente

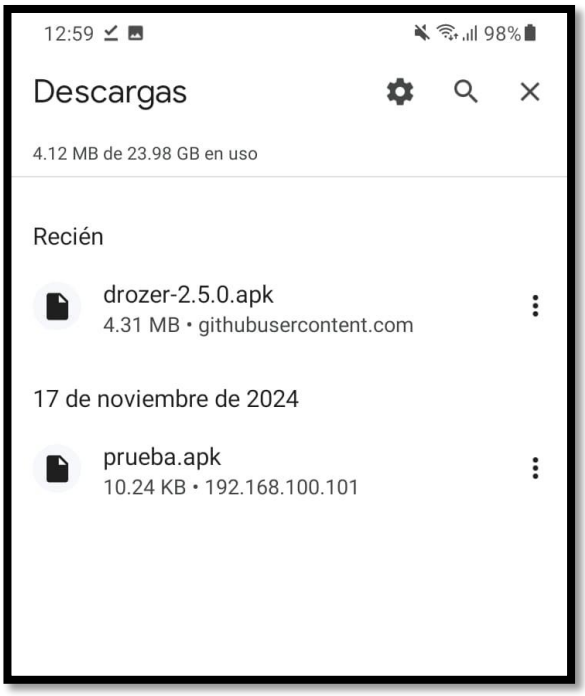


Imagen 21: Drozer.apk descargado exitosamente

22. Ejecutar la instalación de la aplicación

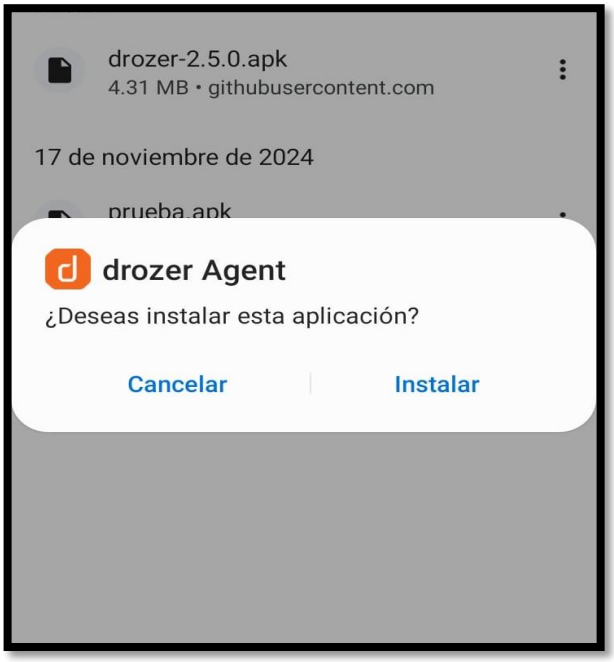


Imagen 22: Instalar Drozer

23. Drozer instalado correctamente con la opción de server en off

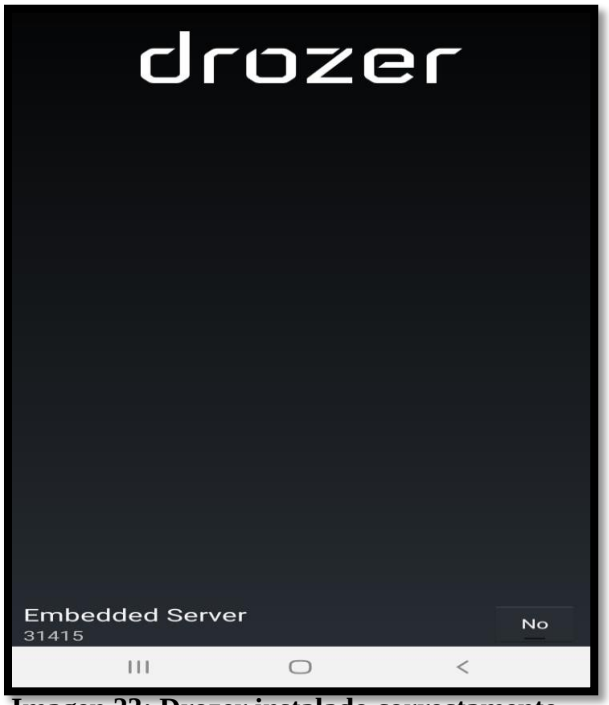


Imagen 23: Drozer instalado correctamente

24. Se da por encendido el server drozer del agent que cuenta con puerto 31415

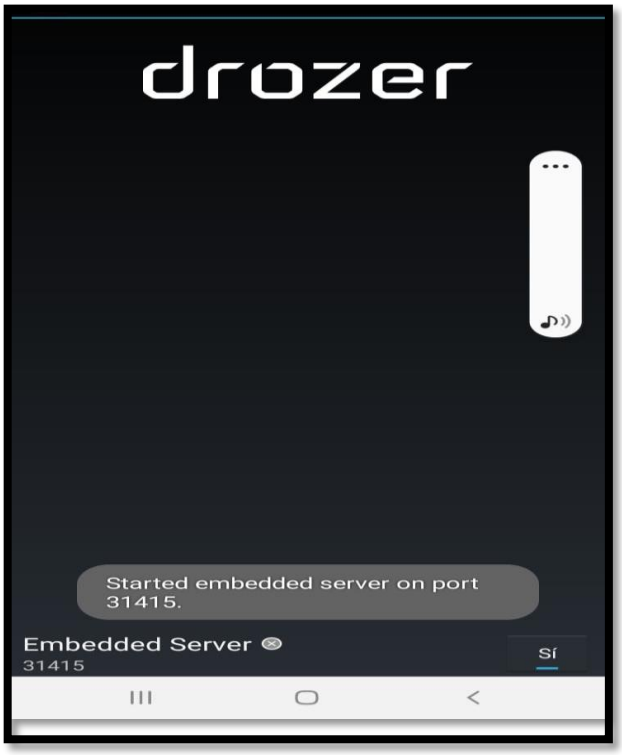


Imagen 24: Activar la opción Embedded Server

25. Para interactuar con las herramientas se necesita la instalación de Android Tools Plataforms para ejercer la visualización de archivo, conexión, entre otros. Dar clic en SDK Windows.

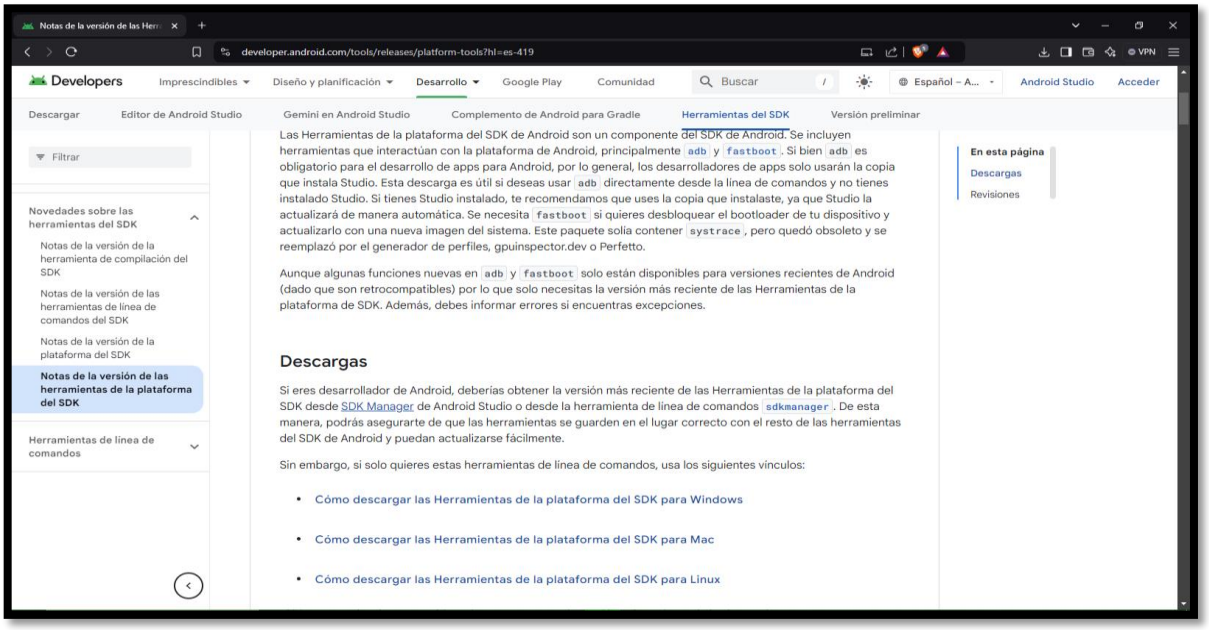


Imagen 25: Descargar Android Tools Plataforms

26. Una vez aceptado SDK para Windows aceptar los términos y descargar la herramienta

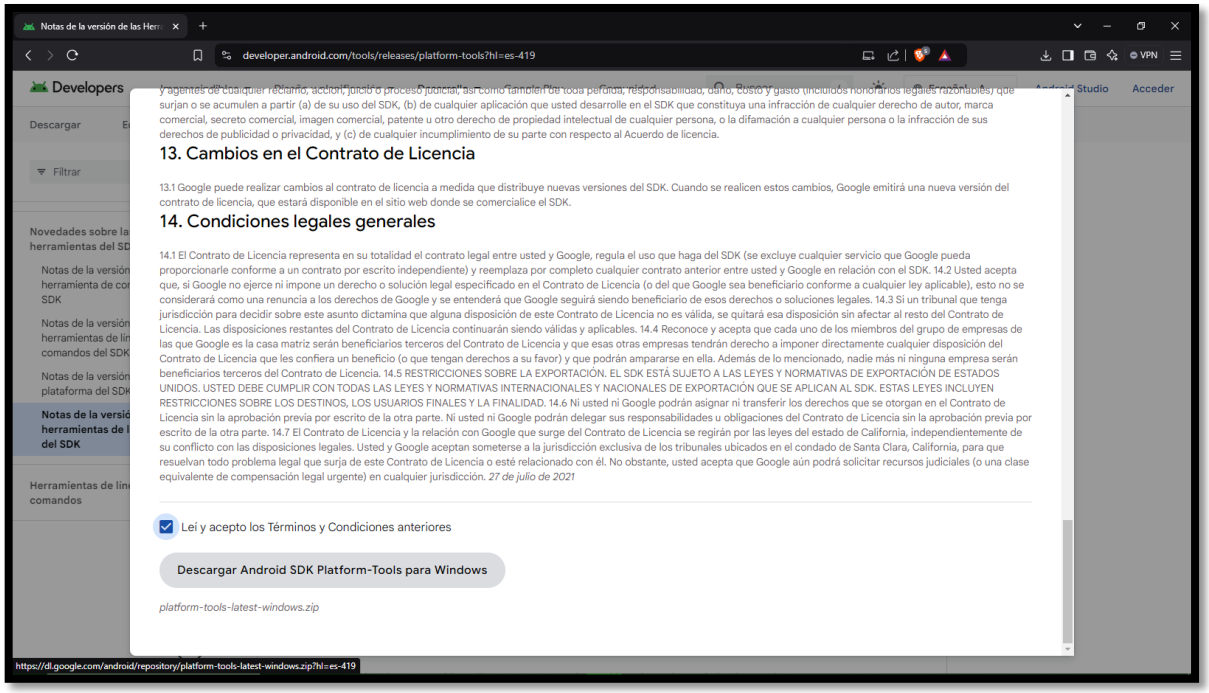


Imagen 26: Condiciones para descargar el SDK de Android Tools Plataforms

27. Guardar la descarga

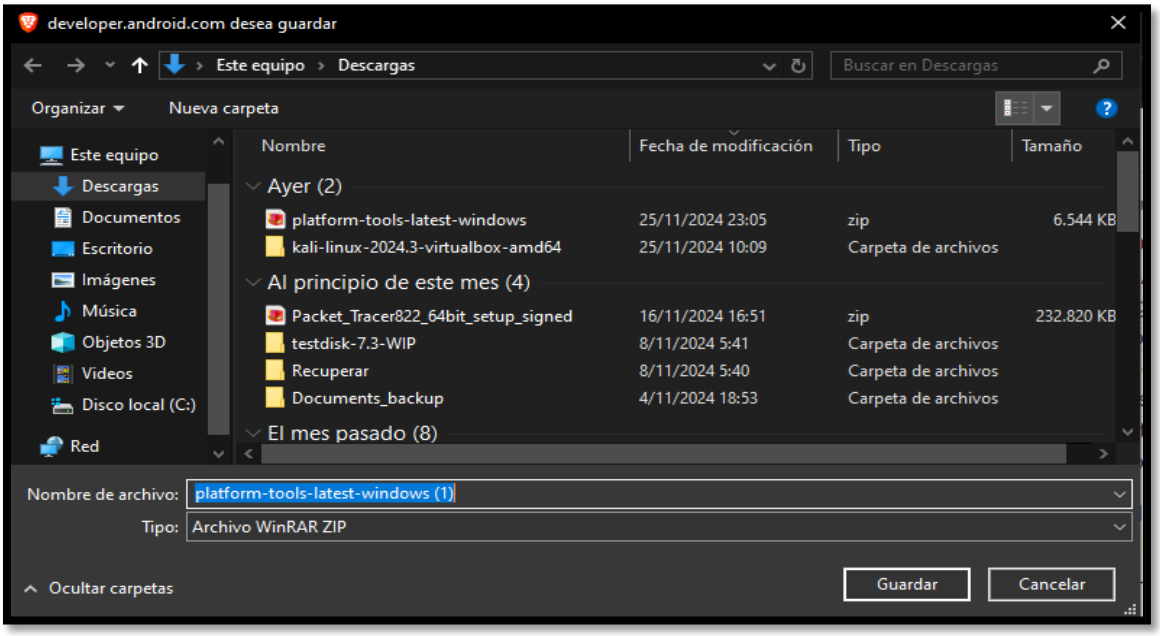


Imagen 27: Plataforms descargado

28. Una vez descargado el archivo, se da en extraer aquí para descomprimir la carpeta del archivo almacenado para así copiar y pega en archivos programas(x86) en el disco.

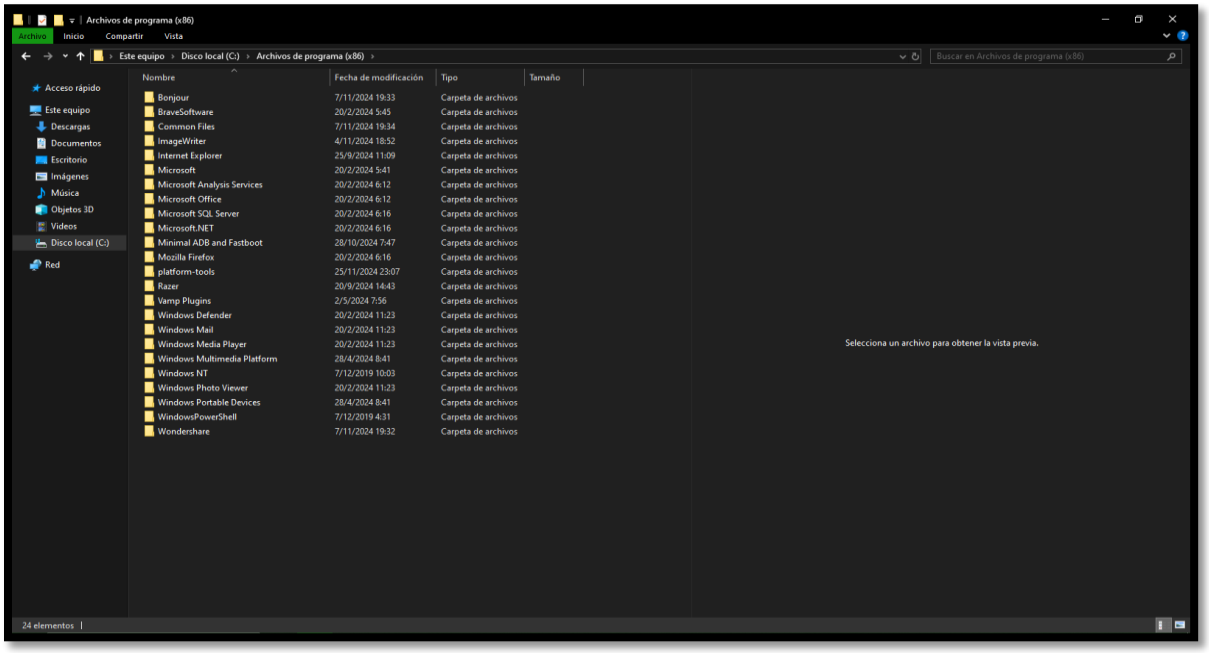


Imagen 28: Extraer el archivo del formado zip

29. Ahora editar las variables de entorno para que la herramienta funcione correctamente efectuando los comando tanto en usuario como administrador. Para ello, en el panel de Windows escribir “editar variables”

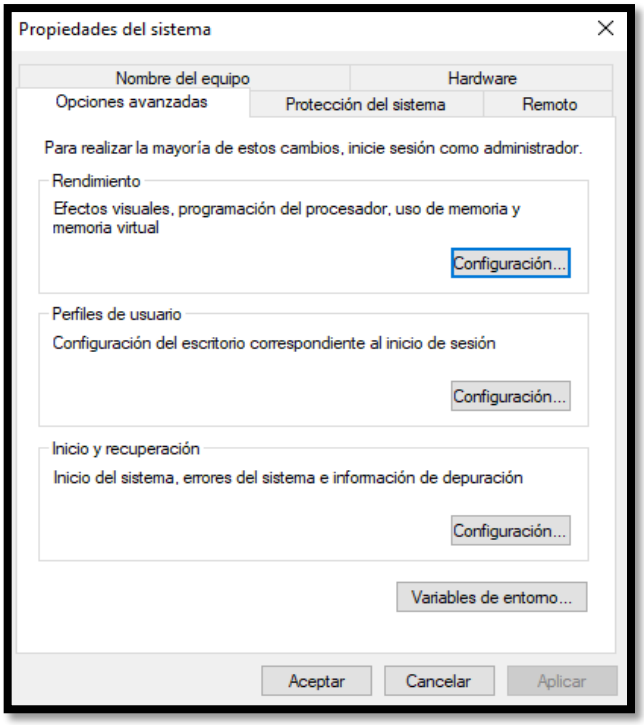


Imagen 29: Editar variables de entorno para el funcionamiento del SDK

30. Dar clic en variables de entorno y seleccionar en variables de entorno del sistema la sección “PATH”

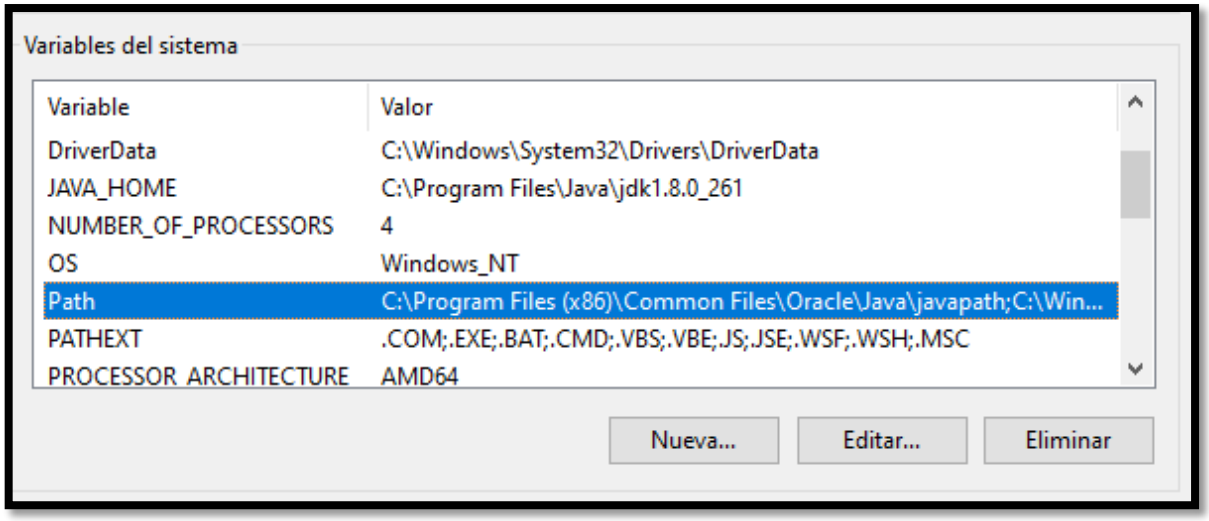


Imagen 30: Editar el path para insertar la ruta del SDK

31. Dar clic en editar y pegar la direccion en donde se encuentra la herramienta adb que corresponde la carpeta “plataforms_tools”, para ello dar clic en nuevo y pegar la ruta y así aceptar, aceptar en todo.

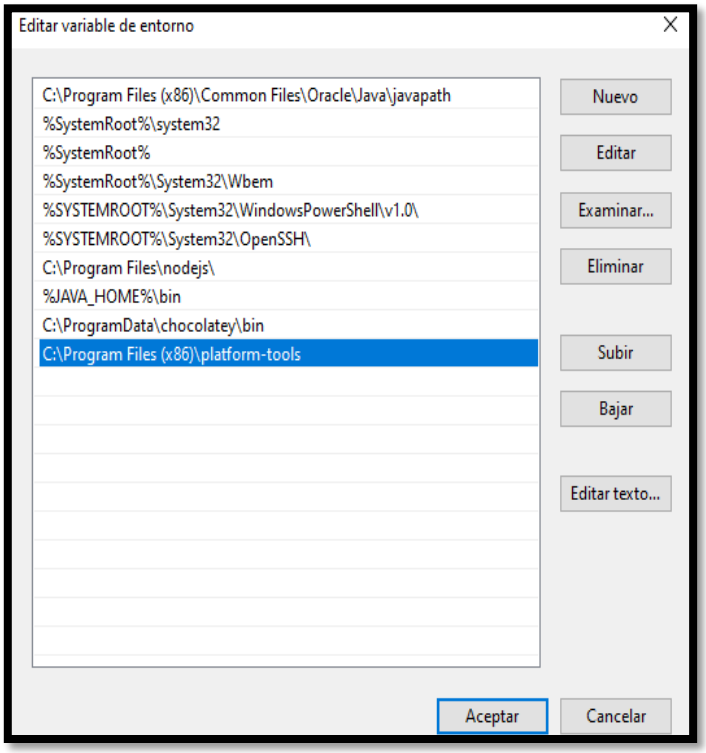


Imagen 31: Insertar la ruta del plataforms tolos

32. Una vez desarrollado aquello se realiza la prueba desde la CMD ejecutando adb

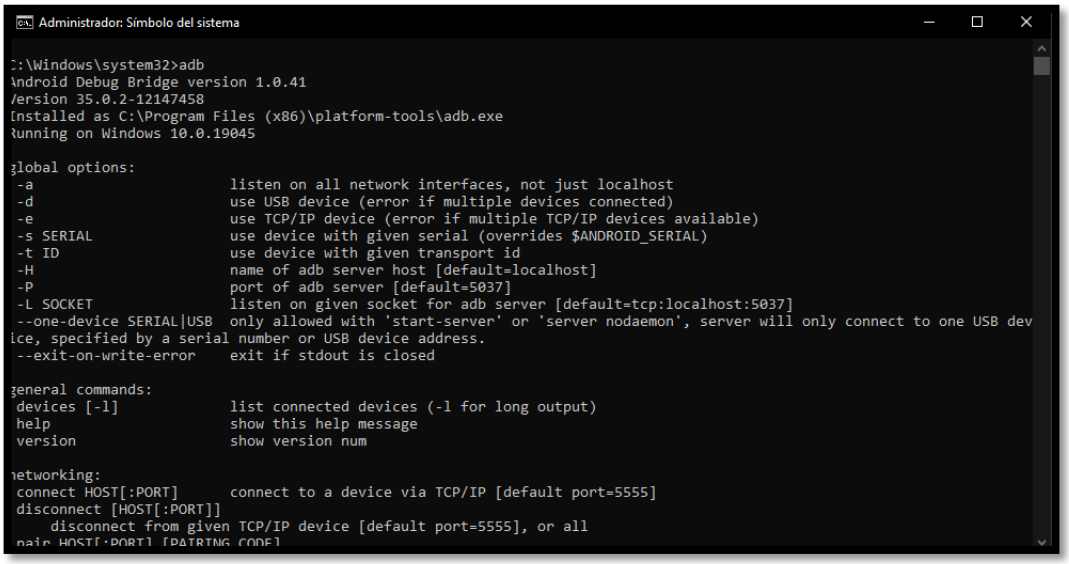


Imagen 32: Prueba de ADB en el CMD

33. Ya instalado la herramienta adb, ejecutar el comando adb devices para conocer el dispositivo conectado para efectuar la interacción con drozer, conectar el dispositivo con cable USB y efectuar el permiso de transferencia de archivos para dar permiso. Y una vez realizado el comando adb devices – sale el identificador del móvil con el status “devices” significa que esta para uso.

Imagen 33: Ejecución adb devices para conocer los dispositivos conectados

34. Luego ejecutar el comando “adb forward tcp:31415 tcp:31415 para permitir la escucha desde el puerto del server del agent hacia el client de drozer en Windows

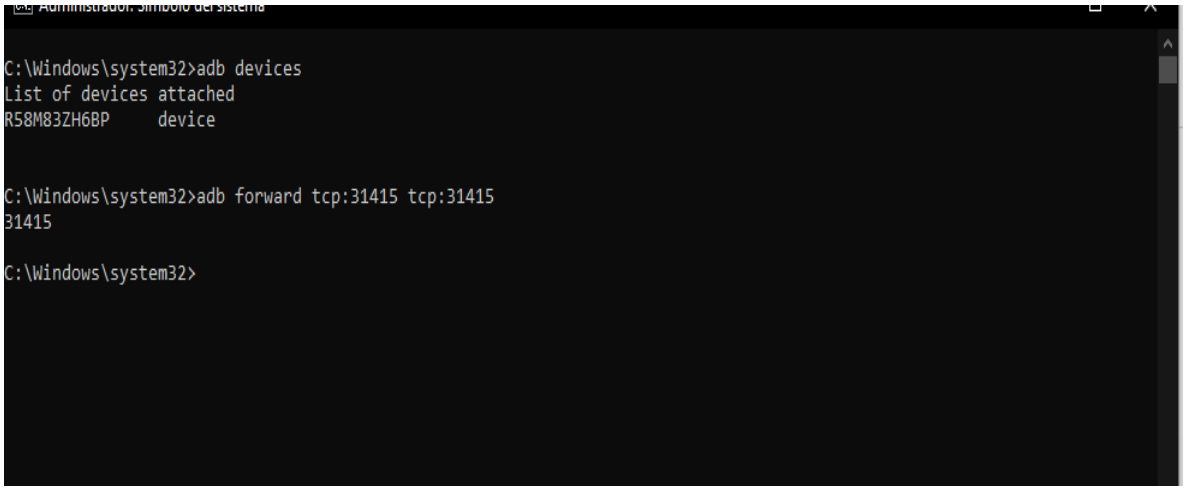


Imagen 34: Configuracion del tcp con el puerto del agent Drozer para la escucha

35. Ahora desde la terminal de python2.7 efectuar el comando “drozer console connect” tomando en cuenta que la aplicación drozer agent debe estar en modo open para su funcionamiento, no en modo reposo o no abierta, ya luego de su conexión en modo reposo se interactúa sin problema.

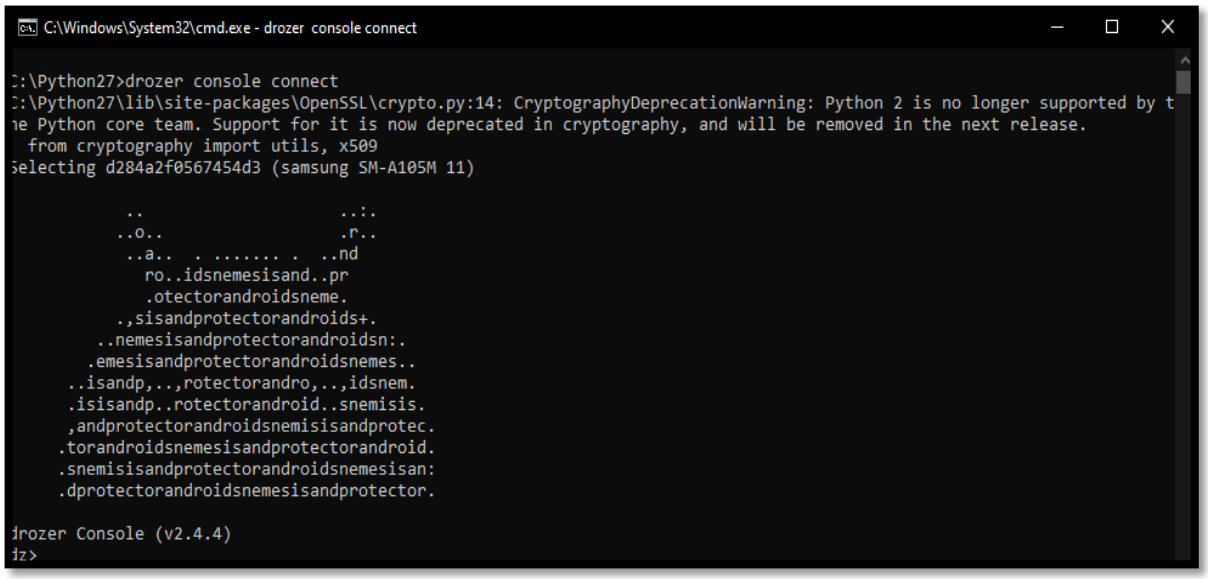


Imagen 35: Ejecución de Drozer console connect con el dispositivo configurado

36. Con el comando list muestra las aplicaciones o archivos que el dispositivo Android cuenta.

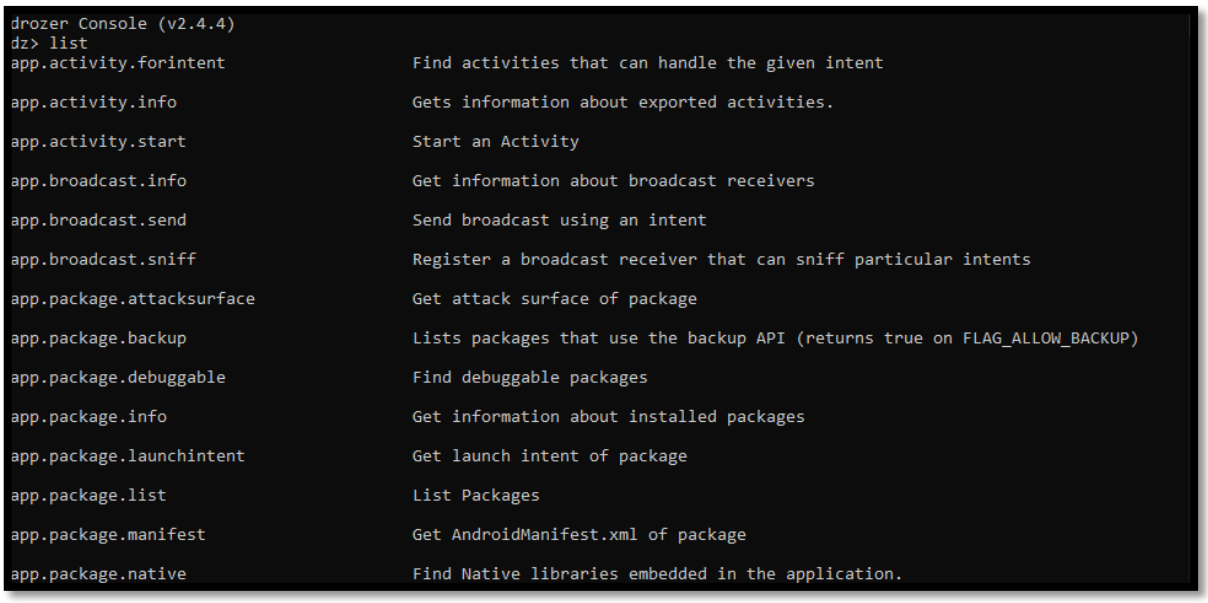


Imagen 36: Comando de ayuda de Drozer para analizar

ANEXO #4
FASE III: ANÁLISIS

ANÁLISIS ESTÁTICO - MALWARE´S

1. Se alza la herramienta MOBSF con el comando ./run.sh y se ingresa las credenciales de la herramienta predefinidad “mobsf/mobsf” para acceder a la interfaz.

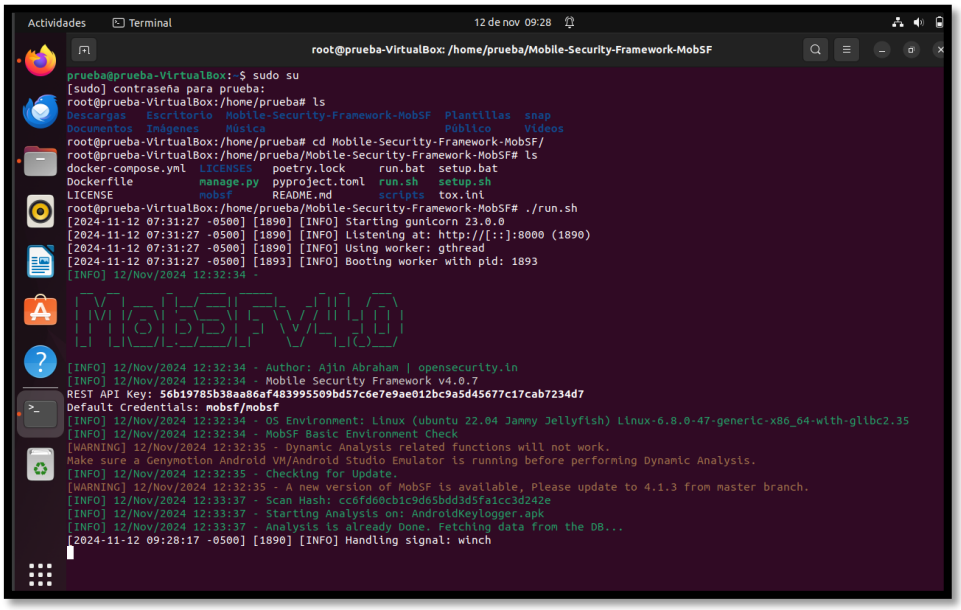


Imagen 37: Ejecución de mobsf para el análisis estáticos de las muestras

2. En la pantalla principal se presenta una interfaz de usuario para insertar o cargar el archivo analizar

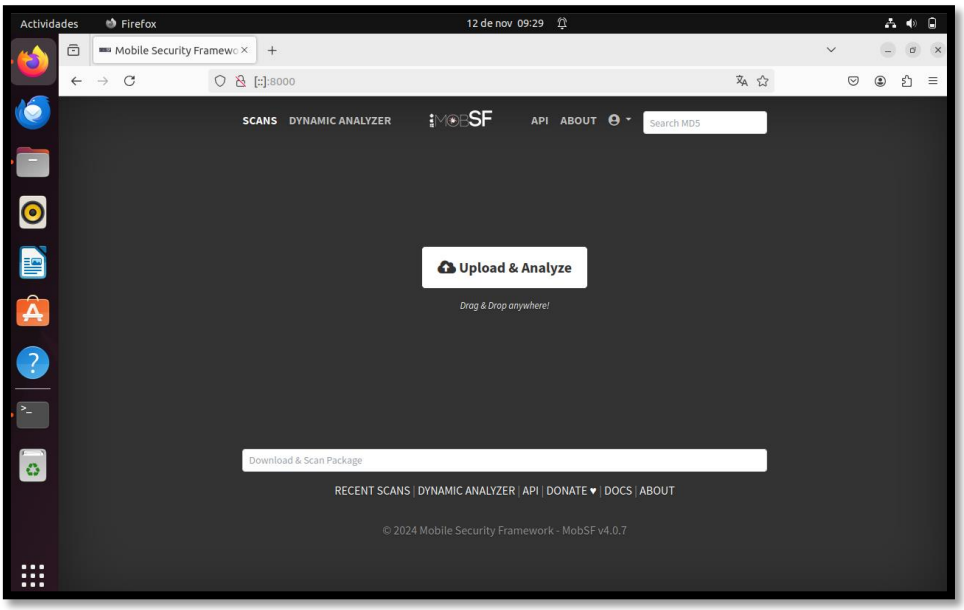


Imagen 38: Acceso a la interfaz de upload de archivo analizar

3. Se busca el archivo analizar por análisis estático y se le da en abrir para seleccionar

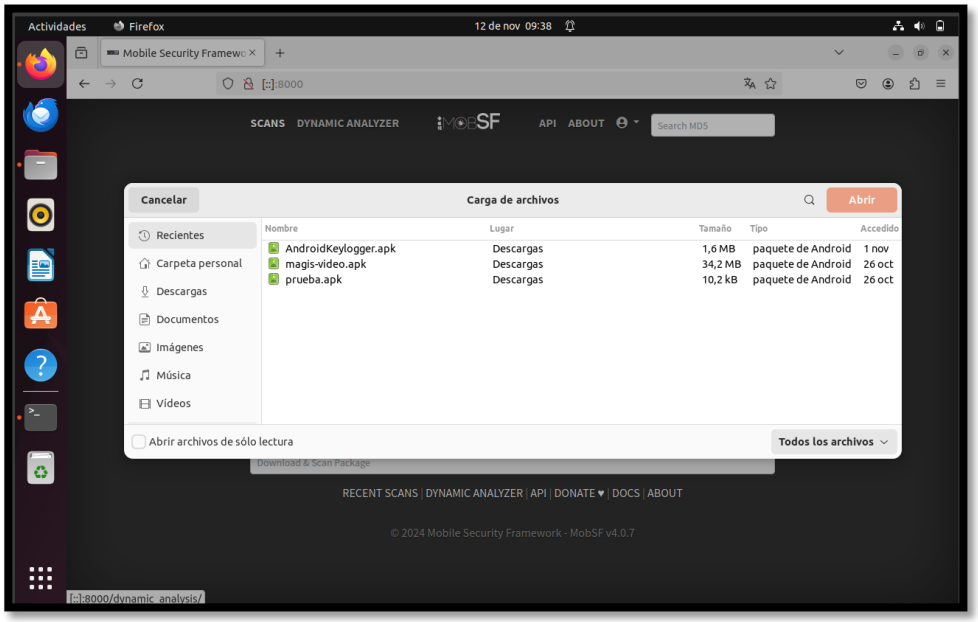


Imagen 39: Seleccionar que muestras analizar (Prueba, keylogger, Magistv)

RESULTADOS ANALISIS DE APK PRUEBA

4. Se ha seleccionado la aplicación prueba.apk para ejercer el análisis estático y ver qué información se logra encontrar.

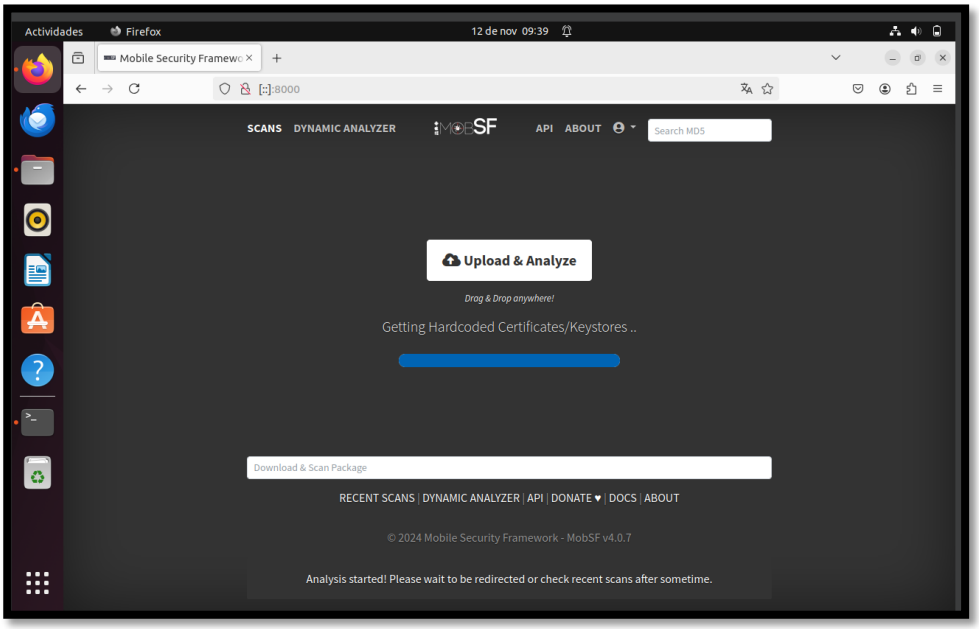


Imagen 40: Archivo Prueba.apk analizar

5. Una vez que finaliza el proceso de análisis del aplicativo APK a estudiar se observa la información general de la APK como nombre, encriptación, tamaño, entre otros aspectos.

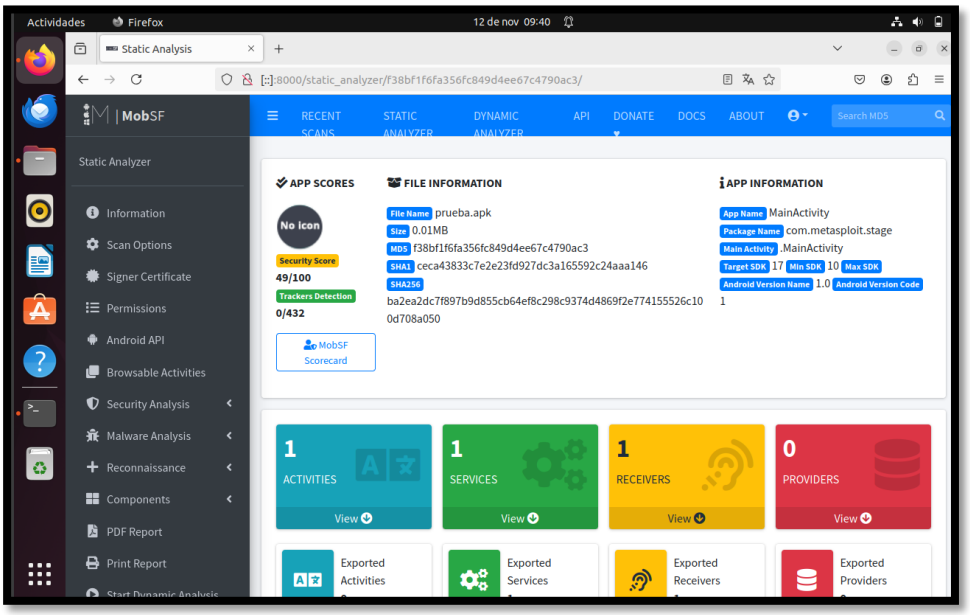


Imagen 41: Se encontró la información del package de la aplicación la info general

6. Se observa el archivo AndroidManifest.xml y se encuentra los permisos de la aplicación que necesita para acceder a recursos del sistema o información del usuario, filtros de intent, información de la aplicación, entre otros.

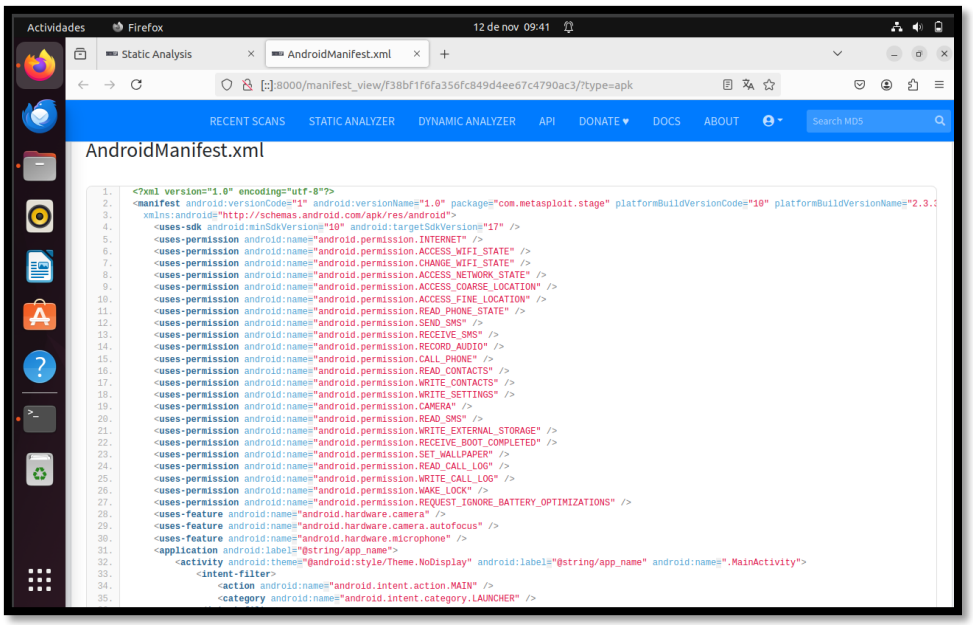


Imagen 42: Análisis del archivo AndroidManifest.xml se observa los permisos

7. Se observan los archivos smali que contiene la aplicación Android, smali es considerado como un ensamblador ya que ejecuta instrucciones del código Dalvik.

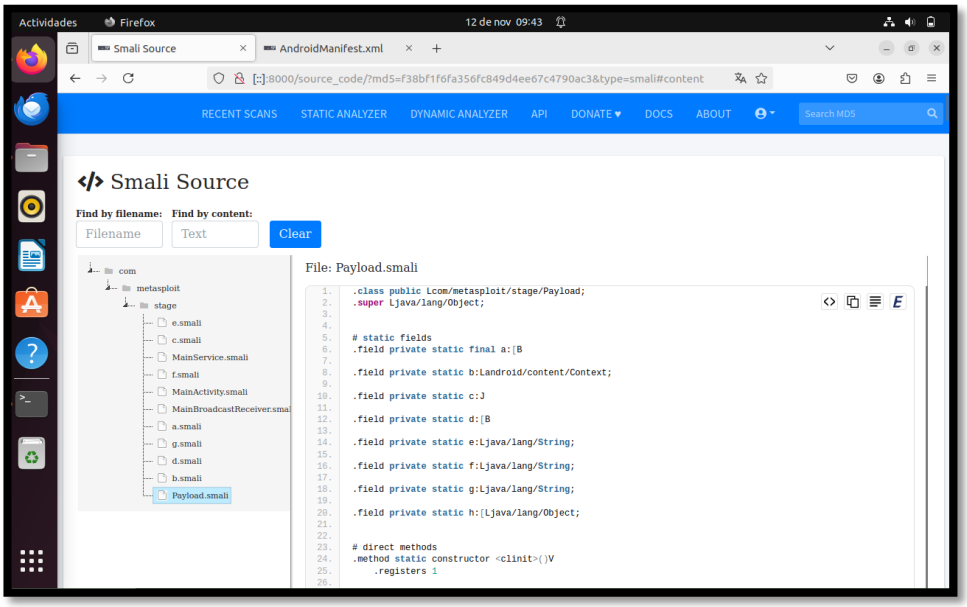


Imagen 43: Archivos smail que contiene configuraciones de send server

8. Se observa en el apartado Singer Certificate – certificado de firma, una parte esencial en el proceso de seguridad y verificación de identidad en Android, debido que permite al sistema y a los usuarios confiar en la aplicación y en su origen.

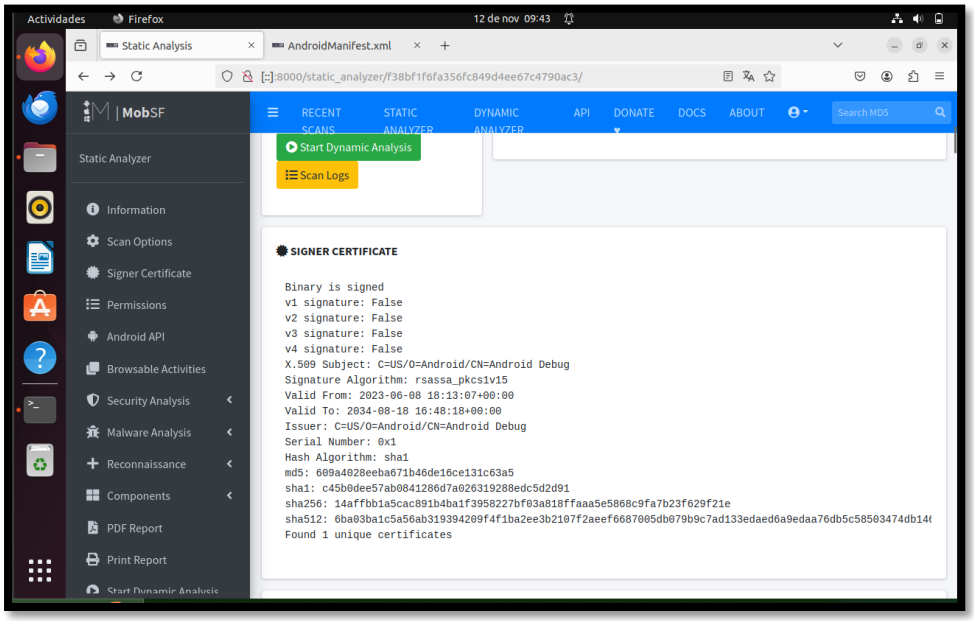


Imagen 44: Información del certificado de firma de la aplicación Prueba.apk

9. En el apartado se muestran los permisos que la aplicación requiere para acceder a los recursos del sistema y desarrollar su correcto funcionamiento, en ello se encuentra permisos de alto riesgo.

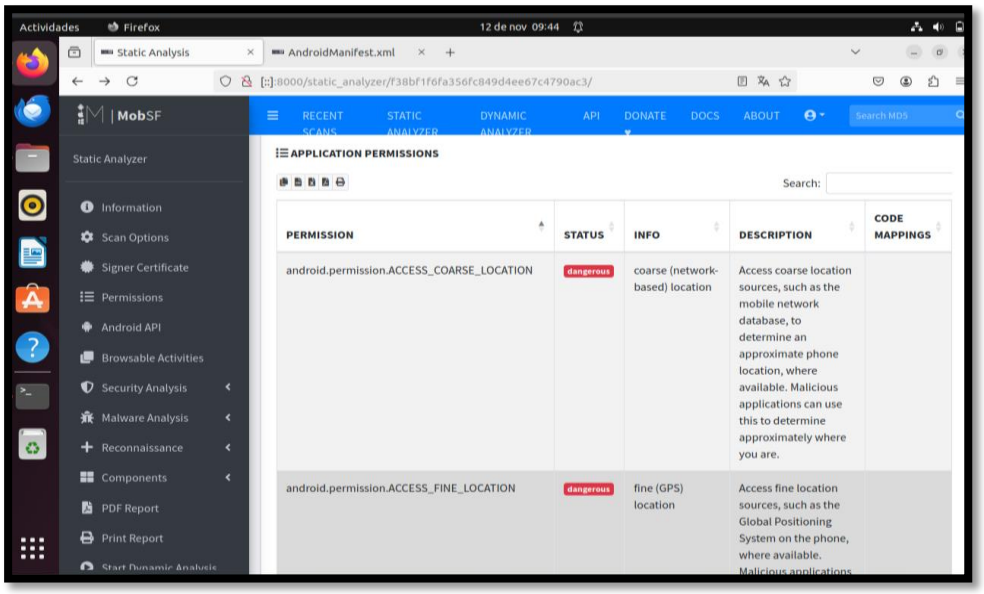


Imagen 45: Información de los permisos de Prueba.apk

10. Se encuentras las API´S asociada a la aplicación Android

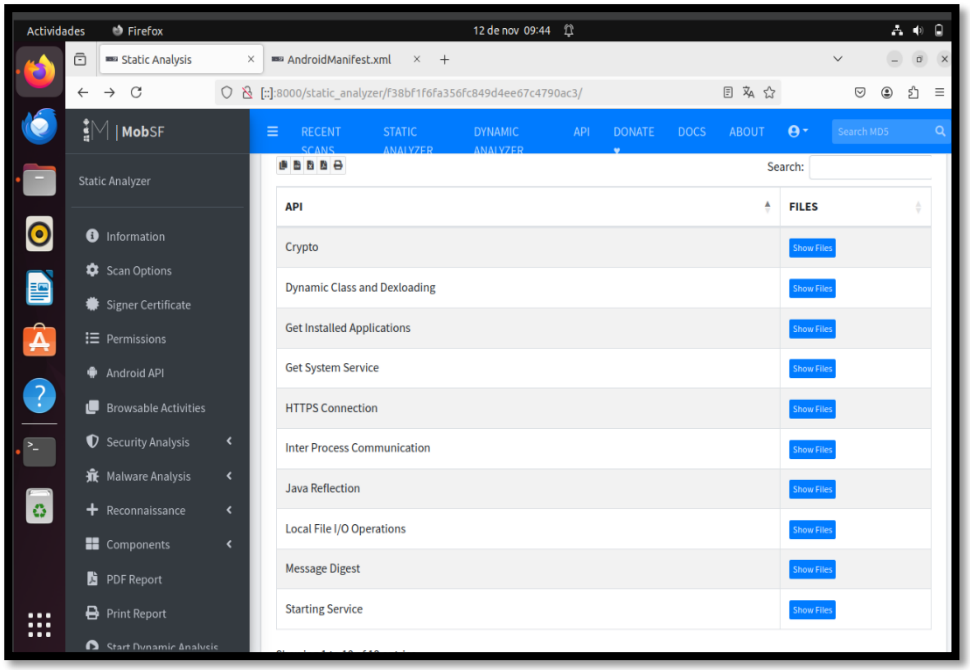


Imagen 46: Información de las API´s que Prueba.apk contiene

11. Se observa los browsable de la aplicación que permite identificar aquellas actividades que están configuradas para ser abiertas desde otras aplicaciones.

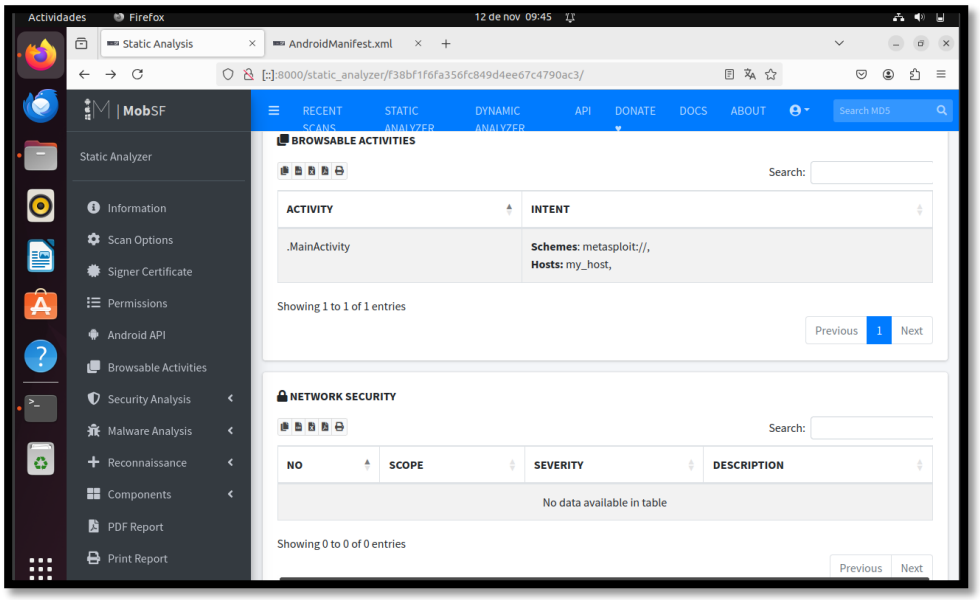


Imagen 47: Actividades configuradas y acciones secundarias

12. Certificate Analysis proporciona detalle sobre el certificado usado para firmar la aplicación Android.

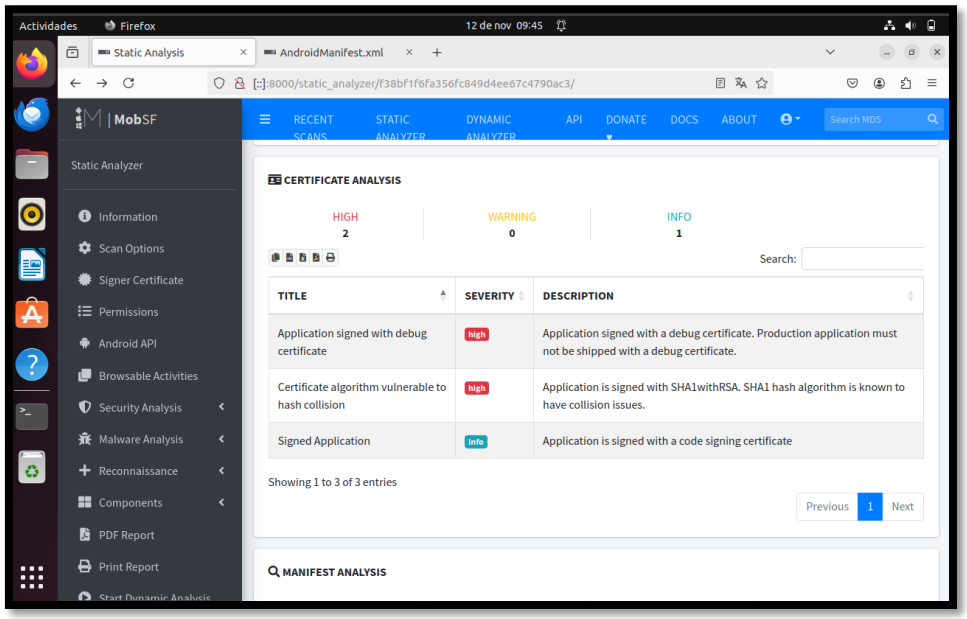


Imagen 48: Detalle del certificado de firma de Prueba.apk

13. Comprende en la revisión meticulosa del archivo AndroidManifest.xml, el archivo es esencial para definir estructura, como permisos, componentes, configuraciones de seguridad.

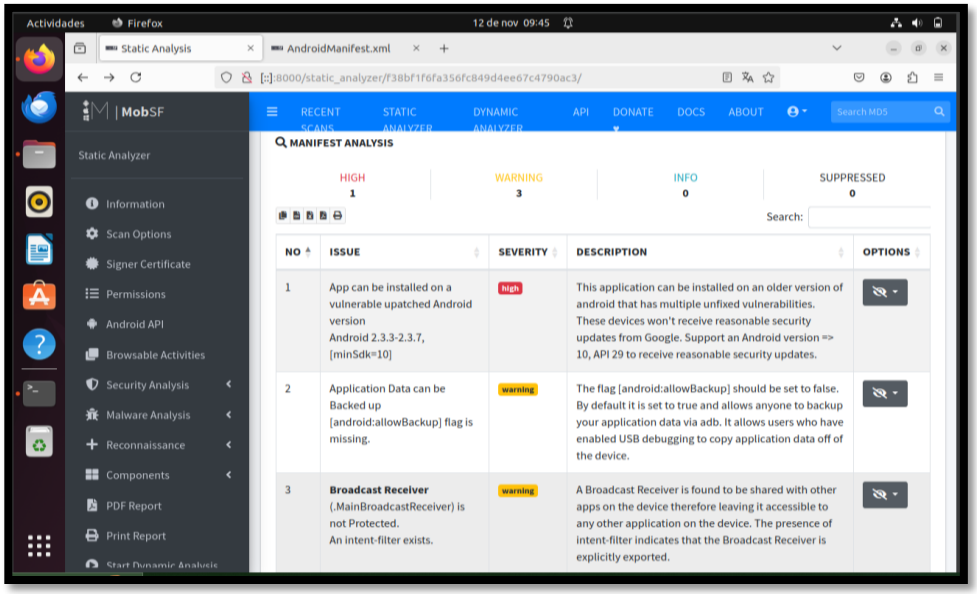


Imagen 49: Detalle a precisión del archivo AndroidManifest.xml

14. En la sección de análisis de seguridad del código o análisis de código estático. Permite identificar debilidades y futuras vulnerabilidades en el código de la aplicación Android.

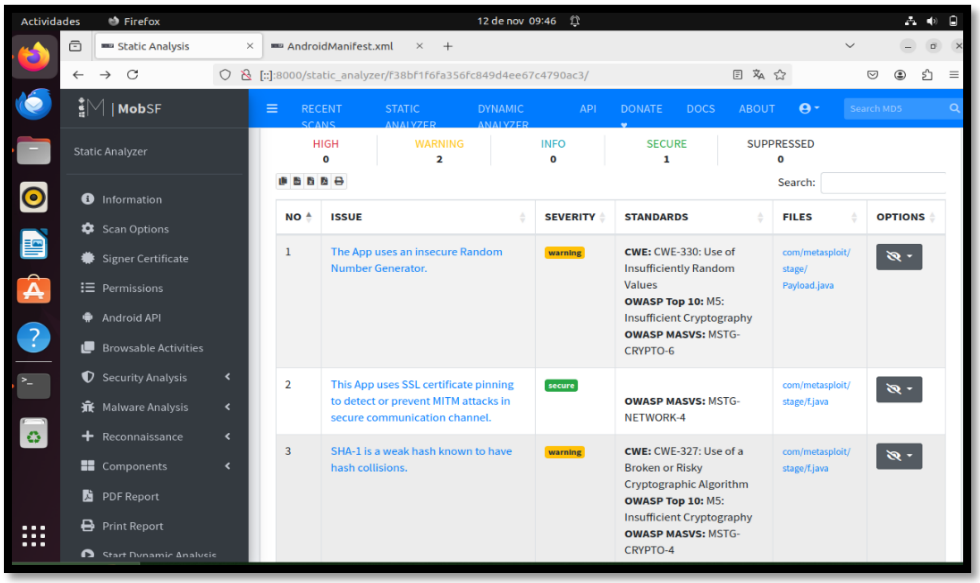


Imagen 50: Identificación de vulnerabilidades de Prueba.apk

15. En la interfaz de la herramienta MobSF hay la opción de Print Report y dar en opción guardar en la ruta que se deseé almacenar el reporte.

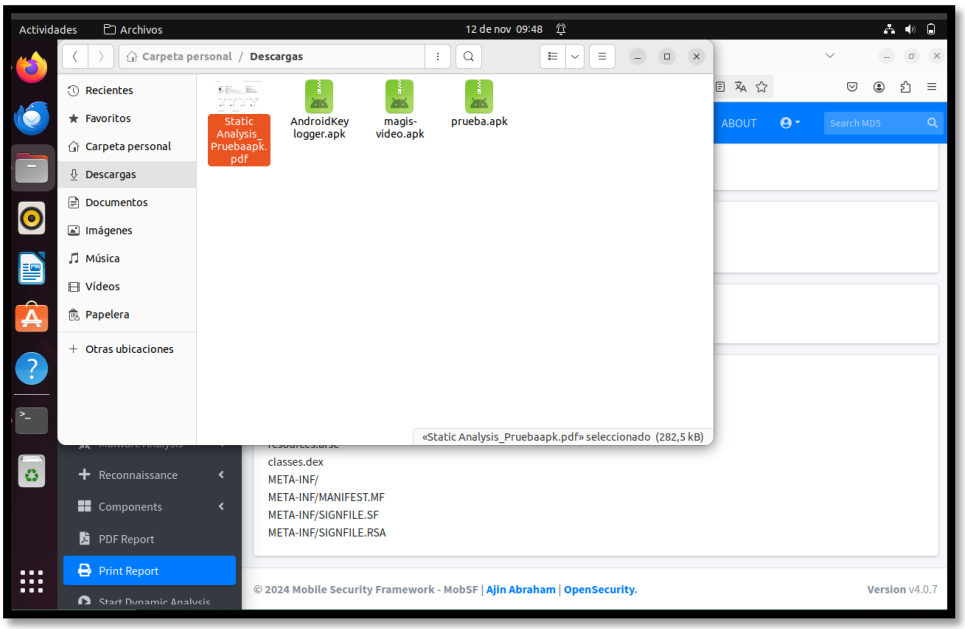


Imagen 51: Descarga del reporte de Análisis Estático de Prueba.apk

RESULTADOS ANALISIS DE APK MAGISTV

16. Análisis estático de la aplicación magistv, se selcciona la aplicación y dar en abrir

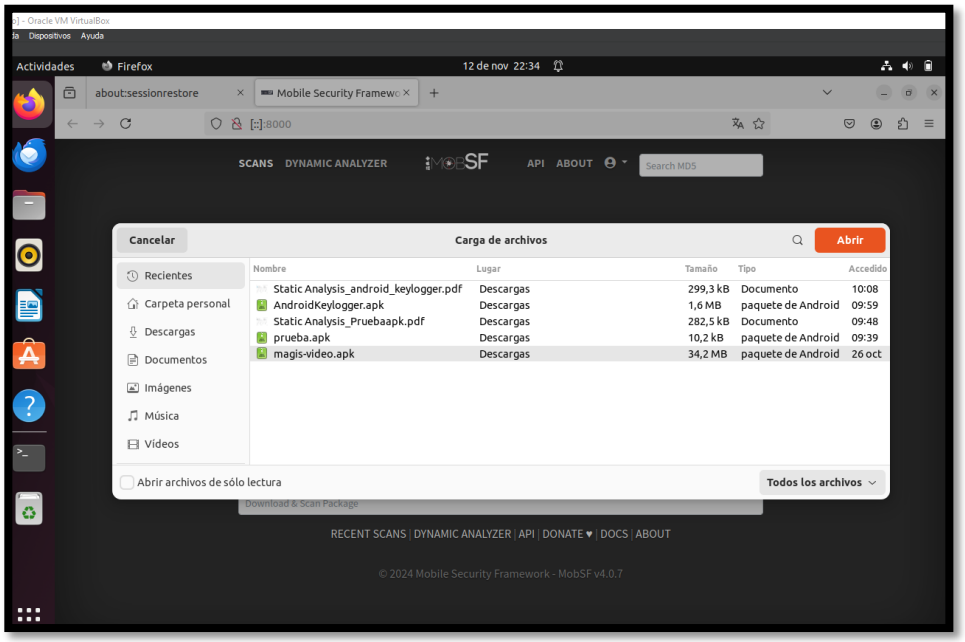


Imagen 52: Análisis Estático de Magistv

17. Se procede el desarrollo del análisis a través de la herramienta y se toma su tiempo para analizar la aplicación.

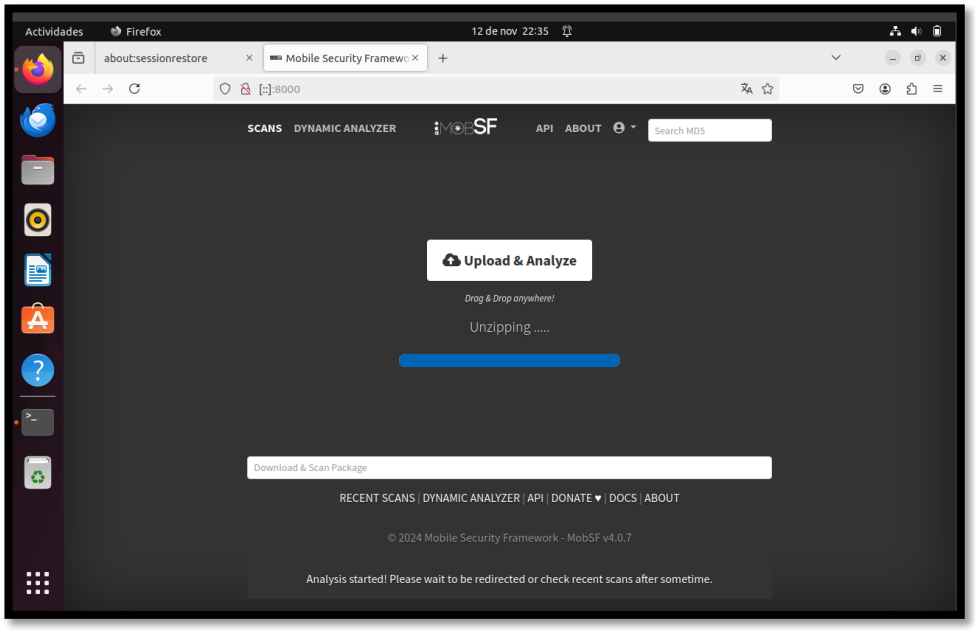


Imagen 53: Upload de la aplicación Magistv para el análisis

18. Al finalizar el análisis de la aplicación se observa la información general de la APK como la visualización del logo, la encriptación, el nombre, el tipo de formato, entre otros.

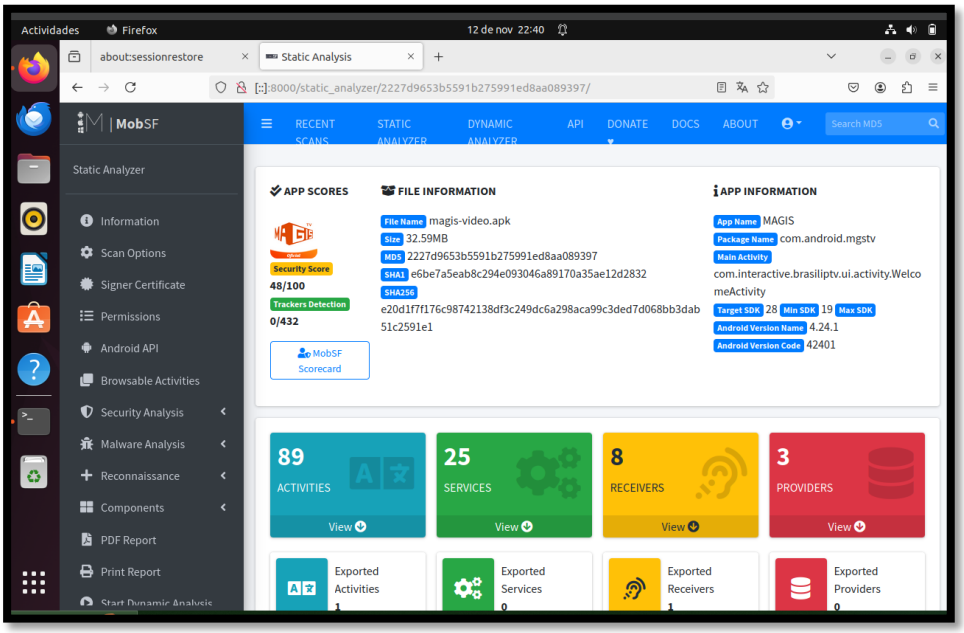


Imagen 54: El info package de la aplicación Magistv

19. Al observar el archivo AndroidManifest, se nota como el usuario debe aceptar muchos permisos en la cual pueden ser expuesto por la aplicación y recuperar información sensible, como internte, wifi, media, entre otros.

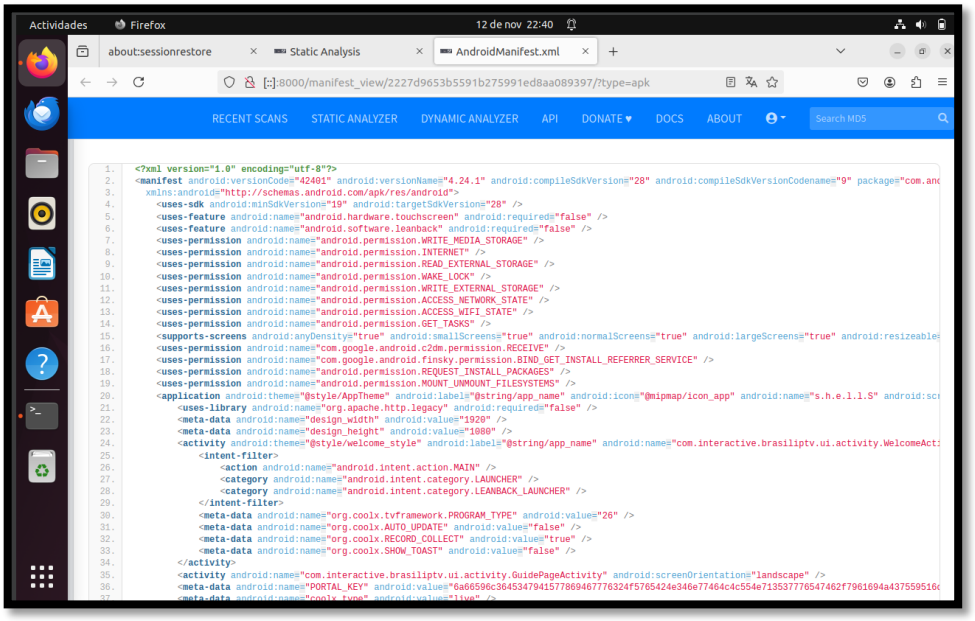


Imagen 55: Información detallada del AndroidManifest.xml de Magistv

20. Se observa la distribución de los archivos smail que cumplen la acción de ensamblador, debido que representa las instrucciones del código Dalvi

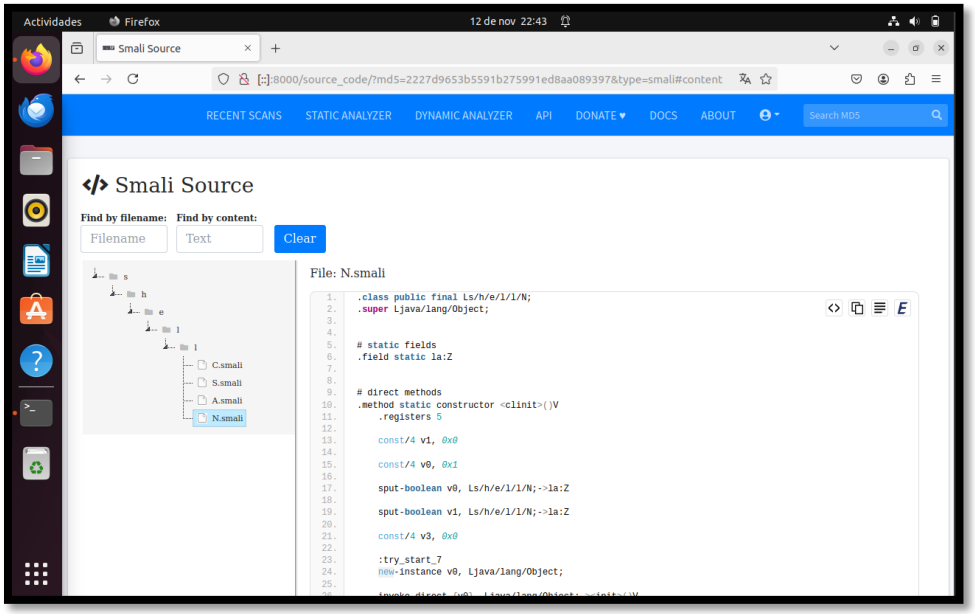


Imagen 56: Información de los archivos Smail de Magistv

21. Se observa la información sobre el certificado criptográfico utilizado como firmar para la aplicación. Es un certificado del proceso de seguridad y verificación de identidad en Android.

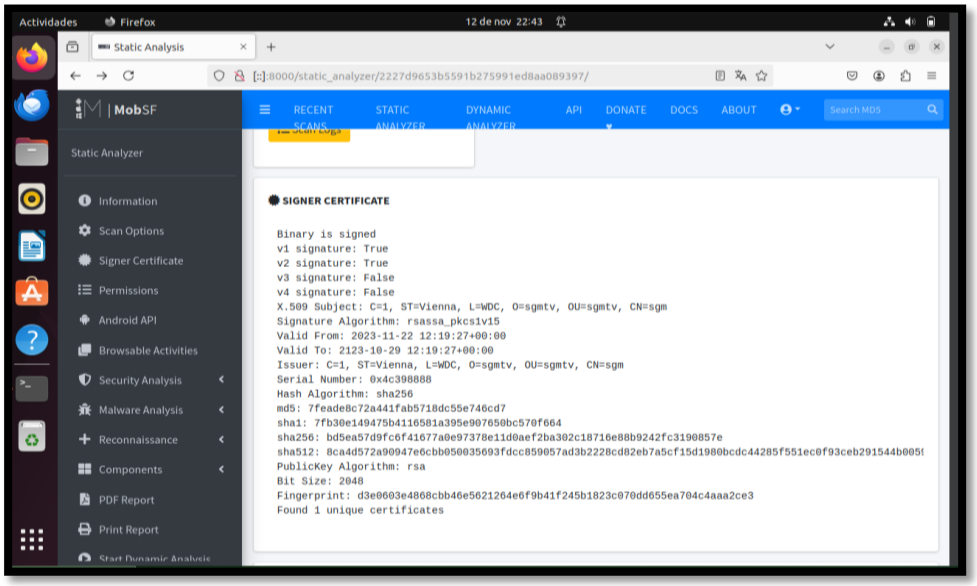


Imagen 57: Certificado de verificación de Magistv

22. En la aplicación se observar ciertos permisos que requiere para su correcto funcionamiento, pero así también el permiso de acceder a ruta nos deseadas para robar información

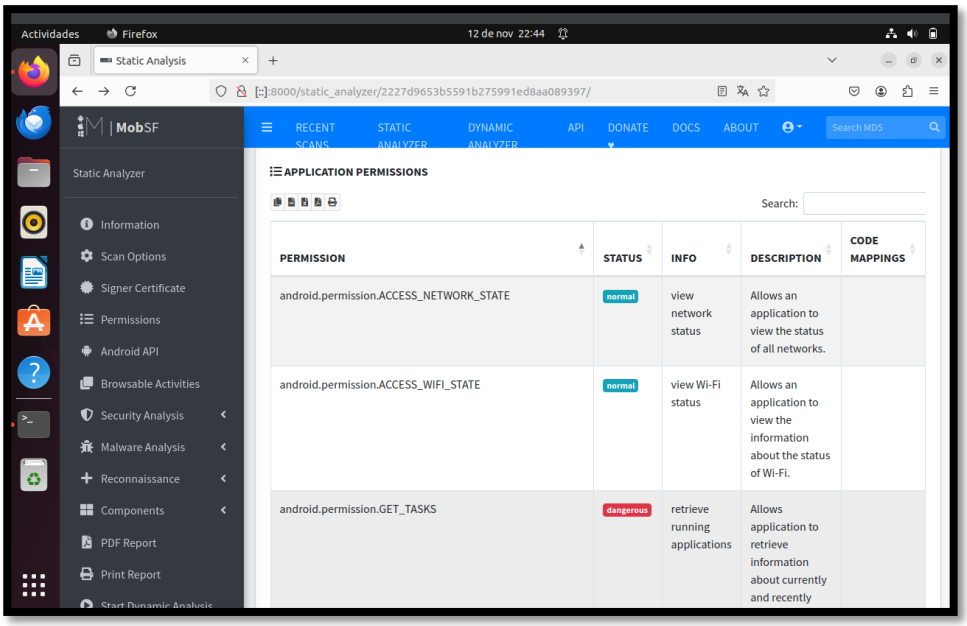


Imagen 58: Permisos que se autoriza la ejecución de magistv

23. Las API´S que la aplicación necesita para contemplar las acciones ideales para la movilidad de datos.

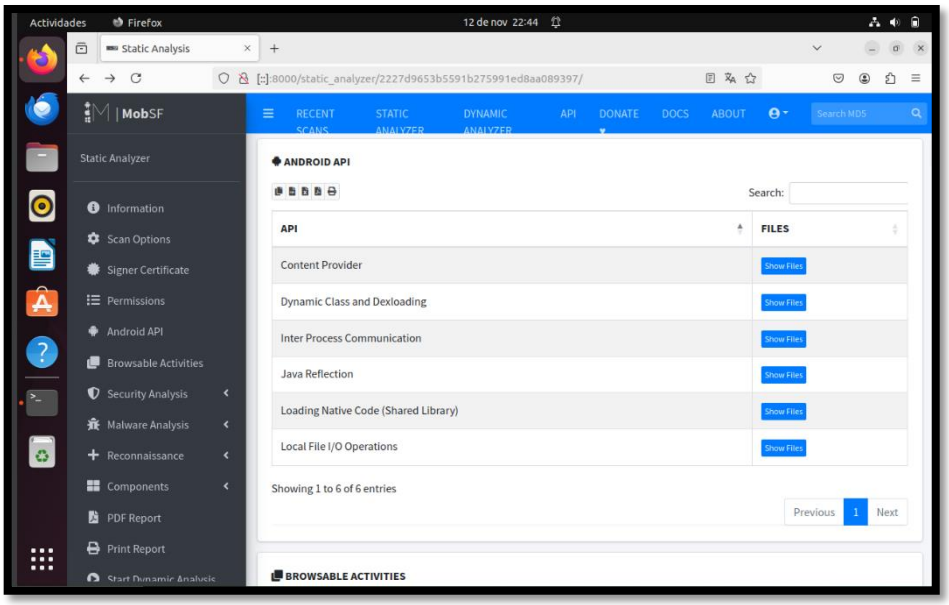


Imagen 59: La api´s que maneja magistv

24. Network Security comprende en las prácticas y configuraciones implementadas para proteger la comunicación de red de la aplicación frente a ataques y acceso no autorizados.

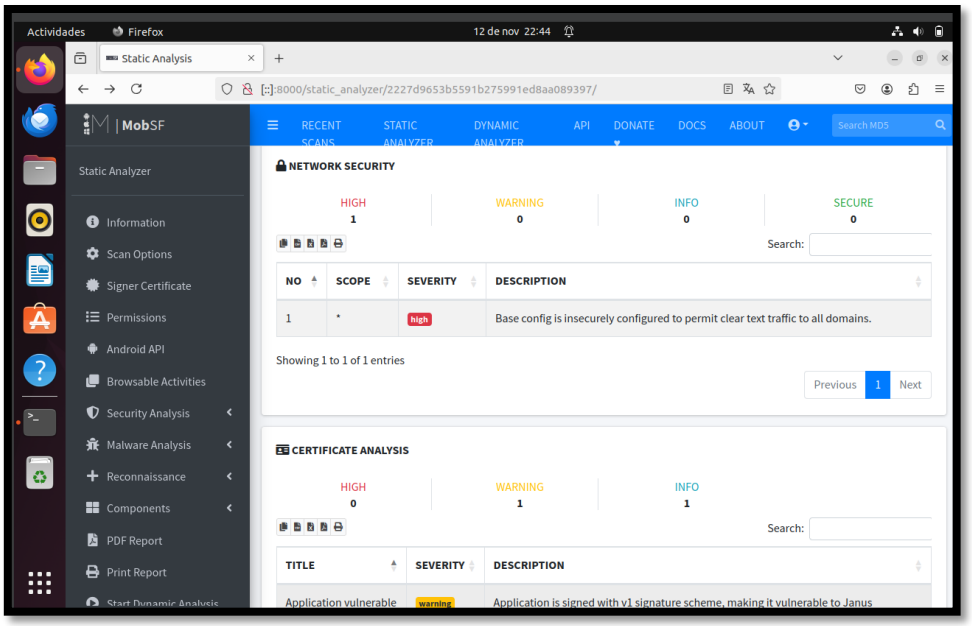


Imagen 60: El Network Security de magistv

25. El Certificate Analysis promete el análisis crucial para revisar la autenticidad, integridad y seguridad de la aplicación

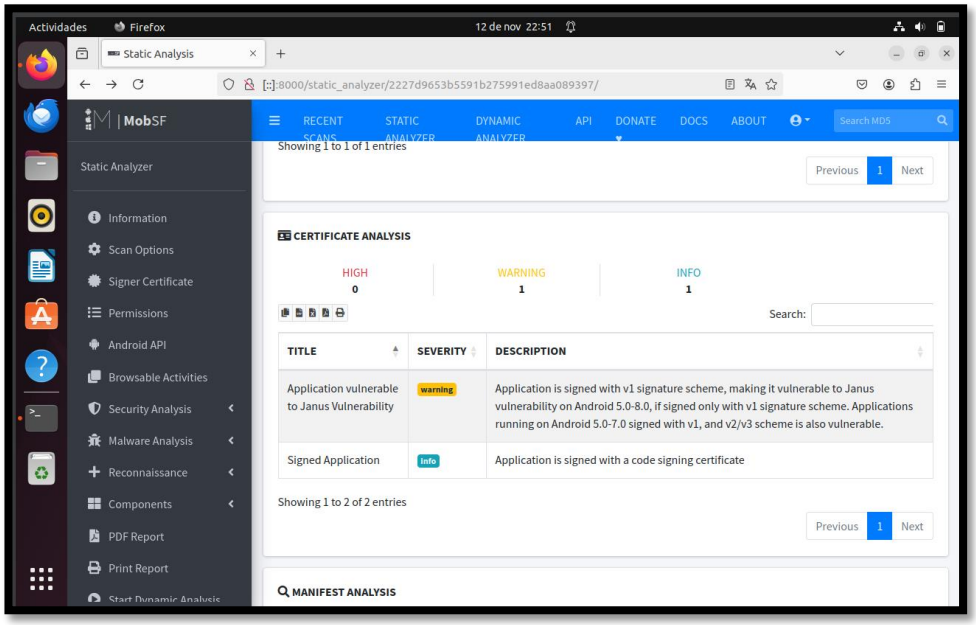


Imagen 61: La firma de certificación de magistv

26. En manifest analysis se observa todo los componentes y funciones que se encuentran sospechoso en la seguridad como los permisos, componentes, entre otros.

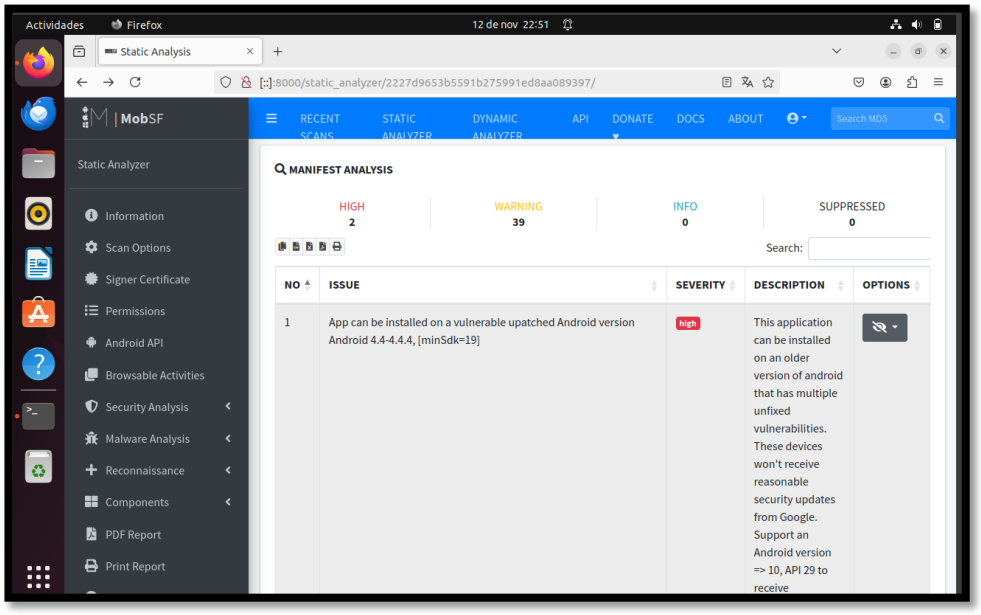


Imagen 62: Componentes de manifest Analysis de magistv

27. Shared Library Binary Analysis realiza un análisis de las bibliotecas compartidas que incluyen en una aplicación Android. El análisis es importante ya que las bibliotecas nativas pueden introducir vulnerabilidades críticas en una aplicación

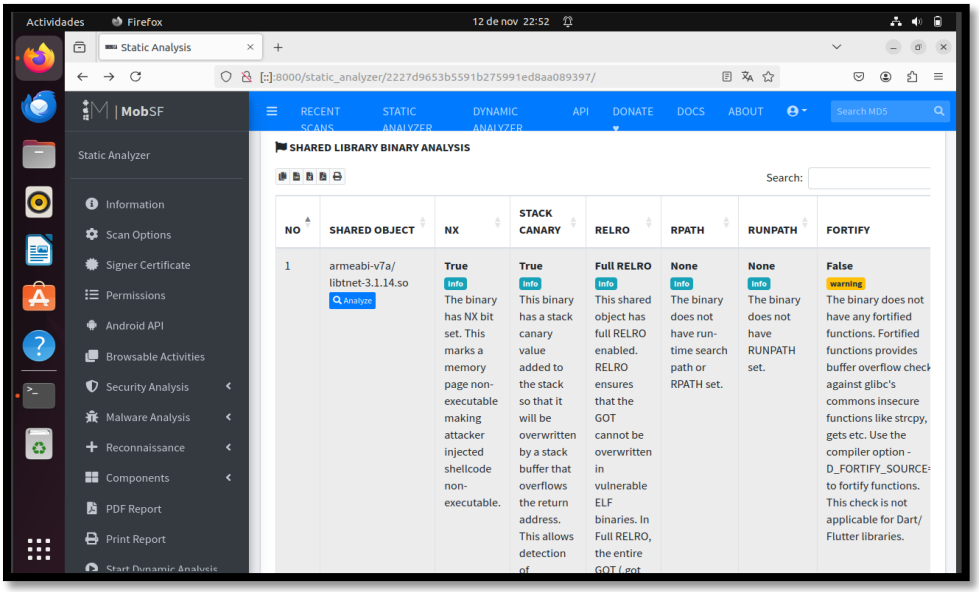


Imagen 63: Shared Library de magistv

28. APKID ANALYSIS – se observa detenidamente como la aplicación cuenta con ID en la especificación root, classes, entre otros.

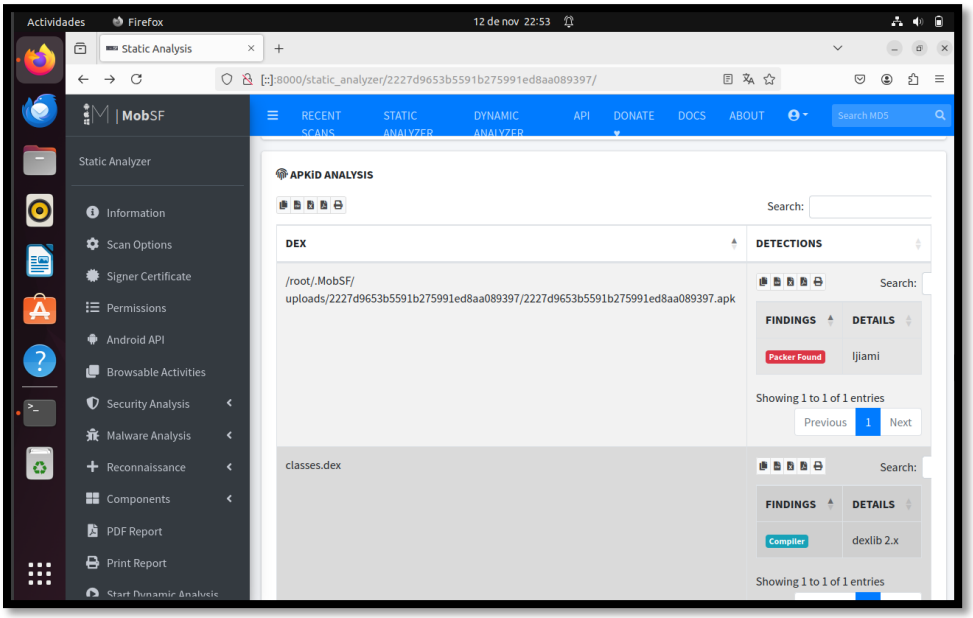


Imagen 64: ApkID Analysis de magistv

29. La descripción general de los permisos que la aplicación con otra clasificación, debido que existen grupos como “top malware permissions” y other – se refiere a los permisos comunes que otras aplicaciones requerida.

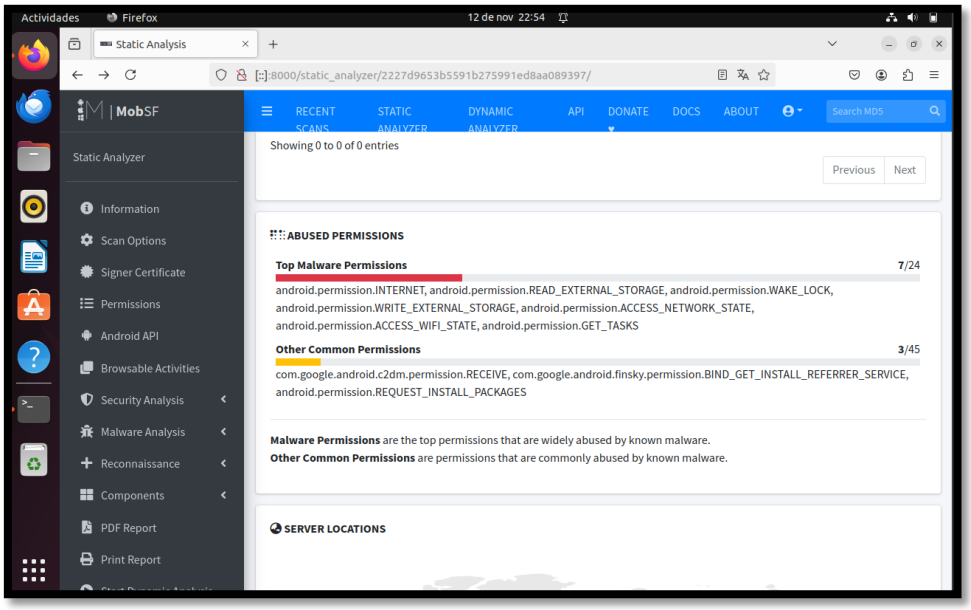


Imagen 65: Descripción general de permisos magistv

30. DOMAIN MALWARE CHECK – se observa como la herramienta cuenta con dominios de servidores de otra País y la dirección IP

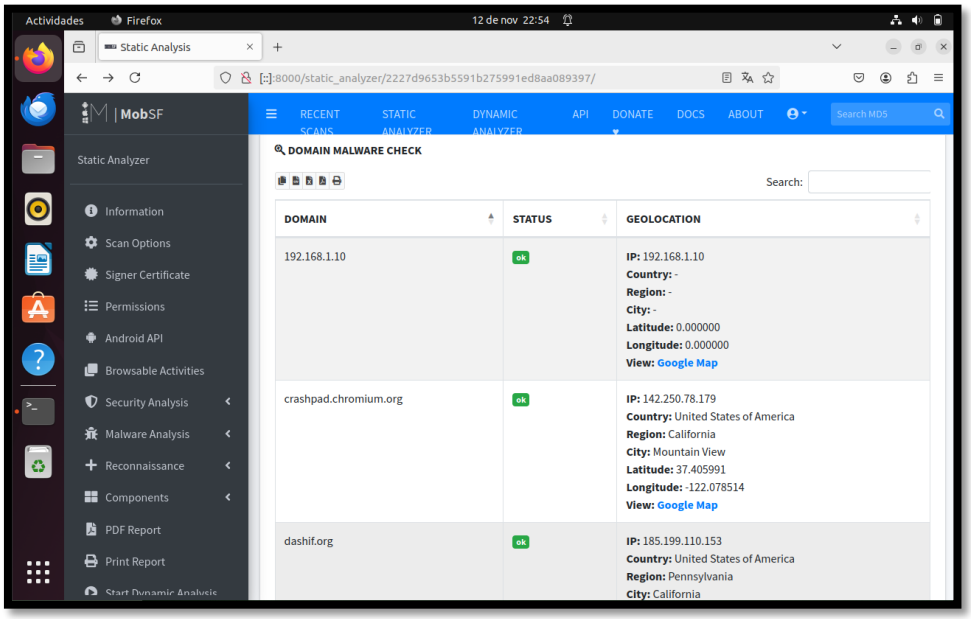


Imagen 66: Domain Malware Check de magistv

31. Se da clic en la sección Print Report se localiza la ruta en donde se almacenera el reporte del segundo malware.

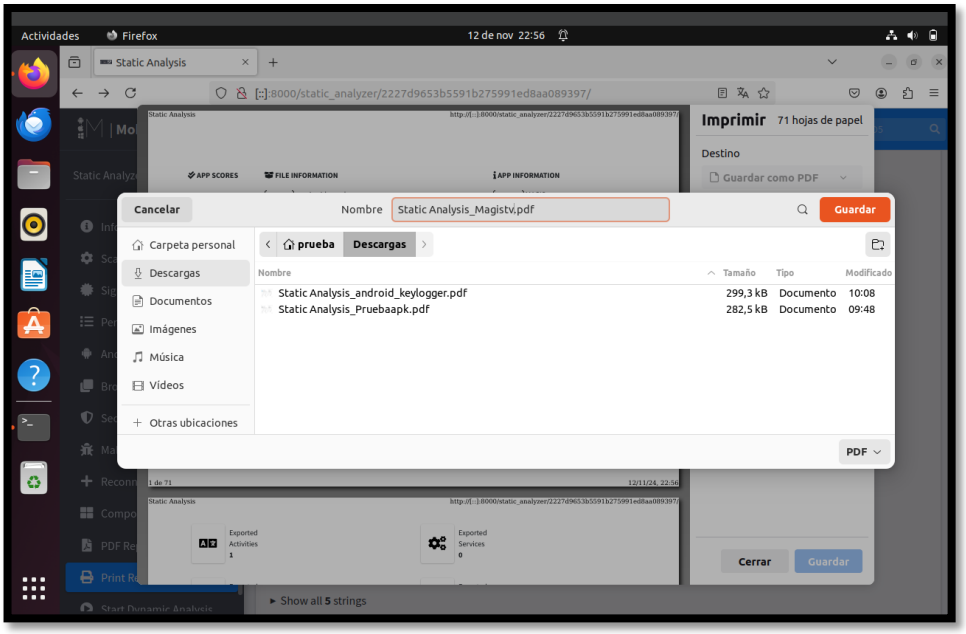


Imagen 67: Reporte de Análisis Estático magistv

RESULTADOS ANALISIS DE APK ANDROIDKEYLOGGER

32. Se selecciona la herramienta Android_Keylogger para el análisis,

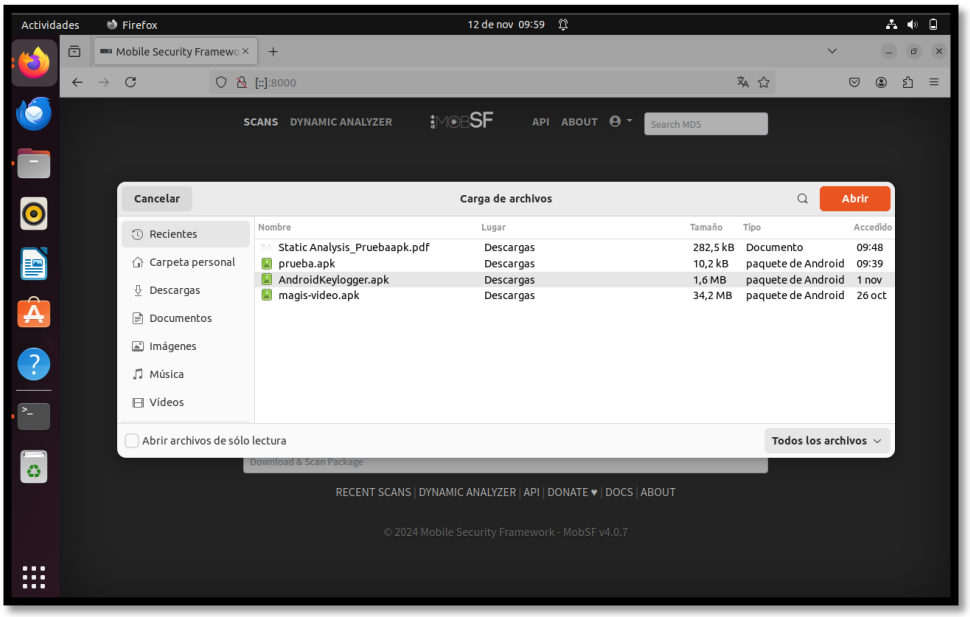


Imagen 68: Android_Keylogger para Análisis Estático

33. Una vez cargado el archivo, se procede a buscar los archivos que pertenece a la aplicación.

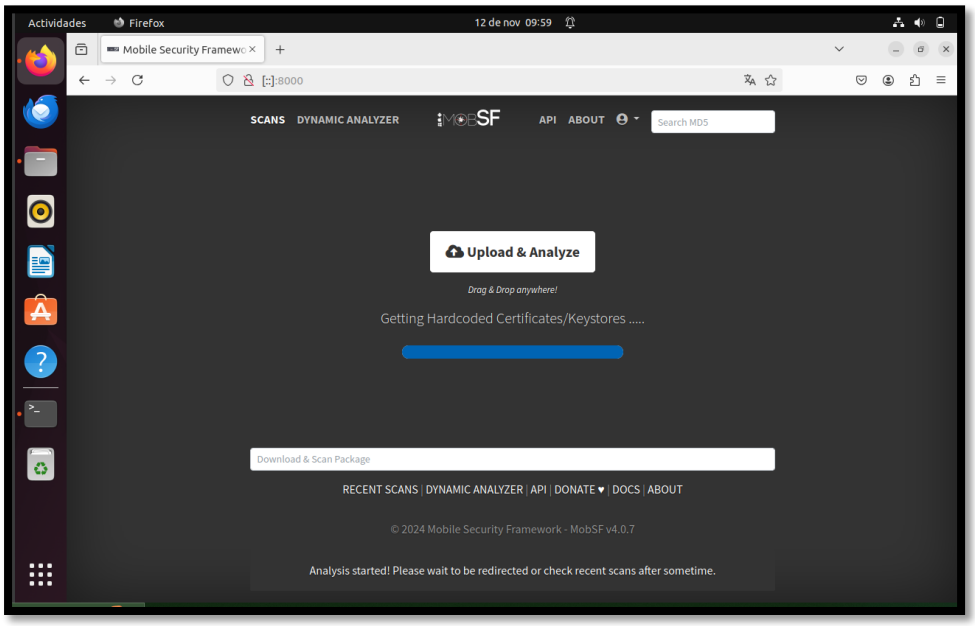


Imagen 69: Proceso de carga de Android_Keylogger

34. En el primer apartado se observa la información general de la aplicación como el icono, nombre de la aplicación, tipo, encriptación, entre otros.

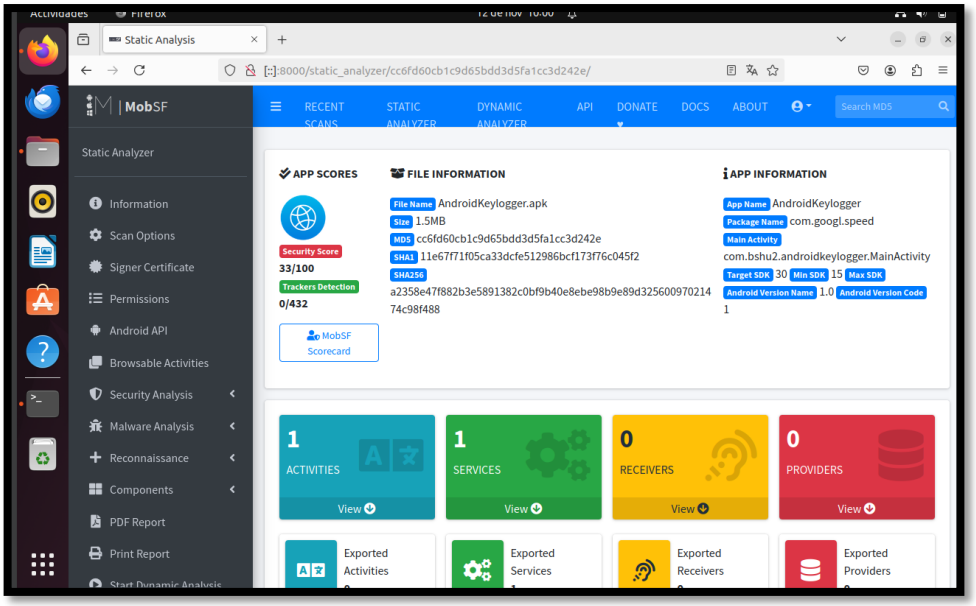


Imagen 70: Información del package de Android_Keylogger

35. En el archivo AndroidManifest se manifiesta las configuraciones de acceso de la aplicación al dispositivo Android y doblegar la seguridad para robar activos de información por los permisos que interactúa la aplicación y el dispositivo

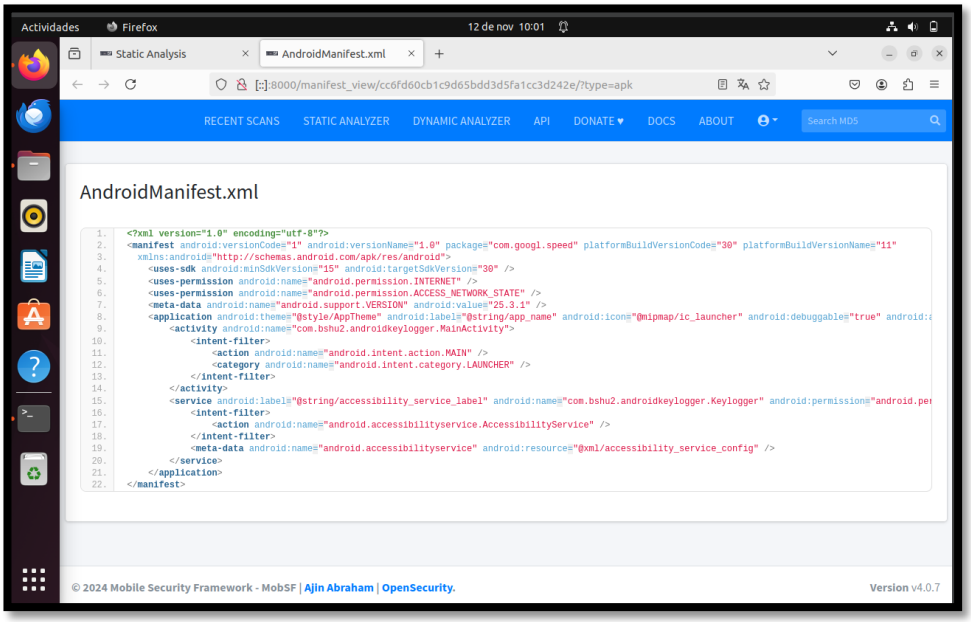


Imagen 71: Análisis de AndroidManifest.xml de Android_Keylogger

36. Aquí se observa los archivos que cumple la solicita de un ensamblador para que acepte las instrucciones del código Dalvik

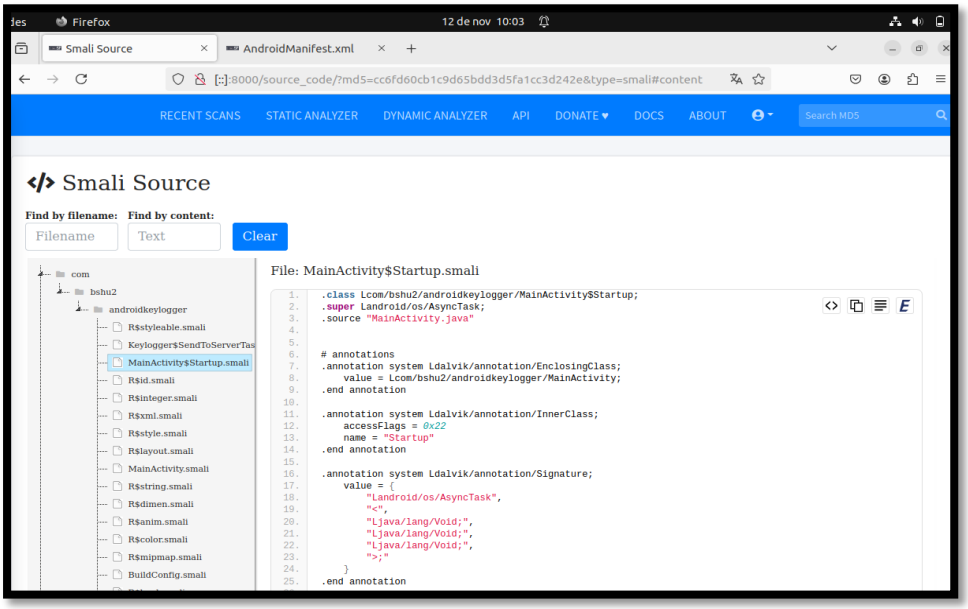


Imagen 72: Archivo Smail de Android_Keylogger

37. Aquí se observa el certificado de firma de la aplicación, es proceso esencial para la seguridad y la verificación de identidad de la aplicación Android.

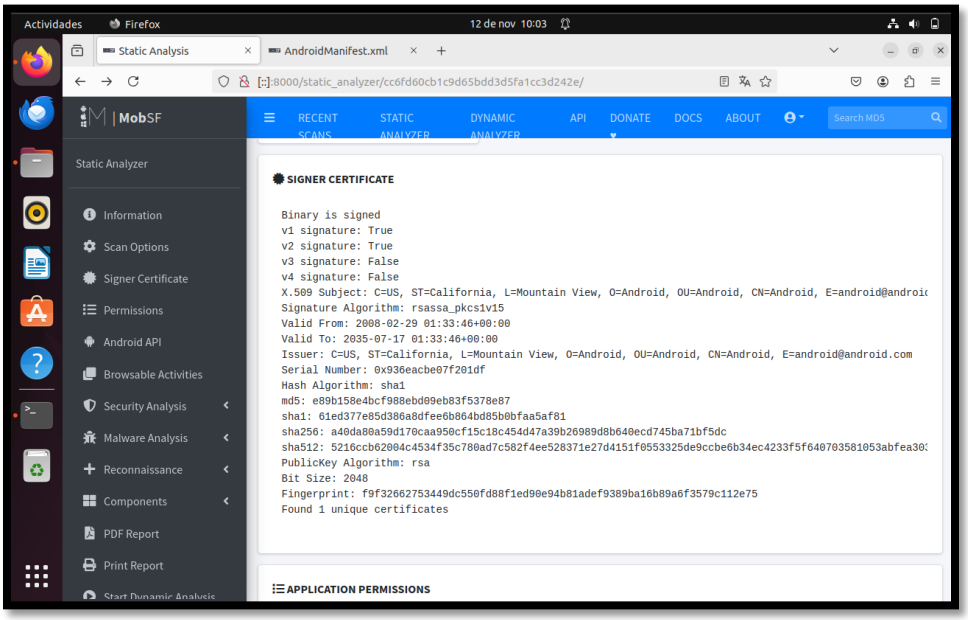


Imagen 73: Certificado de firma de Android_Keylogger

38. La lista general de los permisos que se deben aceptar para la ejecución del aplicativo Android

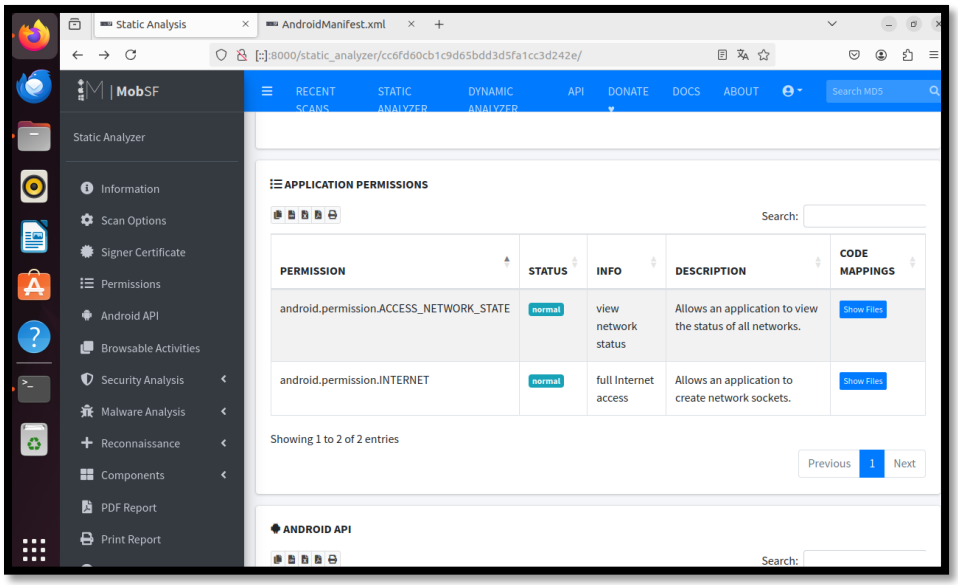


Imagen 74: Lista general de permisos de Android_Keylogger

39. Las API´S de la aplicación Android

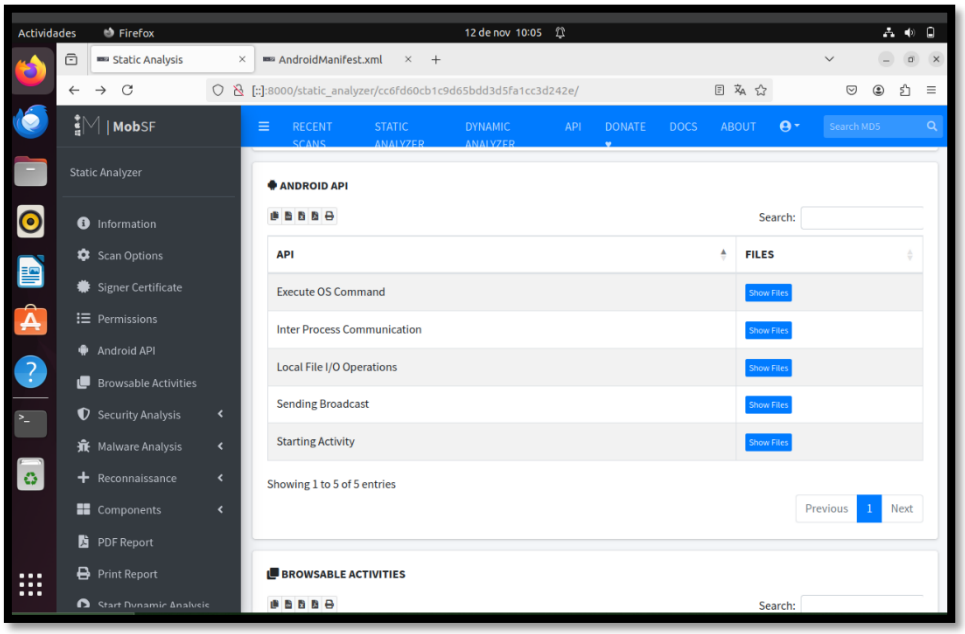


Imagen 75: Api´s de Android_Keylogger

40. Aquí se encuentra el certificado de análisis de la aplicación con un severity alta y media.

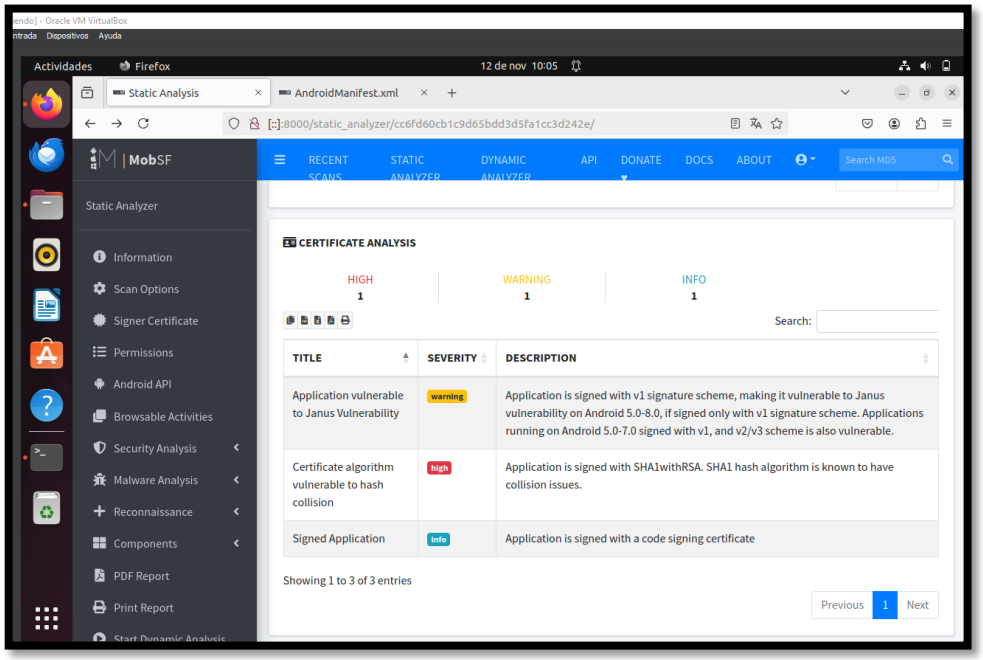


Imagen 76: Certificado de Análisis de Android_Keylogger

41. En este apartado se presenta la definición de la estructura y el comportamiento de la aplicación, se manifiestan los permisos, componentes, configuraciones de seguridad.

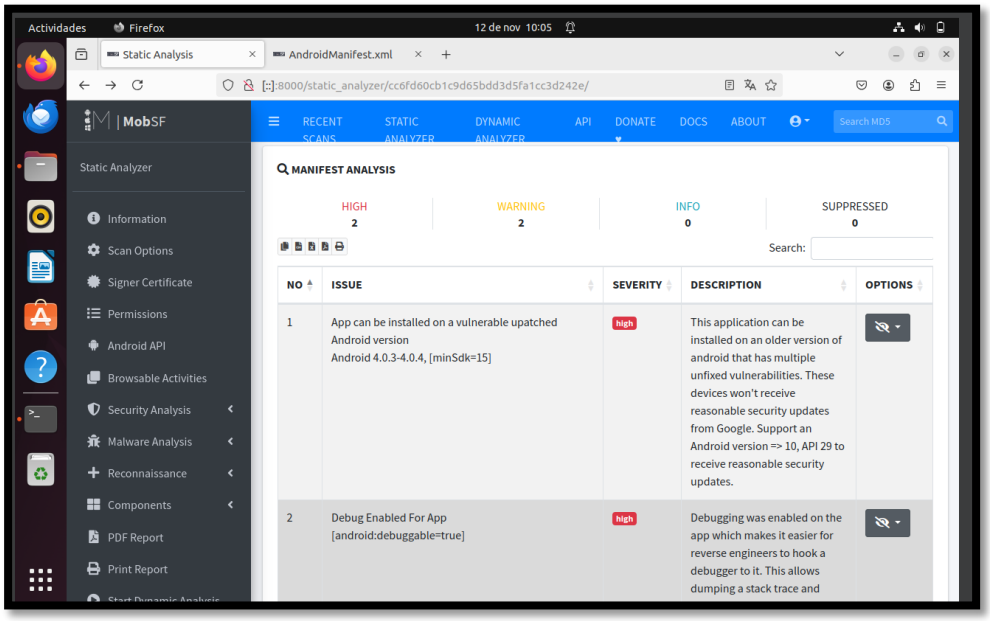


Imagen 77: Manifest Analysis de Android_Keylogger

42. Aquí en Code Analysis se presenta el análisis de seguridad del código fuente con el objetivo de identificar debilidades y posibles vulnerabilidades en el código de la aplicación de Android.

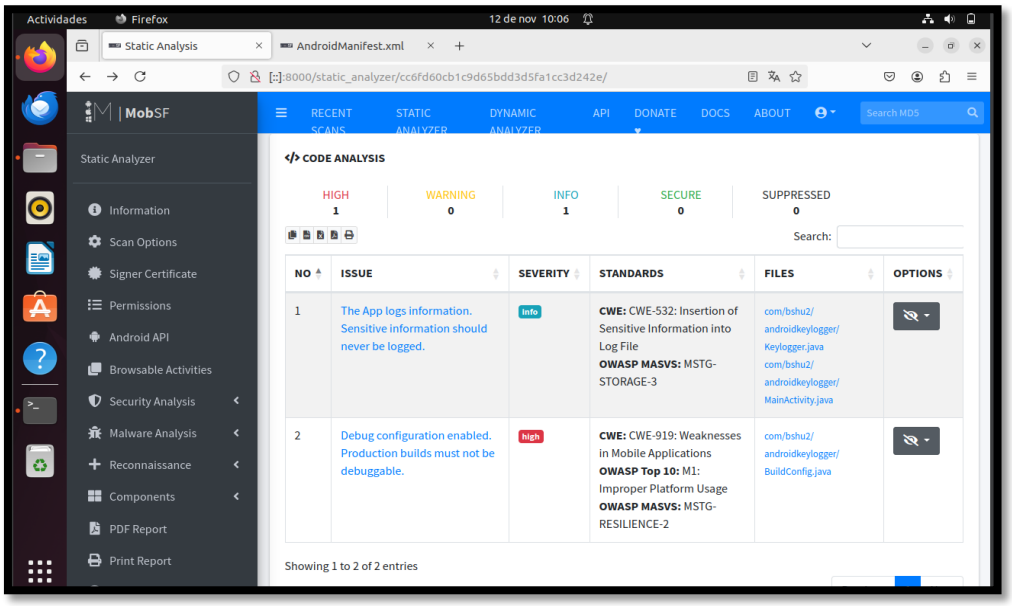


Imagen 78: Code Analysis de Android_Keylogger

43. Aquí se manifiesta el dominio que se encuentra asociado la aplicación, se refiera a la conectividad del servidor

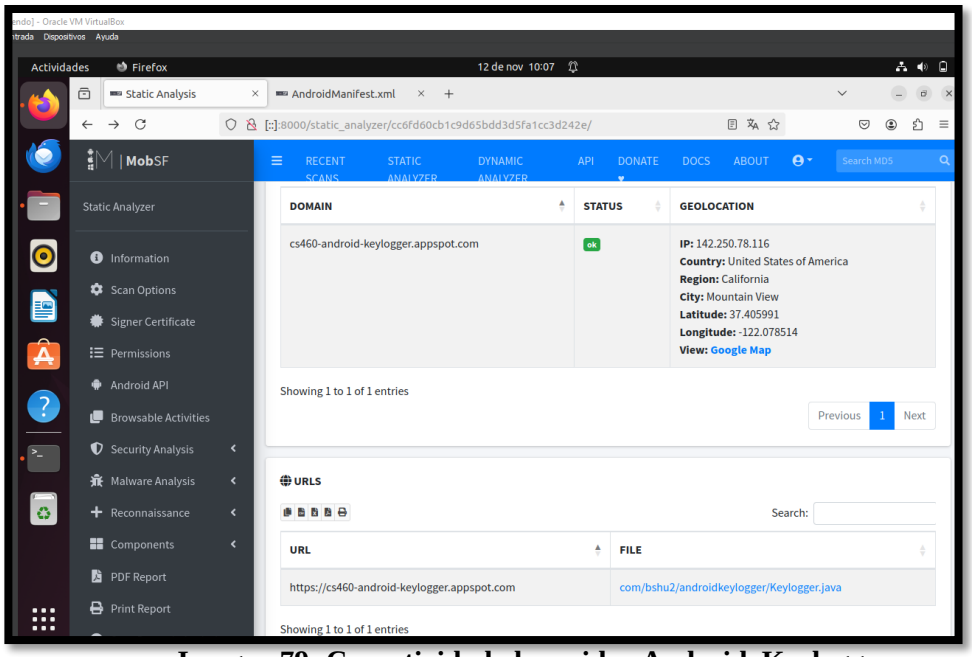


Imagen 79: Conectividad al servidor Android_Keylogger

44. Se ejerce el almacenamiento del archivo en la ruta deseada como reporte general para más visualización

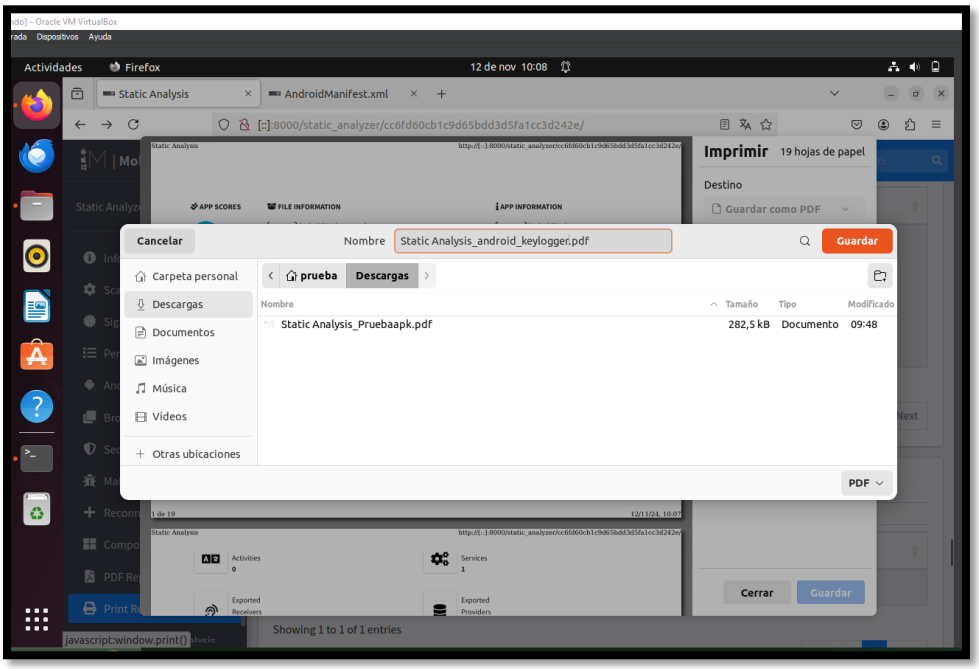


Imagen 80: Guardar el reporte de análisis estático de Android_Keylogger

ANÁLISIS DINÁMICO – MALWARE´S

MALWARE – PRUEBA.APK

45. Una vez instalado la aplicación prueba.apk en el dispositivo desde la console de drozer ejecutar el comando “run app.package.list –f nombre_paquete” en este caso es metasploit el virus “MainActivity”

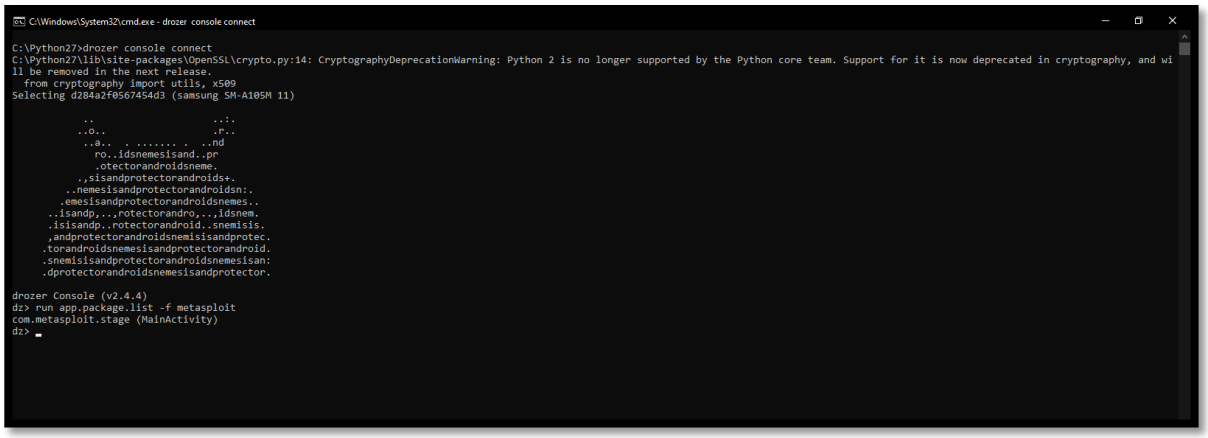


Imagen 81: Buscar el package de prueba.apk – Dinámico

46. Una vez hallado el package, con el comando “run app.package.info –a com.metasploit.stage” se mostrara la información detallada sobre el paquete de la aplicación.

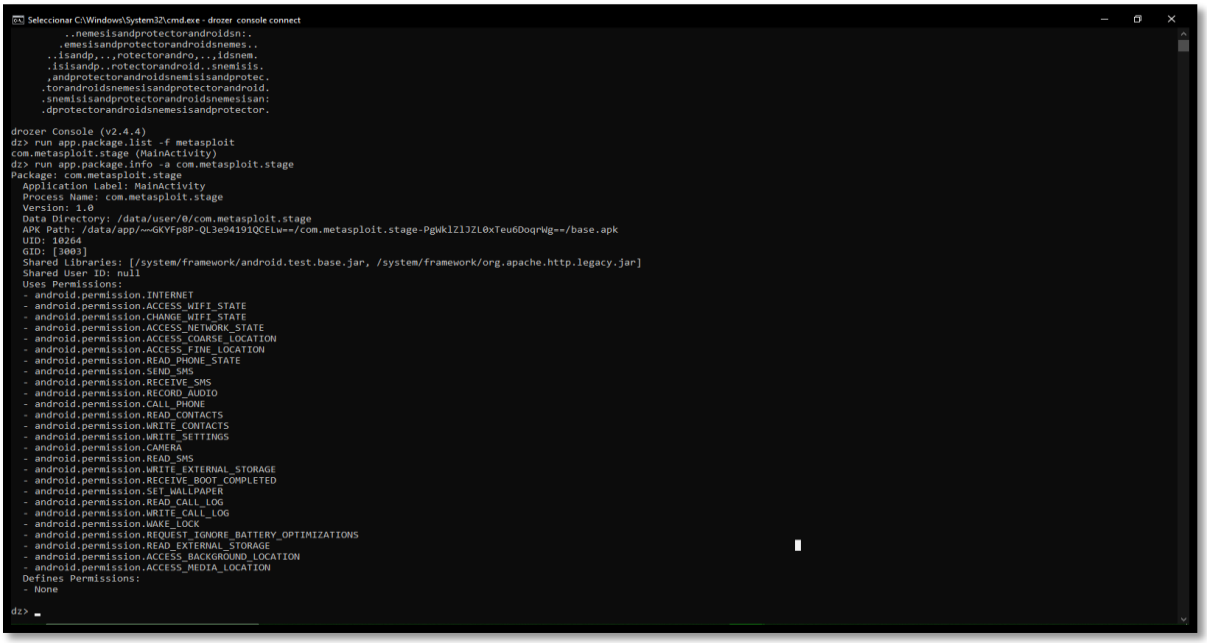


Imagen 82: Información del package prueba.apk – Dinámico

47. Con el siguiente comando “run app.package.attacksurface com.metasploit.stage”, sirve para inspeccionar todos los componentes exportados (Actividades, servicios, receptores, entre otros) de la aplicación para identificar posibles puntos de entradas.

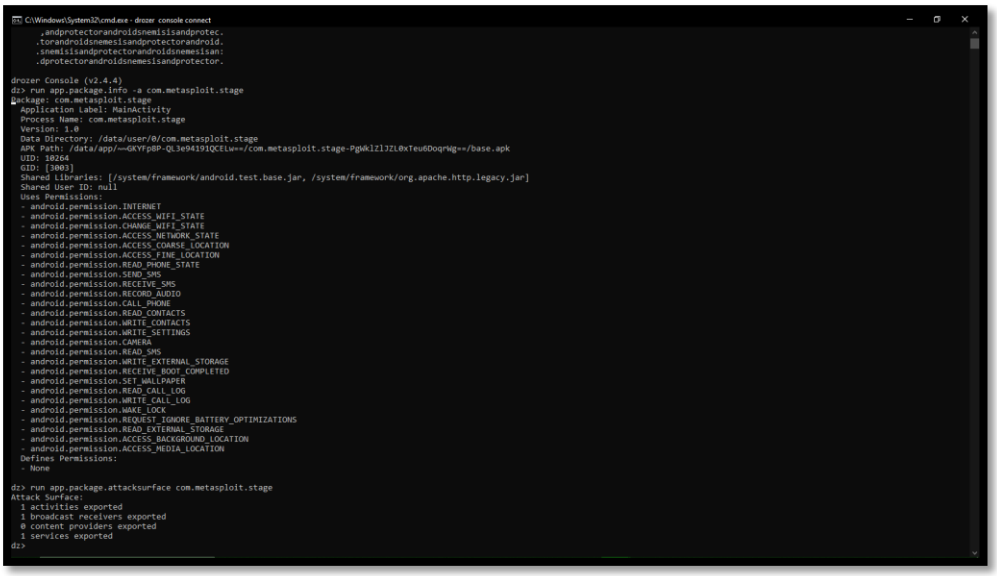


Imagen 83: Inspeccionar los componentes de prueba.apk – Dinámico

48. Con el comando “run app.activity.info -a com.metasploit.stage”, sirve para exponer las actividades que pueden ser vulnerables si no están protegidas

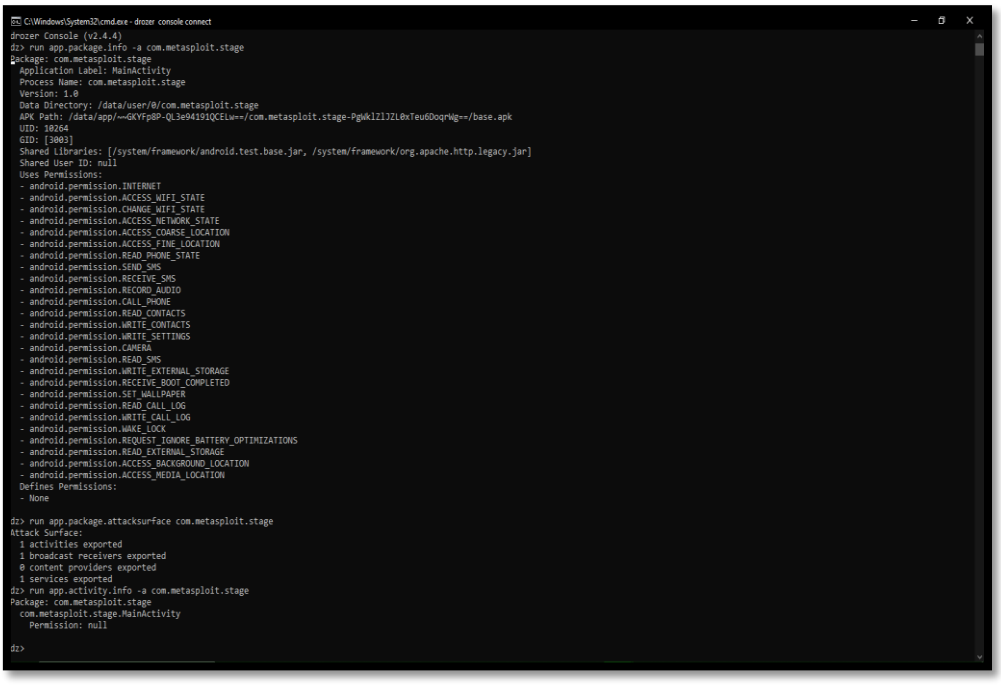


Imagen 84: Exponer las actividades de prueba.apk – Dinámico

49. Con el comando “run app.activity.start --component com.metasploit.stage com.metasploit.stage.MainActivity”, sirve para interactuar con la actividad de la aplicación y se observa que permite activa todo los permisos al abrir la apk

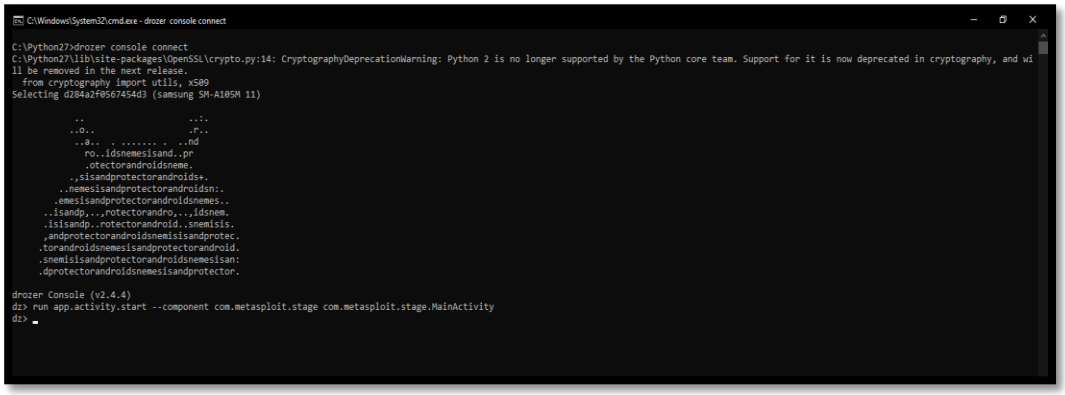


Imagen 85: Interactuar con prueba.apk – Dinámico

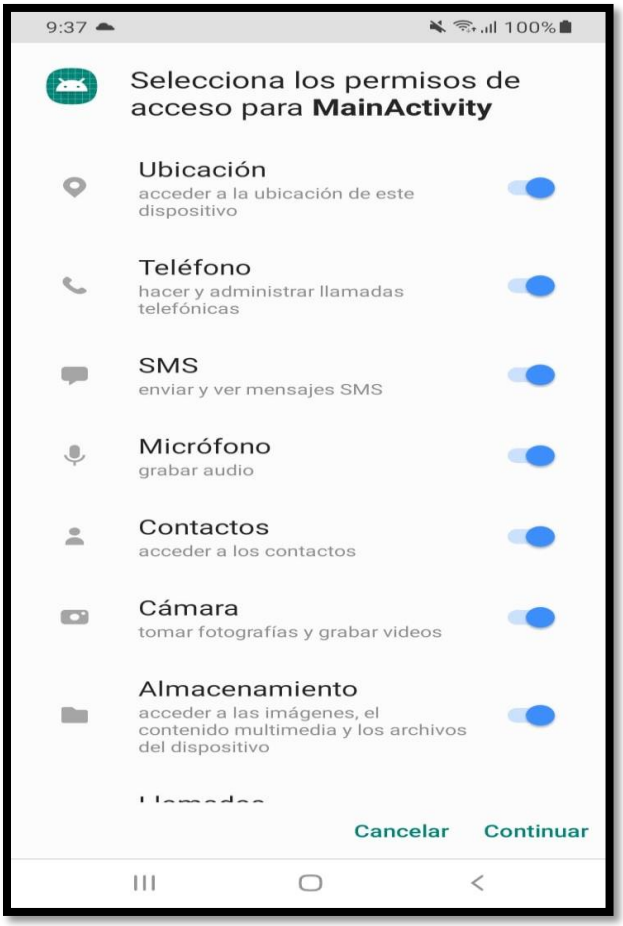


Imagen 86: Inicio de aplicación Prueba.apk – Dinámico

MALWARE – MAGISTV.APK

50. Comando “run app.package.list –f magis” sirve para encontrar el paquete de la aplicación analizar.

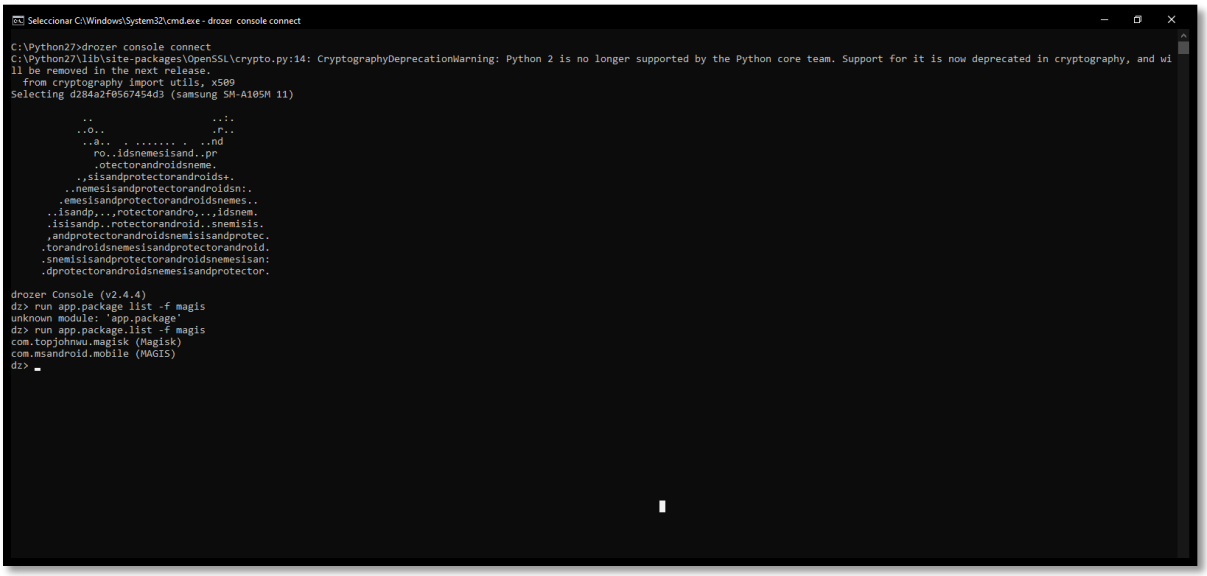


Imagen 87: Buscar el package de Magistv - Dinámico

51. Con el comando “run app.package.info –a com.msandroid.mobile”, sirve para mostrar la información de la aplicación y presentar los permisos que administra.

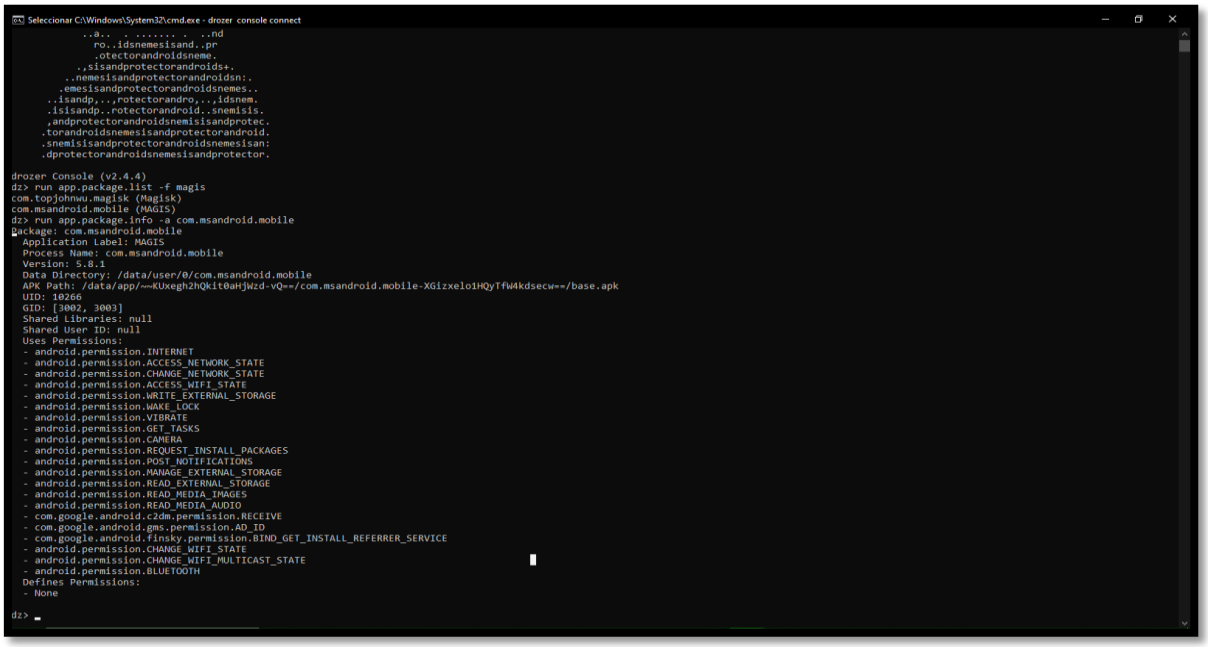


Imagen 88: Información del package de Magistv – Dinámico

52. Con el comando “run app.package.attacksurface” com.msandroid.mobile” sirve para conocer las actividades que presenta el package de las aplicaciones, broadcast y servicios exportados.

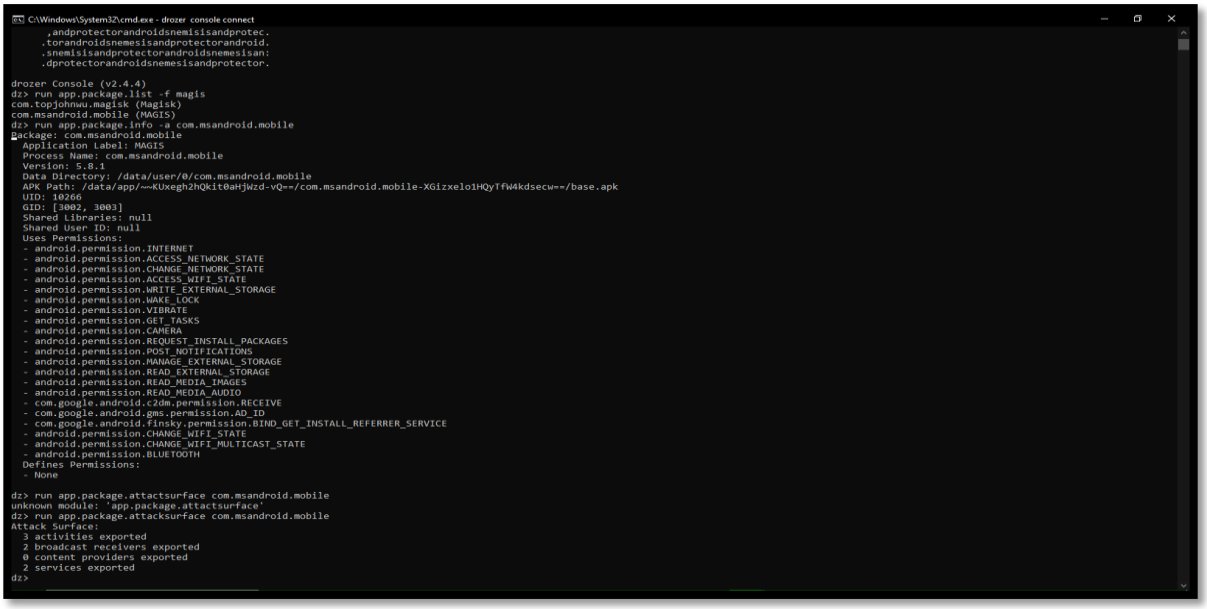


Imagen 89: Listar las actividades del package Magistv – Dinámico

53. Con el comando “run app.activity.info –a com.msandroid.mobile” sirve para listar todos los servicios que cuenta el package de la aplicación

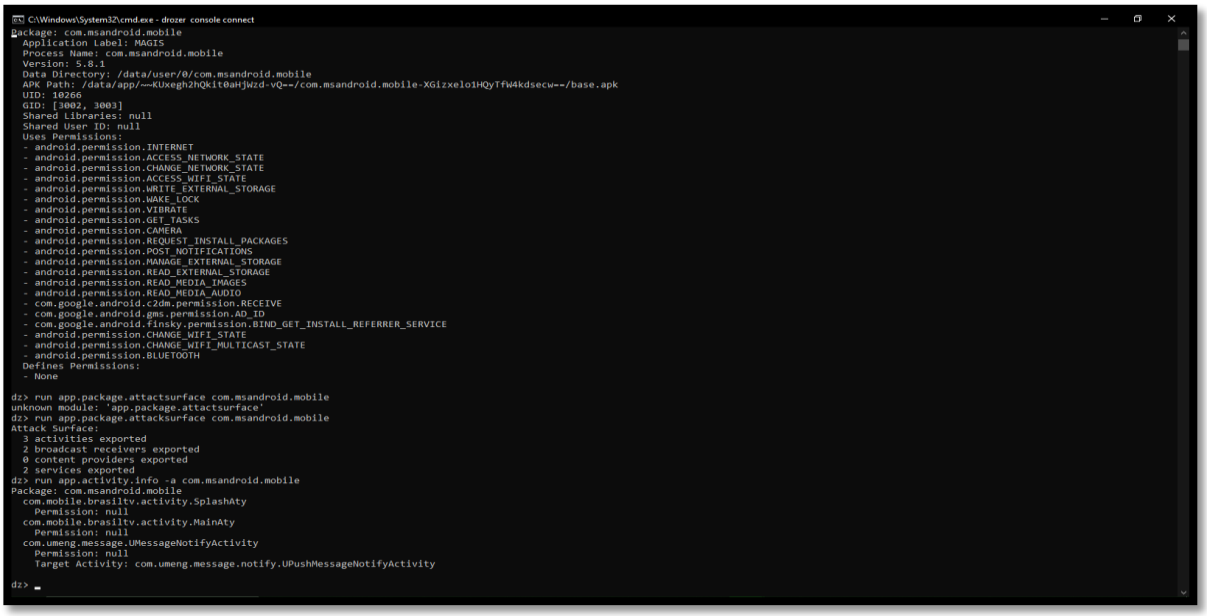


Imagen 90: Lista de servicios del package Magistv – Dinámico

54. Con el comando “run app.activity.strart –component com.msandroid.mobile com.mobile.brasiltv.activity.SplashAty” sirve para levantar servicio al dispositivo y ese corresponde a la pantalla principal para aceptar los permisos.

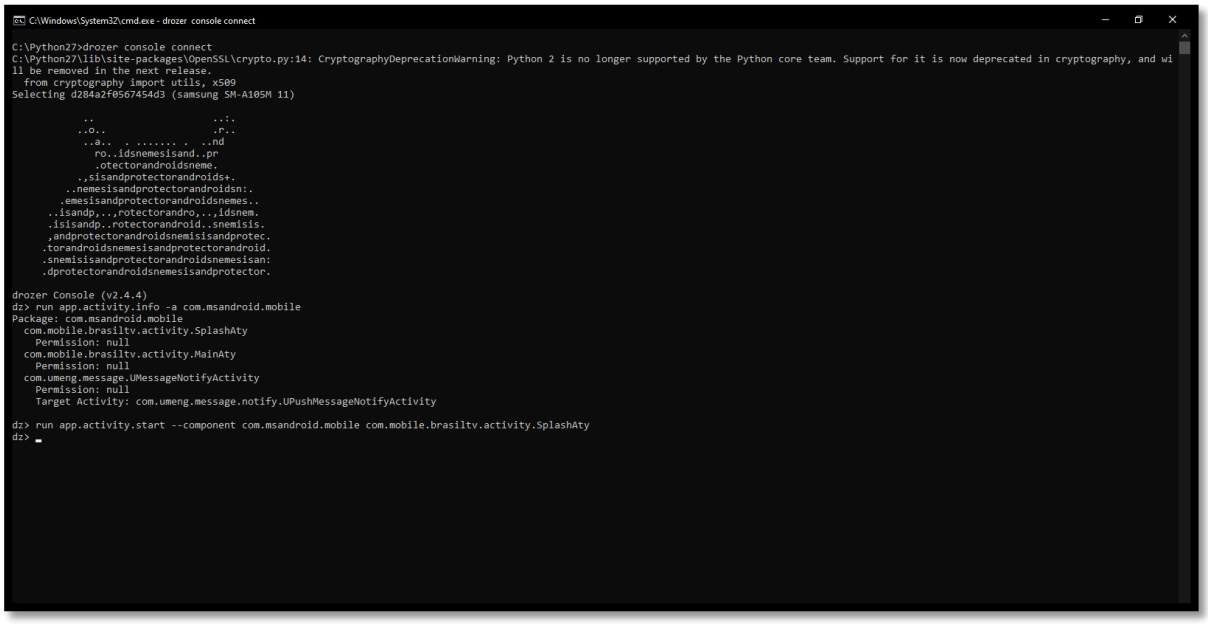


Imagen 91: Levantar los servicios de package Magistv – Dinámico

55. De misma forma se ejecutan los demás servicios para presentar la interaccion con la aplicación y ver su movimiento de acción

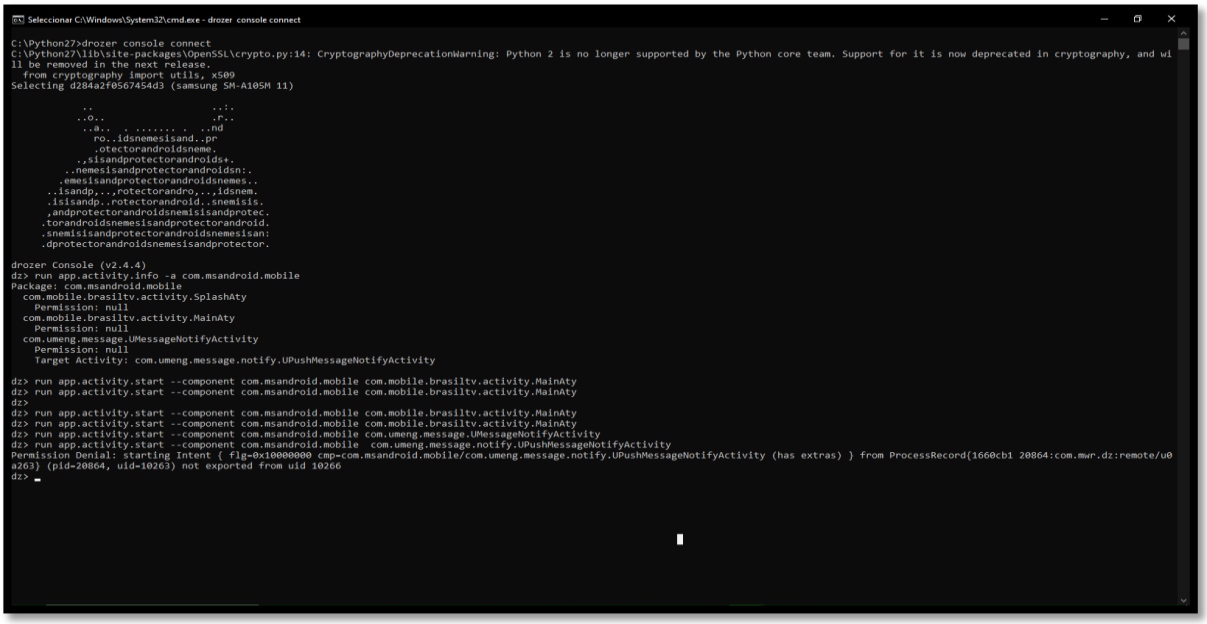


Imagen 92: Ejecución de los demás servicio Magistv – Dinámico

MALWARE – ANDROIDKEYLOGGER.APK

56. Con el comando "run app.package.list –f keylogger" sirve para efectuar la búsqueda del package de la aplicación androidKeylogger para el análisis

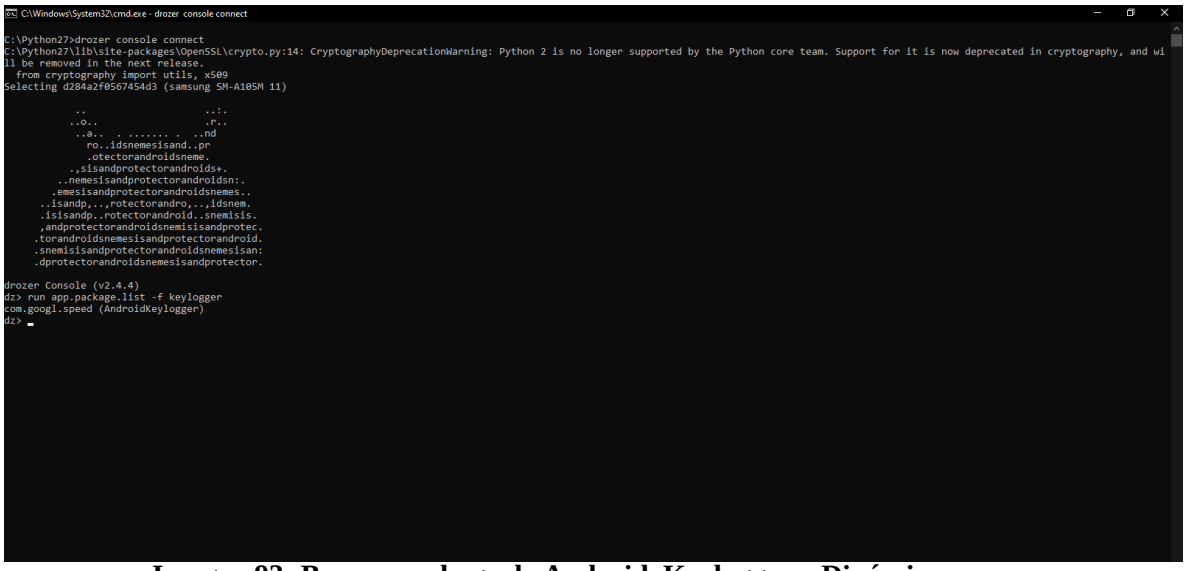


Imagen 93: Buscar package de Android_Keylogger – Dinámico

57. Con el comando "run app.package.info –a com.googl.speed" sirve para mostrar la información del paquete de la aplicación, como los permisos que administra.

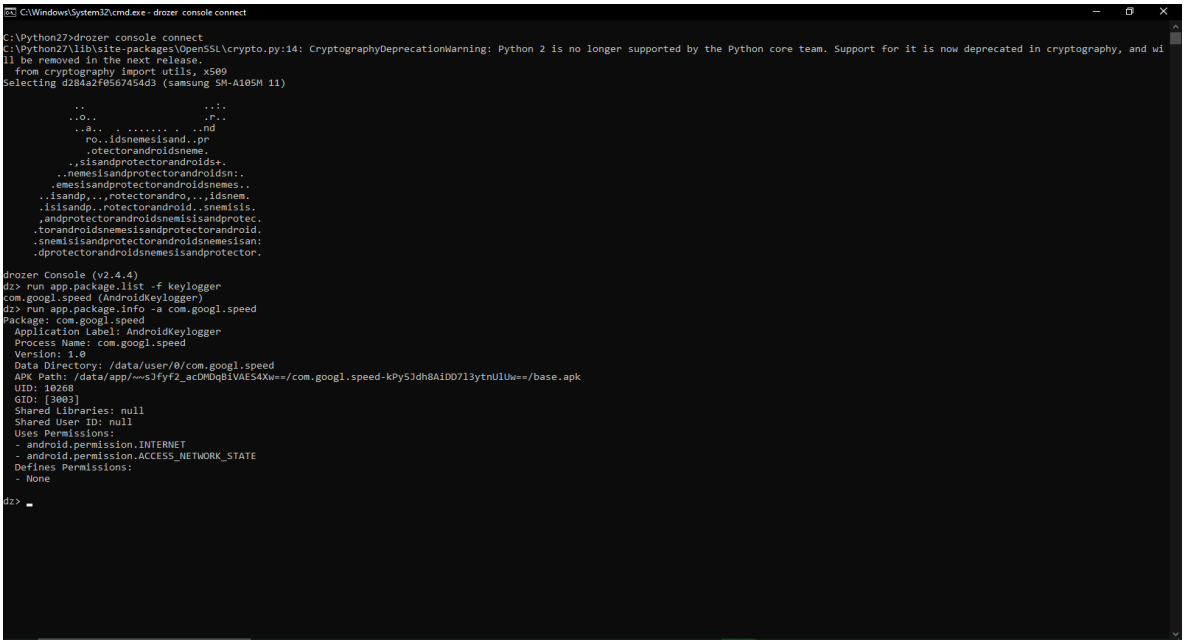


Imagen 94: Mostrar información del package de Android_Keylogger – Dinámico

58. Con el comando " run app.package.attacksurface com.googl.speed" sirve para en listar los servicios, broadcast y las actividades exportadas que el paquete de la aplicación cuenta.

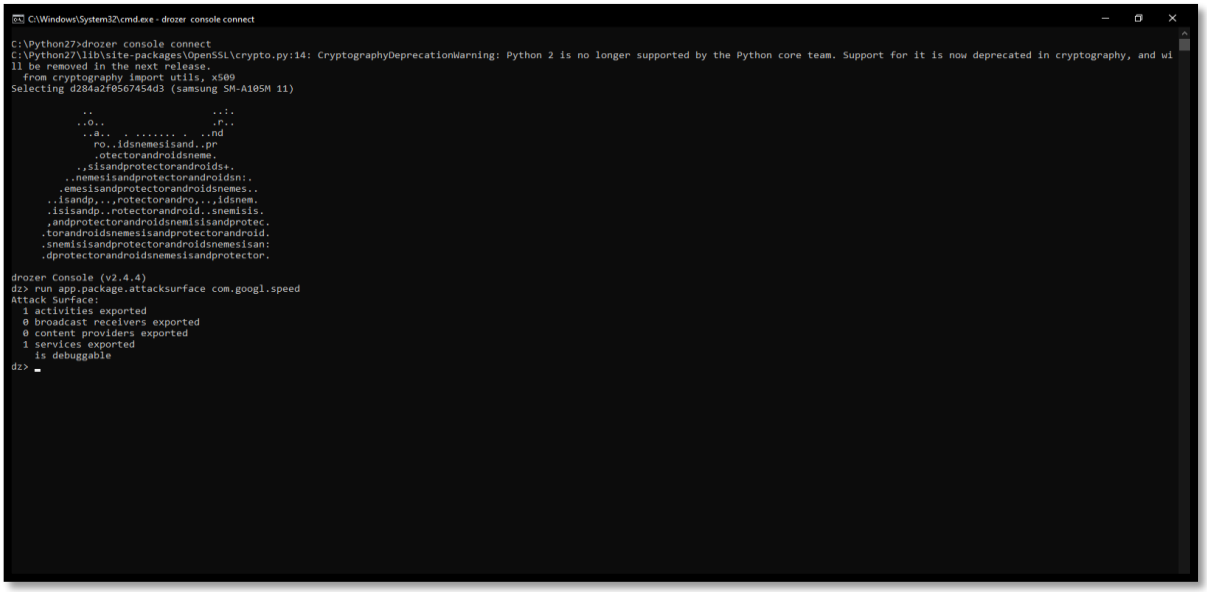


Imagen 95: Listar las actividades de package Android_Keylogger – Dinámico

59. Con el comando "run app.activity.info –a com.googl.speed" sirve para listar los servicios que el paquete cuenta.

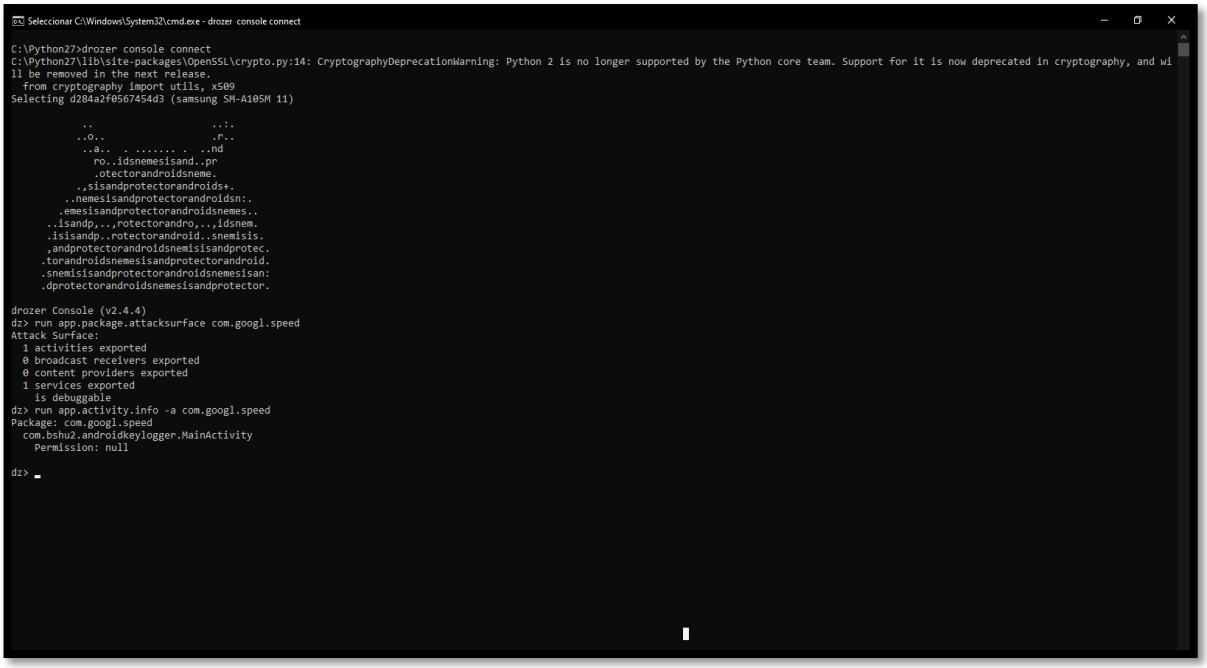


Imagen 96: Listar servicios de package Android_Keylogger – Dinámico

60. Con el comando "run app.activity.strart –component com.googl.speed com.bshu2.androidKeylogger.MainActivity, sirve para iniciar un servicio del paquete.

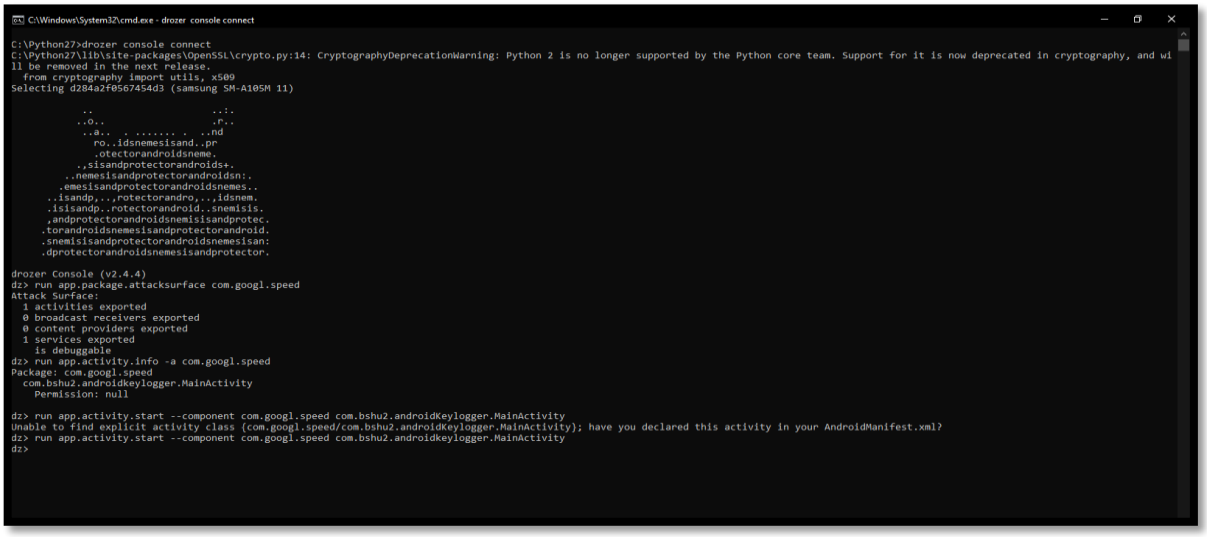


Imagen 97: Iniciar un servicio de package Android_Keylogger – Dinámico

61. Al ejercer el inicio de la actividad la aplicación mostro cambio y envió un mensaje de permiso al móvil.

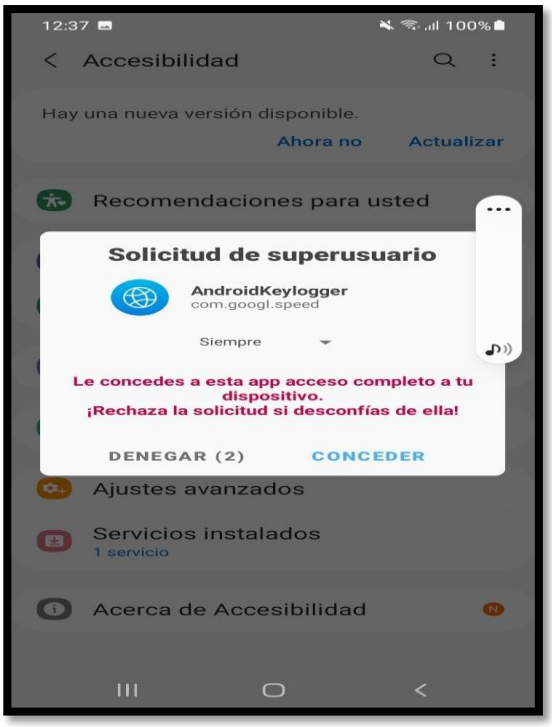


Imagen 98: Inicio de la aplicación Android_Keylogger por el servicio – Dinámico

ANEXO #5
FASE IV: PRUEBAS DE PENETRACIÓN

ESCENARIOS DE PRUEBAS

MALWARE VIRUS – PRUEBA.APK

Escenario: _1: Conocer el status del Android y crea un archivo de prueba txt

Objetivo: Elevar privilegios para evadir seguridad y encontrar activos informáticos

Tiempo: 25 minutos

1. Conocer el hostname de la maquina atacante para crear el virus informático con el siguiente comando, -p referencia al payload, LHOST al hostname de la máquina, LPORT el puerto de escucha por reverse_tcp y –o es el archivo de salía, que es formato. Apk

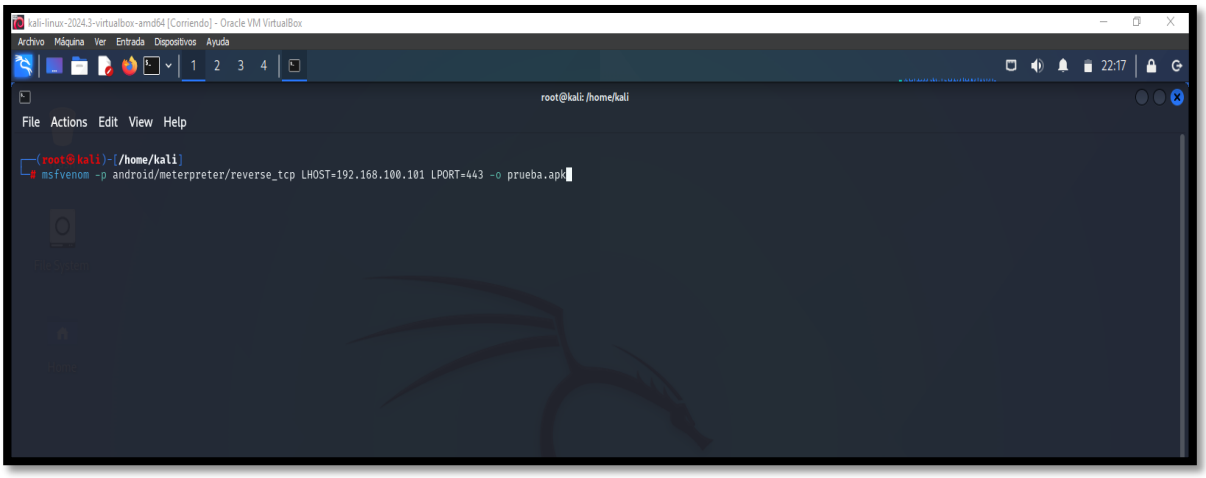


Imagen 99: Creación de payload malicioso como prueba.apk

2. Virus informático creado correctamente con las configuraciones definidas para ser usado en el dispositivo Android para evadir seguridad y conocer de su status como activos de información

Imagen 100: Creación de payload exitoso

3. Activar el servidor Python para que el archivo payloads se descargue desde la maquina victima desde la misma línea de comunicación de red.

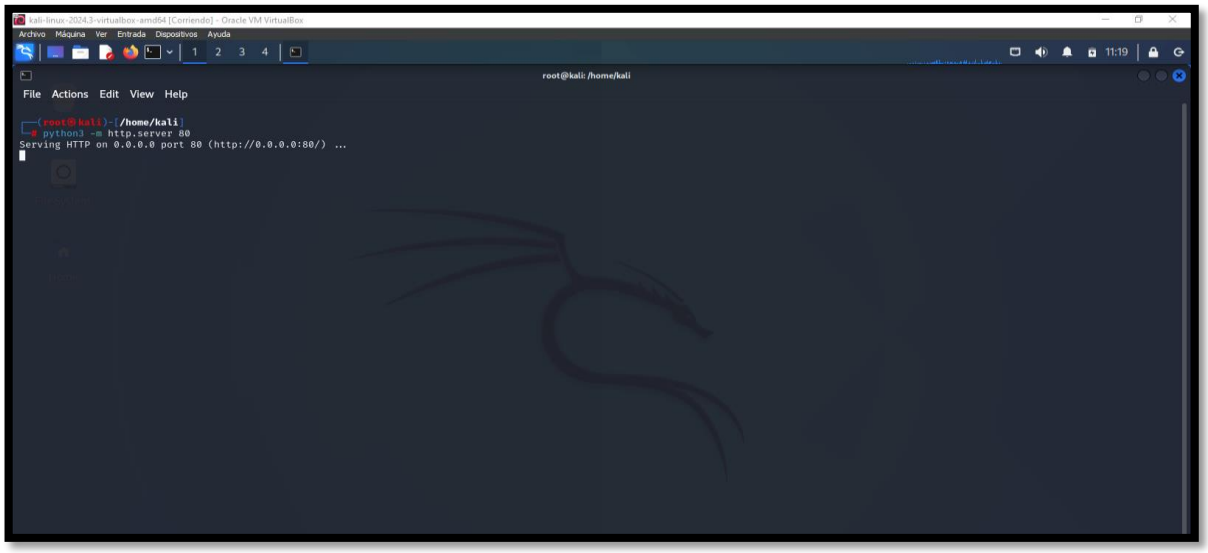


Imagen 101: Activar servidor python

4. En el dispositivo móvil se conecta al servicio de red puente del router, y se inserta la dirección en el navegador “http://192.168.100.101/ y se debe conectar al servidor de python para interactuar con los archivos presentes

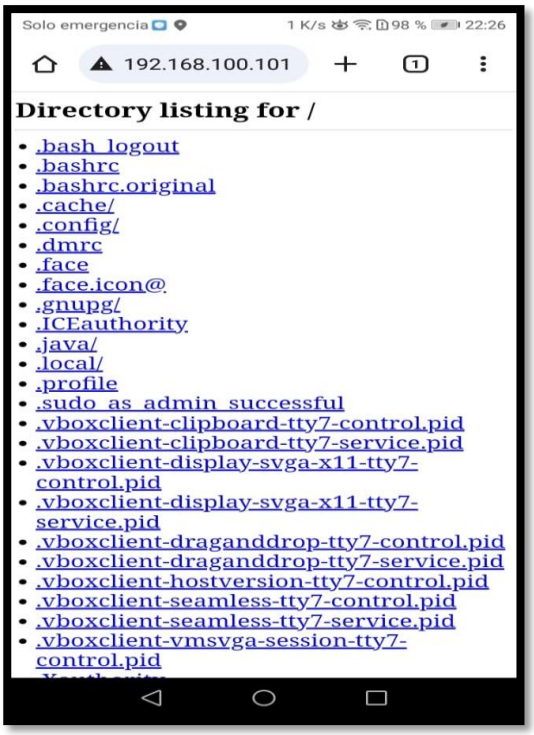


Imagen 102: Descargar la Apk desde el móvil

5. Buscar el archivo creado por msfvenom nombrado “prueba.apk”

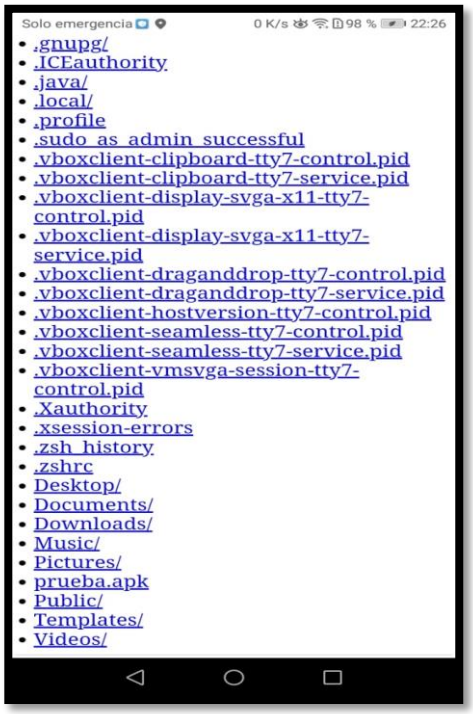


Imagen 103: Buscar el archivo prueba.apk

6. Descargar el archivo apk para proceder a su instalación en el dispositivo móvil. Dar en conservar para descargar el archivo sin importar las indicaciones de segura

Imagen 104: Aceptar en conservar la apk aunque no sea segura

7. Se descarga correctamente el archivo apk para abrir

Imagen 105: Abrir la apk para instalar

8. Se presenta el portal de inicio de instalación de la aplicación con los permisos que se requiere

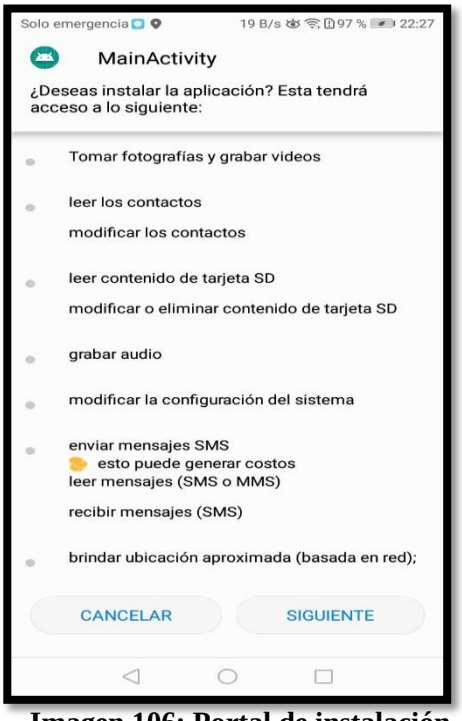


Imagen 106: Portal de instalación

9. Dar clic en instalar para aceptar todos los permisos de la aplicación sin riesgo alguno

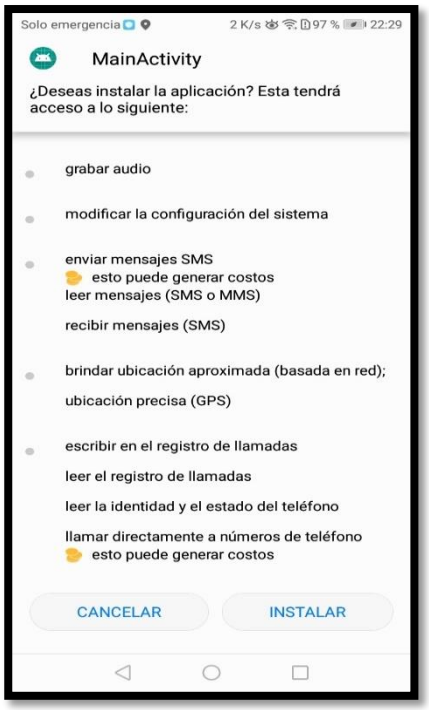


Imagen 107: Aceptar todos los permisos

10. Al iniciar el proceso de instalación de la aplicación, google play protect manda una notificación de seguridad para activar, pero dar clic en rechazar para continuar con la instalación.

Imagen 108: Dar en instalar de todos modos

11. La instalación de la aplicación esta completada, pero así mismo google play protect manifiesta que la aplicación instalada posiblemente sea un archivo dañino que puede dañar el dispositivo como virus informático, pero aun asi se da clic en ignorar.

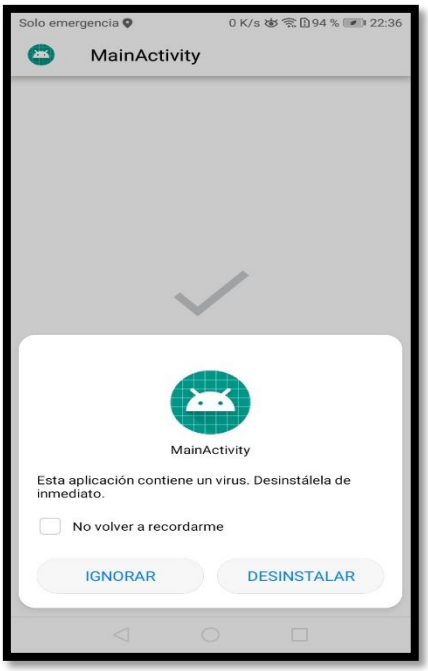


Imagen 109: Dar clic en ignorar

12. Aplicación instalada correctamente y se presenta en la lista de las aplicaciones del dispositivo Android móvil.

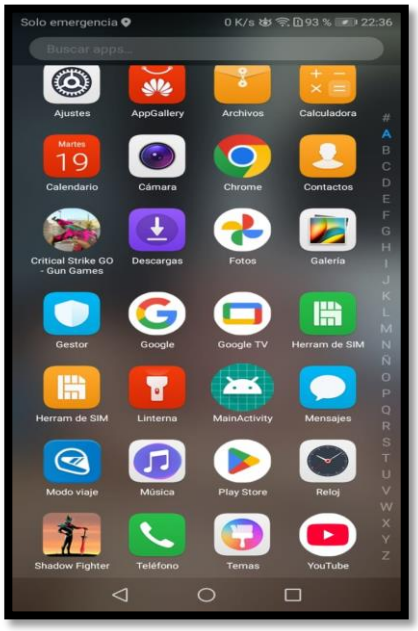


Imagen 110: Instalación completa de la apk

13. Se ejecuta el comando msfconsole para configurar correctamente la interacción del archivo malicioso cuando es ejecutado con un click

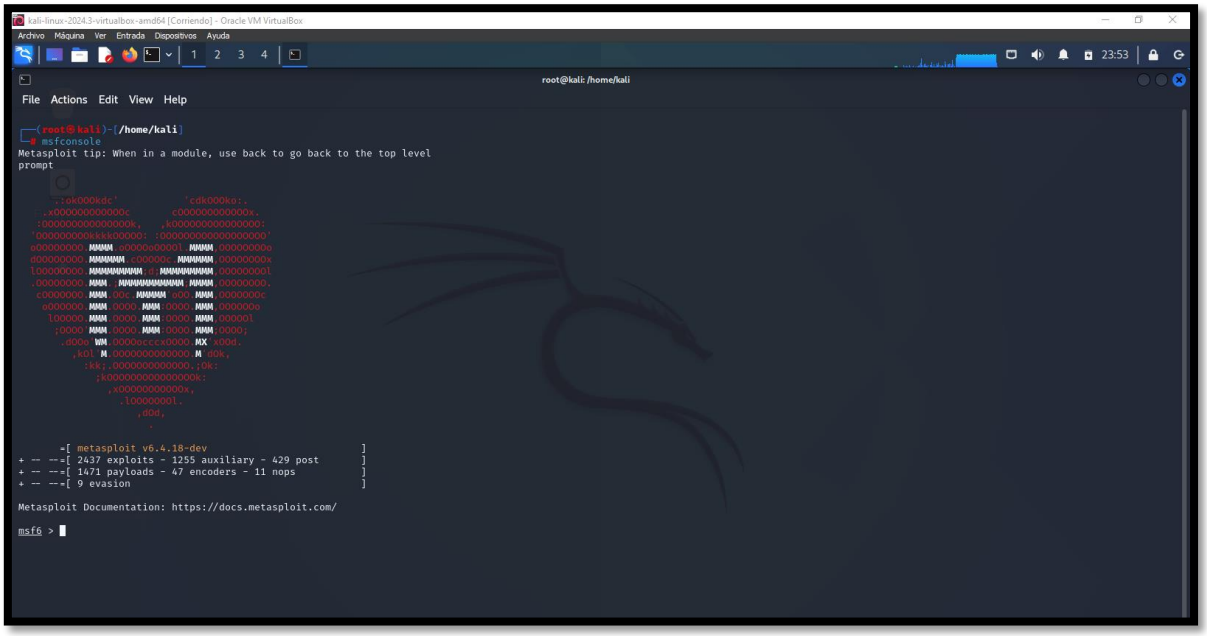


Imagen 111: Ejecutar msfconsole para configurar el payload de la APK

14. Metasploit ejecutado perfectamente, se inserta el comando “use multi/handler” para permitir recibir sesión de control de un dispositivo objeto y gestionar el acceso lo cual es fundamental en pruebas de penetración.

Imagen 112: Ejecutar el comando use multi/handler

15. Comando “set payload android/meterpreter/reverse_tcp” es el tipo de payload utilizado en el archivo malicioso que va a permitir escuchar entre máquina victima a máquina atacante

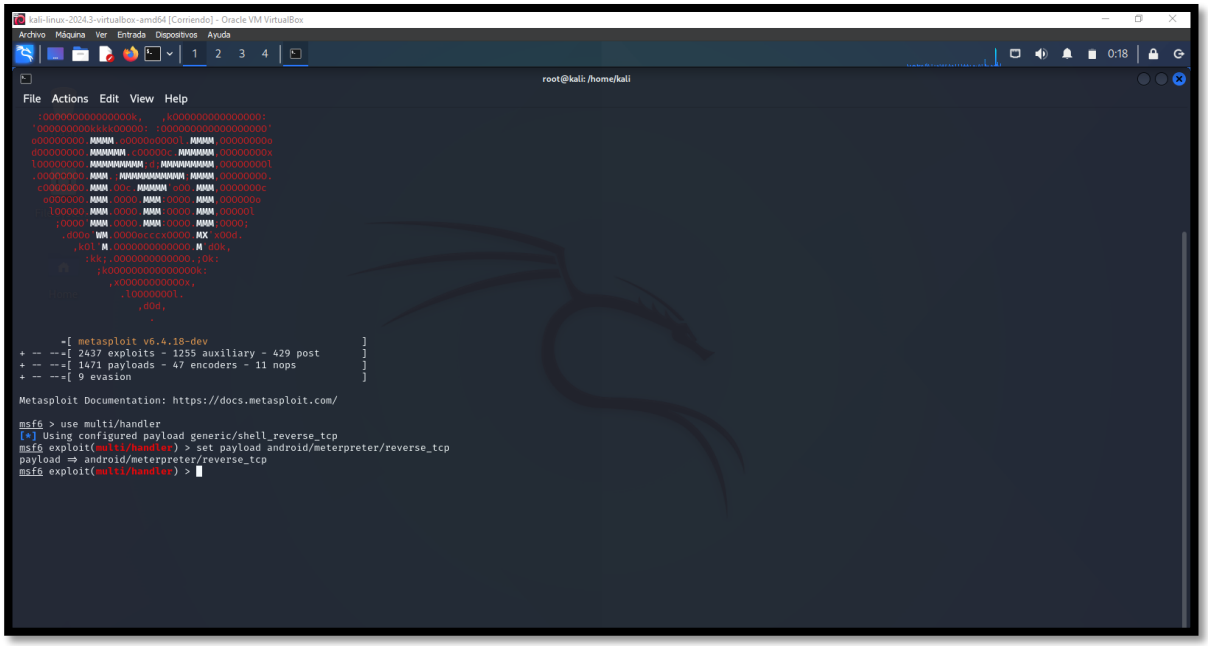


Imagen 113: Insertar el payload de configuración a la prueba.apk

16. Con el comando show options se libera todas las configuraciones que se deben definir para la correcta función del use multi/handler hacia la carga útil(payloads)

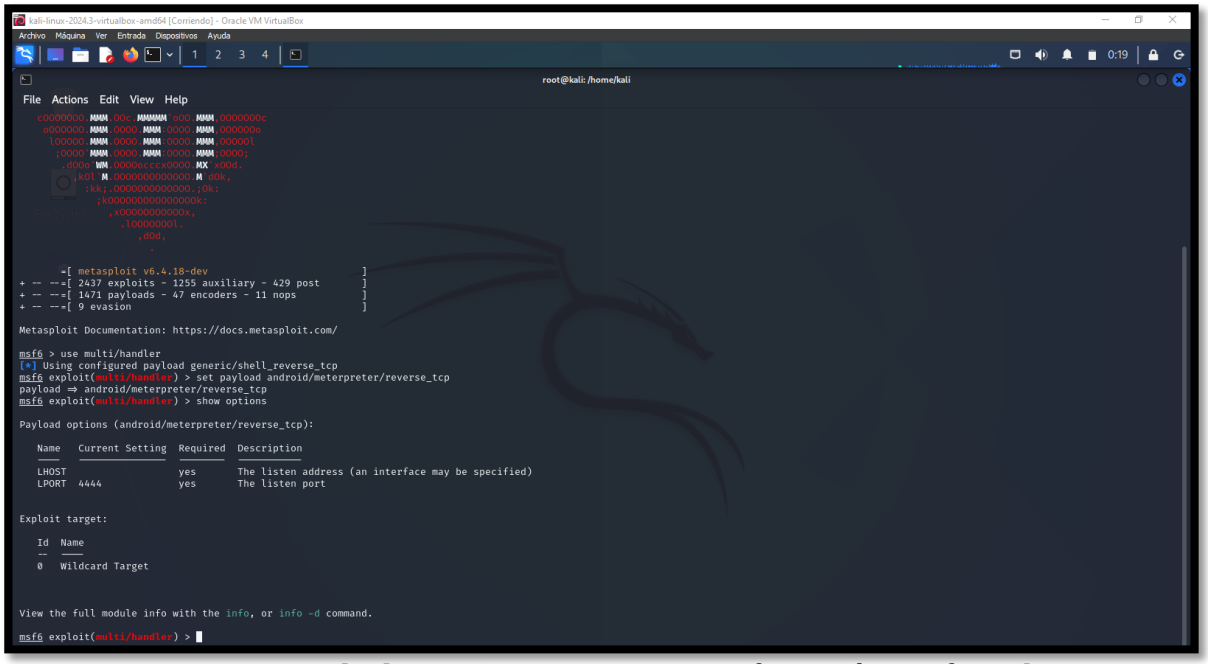


Imagen 114: Comando show options para ver que configuración esta firmada

17. Se configura la dirección de la maquina víctima y el puerto de escucha con el comando set LHOST y el set LPORT

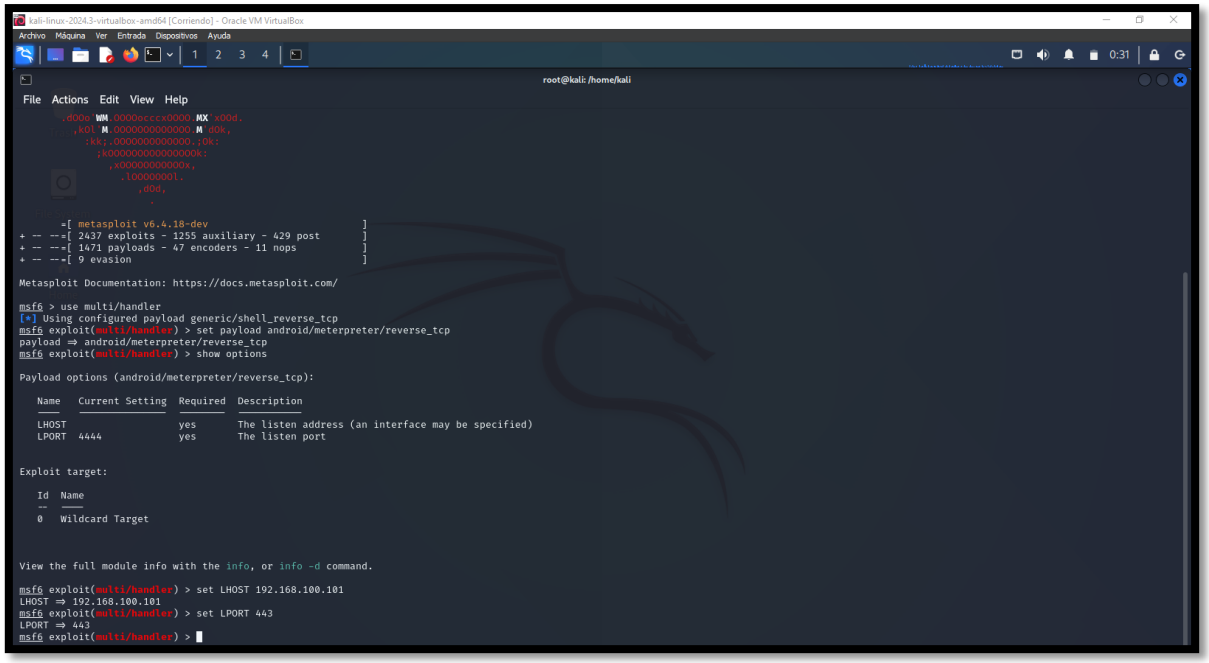


Imagen 115: Insertar el LHOST Y LPORT de la máquina victima

18. Una vez configurado exitosamente se libera el comando “run o exploit” para correr la escucha del reverse_tcp hacia el payload malicioso instalado en el dispositivo Android

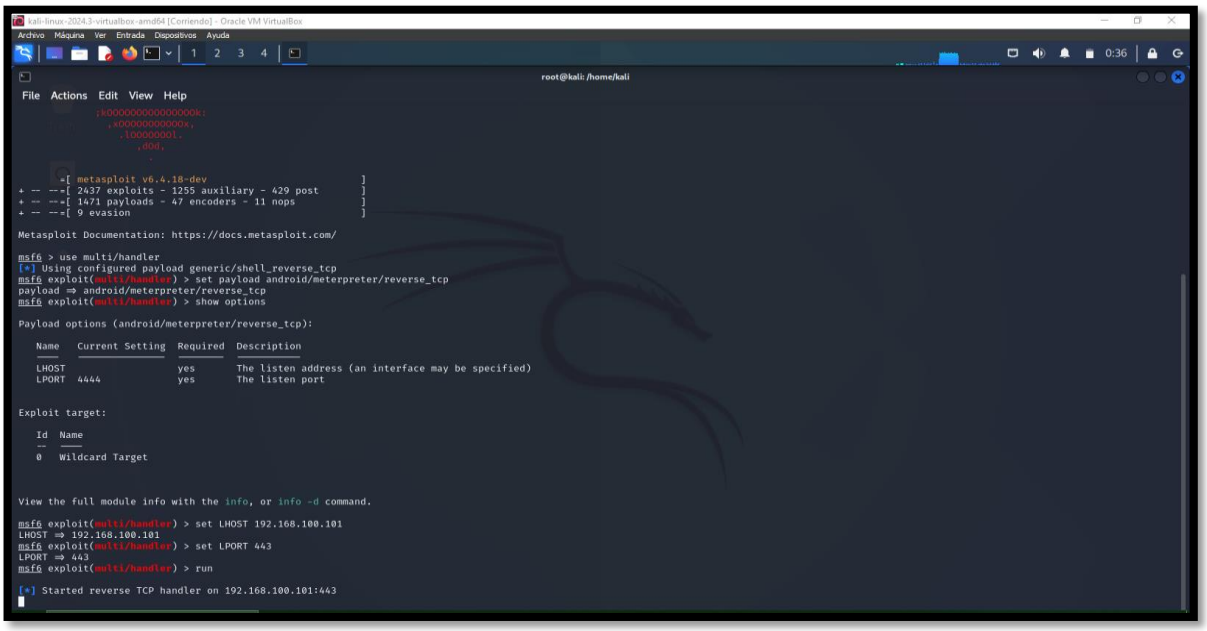


Imagen 116: Ejecutar run para que desarrolle la herramienta

19. Una vez que el usuario víctima de clic a la aplicación malicioso en su dispositivo móvil, se da sesión en la maquina atacante en formato terminal.

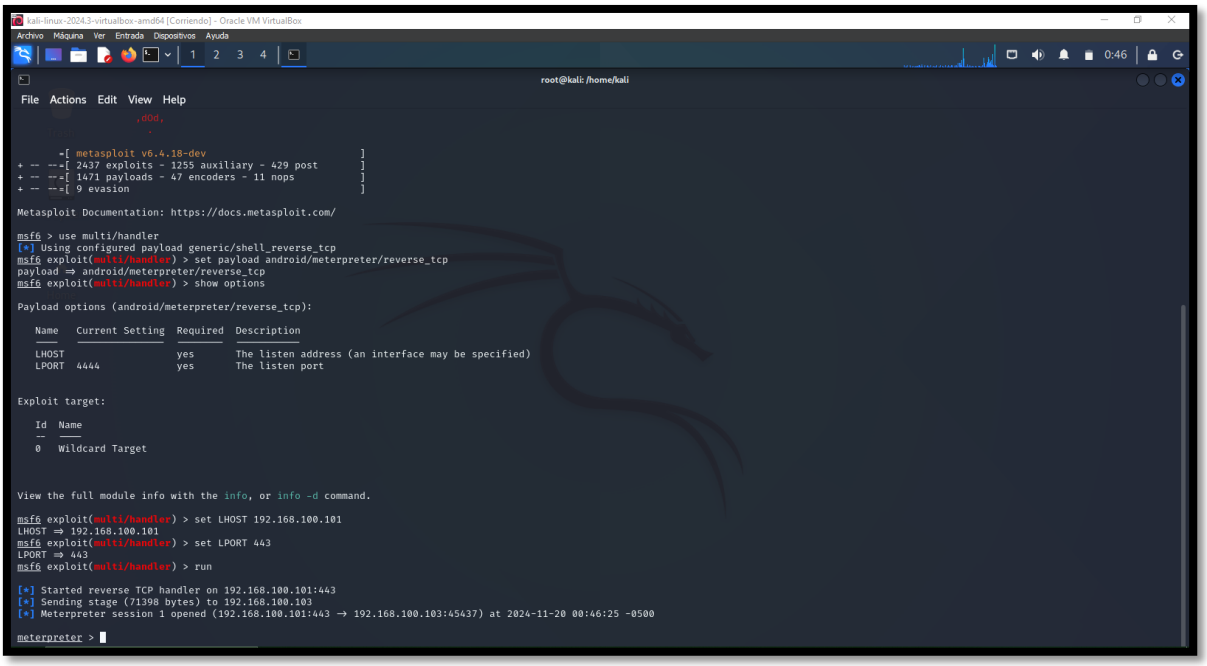


Imagen 117: Sesión iniciada correctamente

20. Con el comando help se observa una lista de acción que puede emplear meterpreter y a su vez el uso del comando “dump_contacts” que sirve para extraer todos los contactos que existen en el dispositivo.

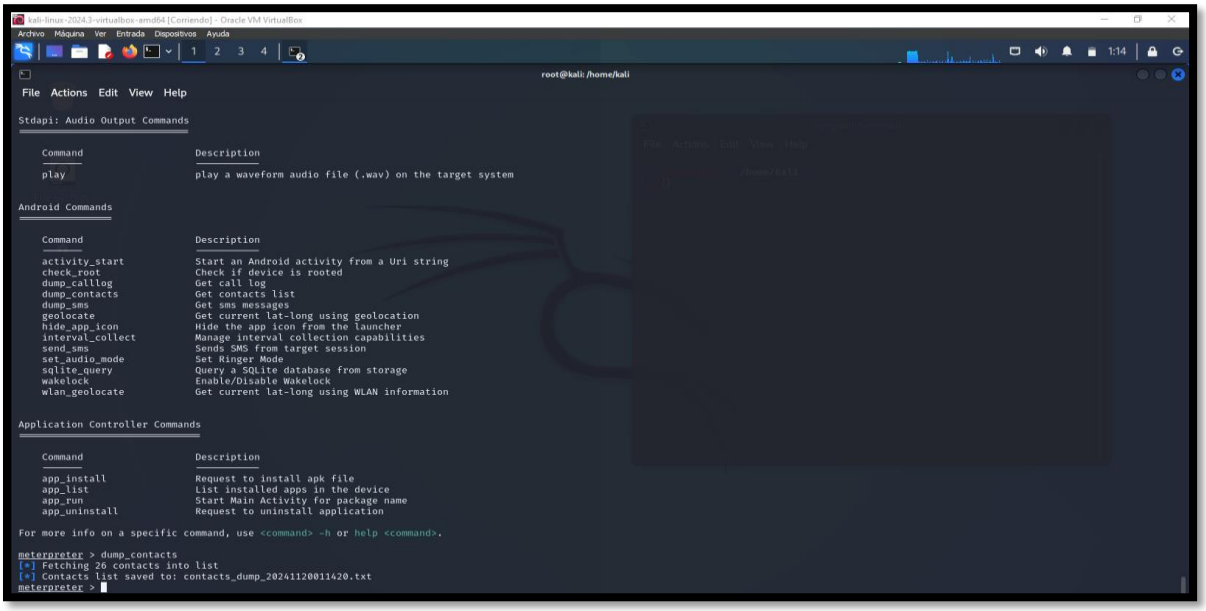


Imagen 118: dump_contacts descragar la lista de contactos

21. Con el comando “dump_sms” se extraer de misma forma toda la información relevante de los mensajes de texto que el dispositivo cuenta.

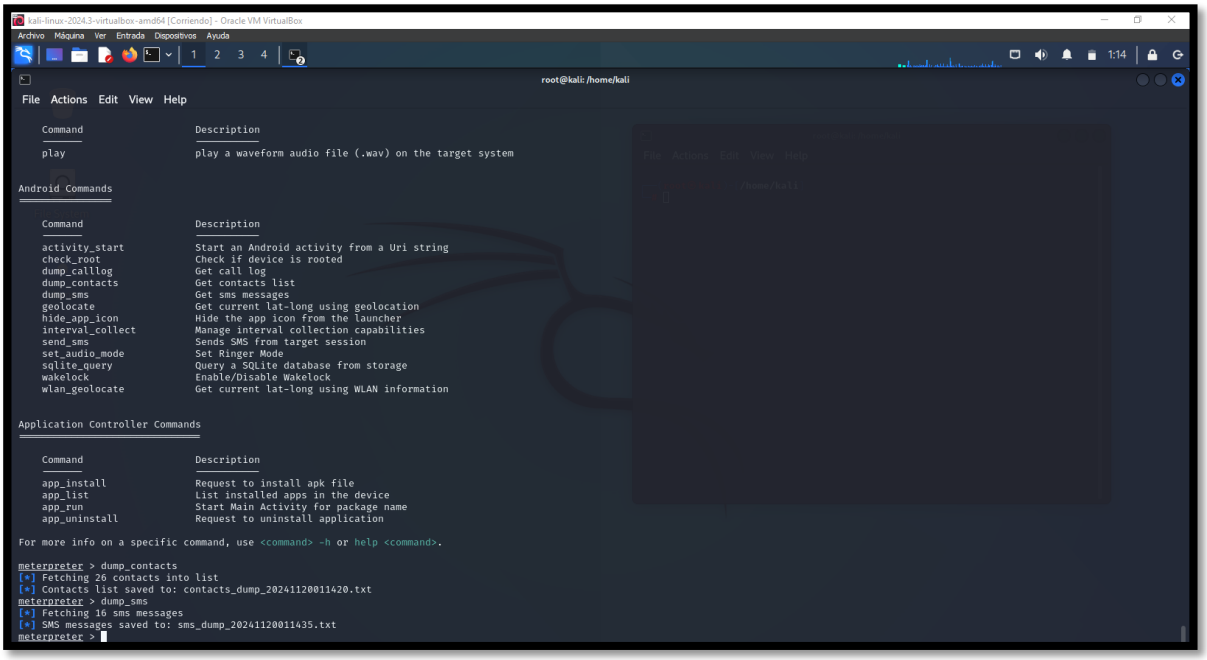


Imagen 119: Dump_SMS extracción de mensajes del móvil

22. Con el comando app_list se observa la lista de las aplicaciones que el usuario cuenta en su dispositivo Android.

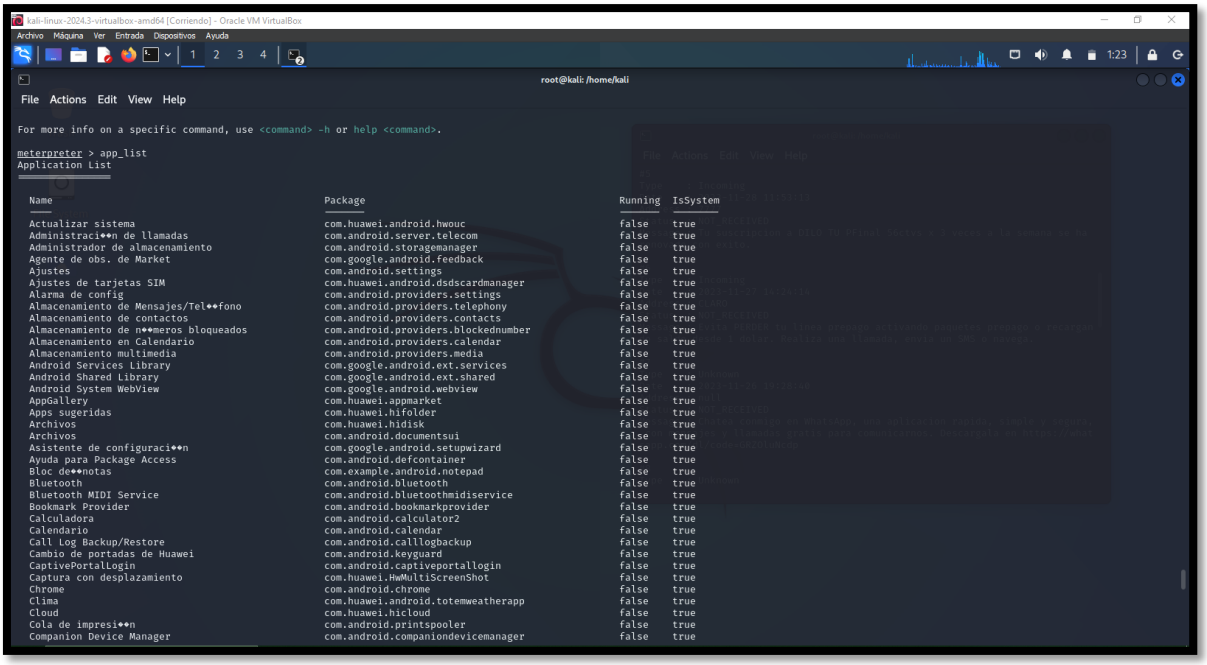


Imagen 120: Comando app_list ejecuta todas las aplicaciones del Android

23. Con el comando geolocate, permite extraer la información de la ubicación del dispositivo Android

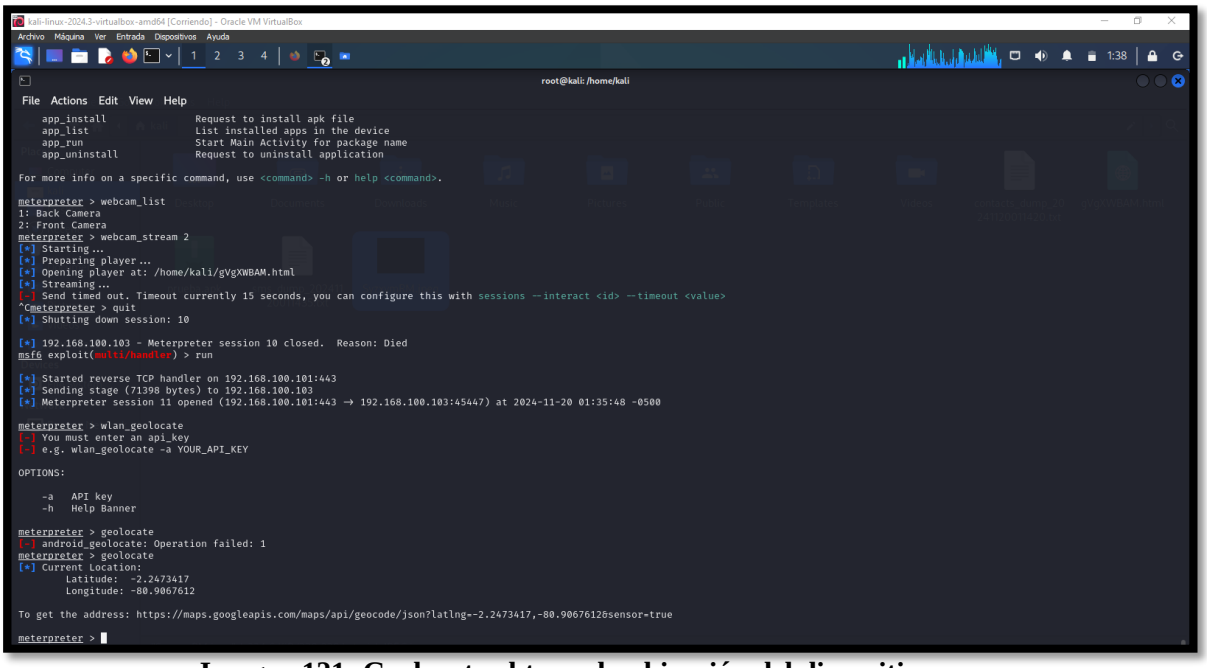


Imagen 121: Geolocate obtener la ubicación del dispositivo

24. Direccionarse a las carpetas del sdcard para indagar que imágenes cuenta y se observa 3 imágenes en cuestión

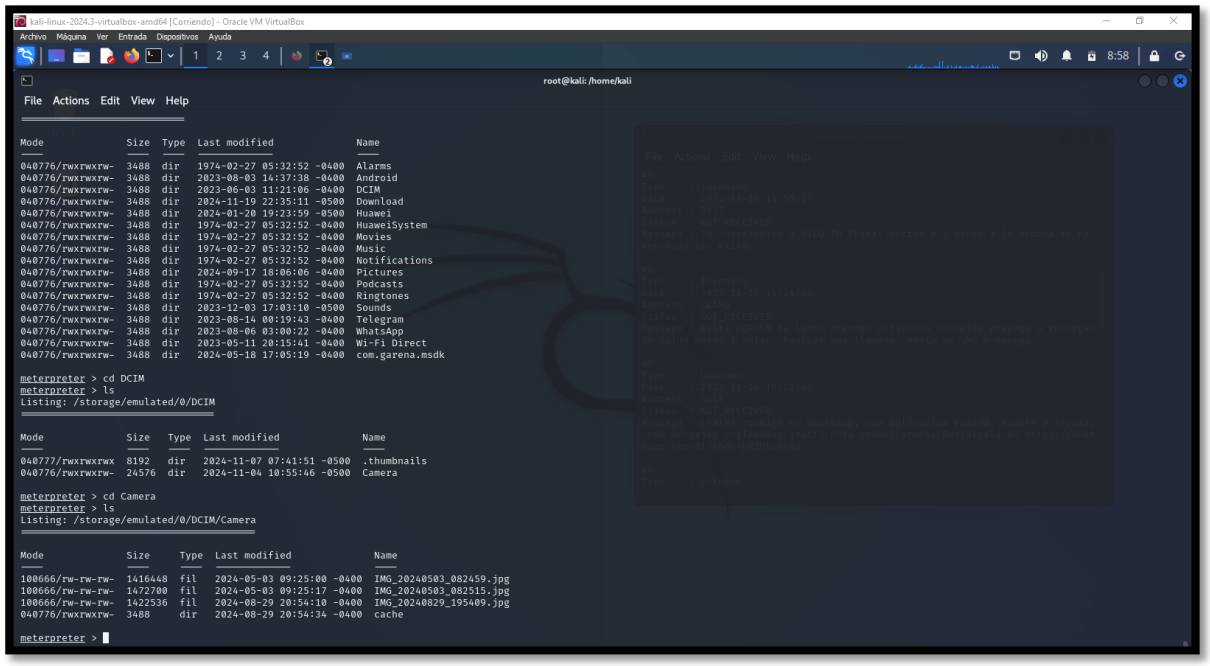


Imagen 122: Verificación de rutas para hallar imágenes

25. Con el comando “download /storage/emulted/0/DCIM/Camera/nombre_archivo.jpg” se desarrolla la descarga de la imagen, que en este caso son 3 imágenes.

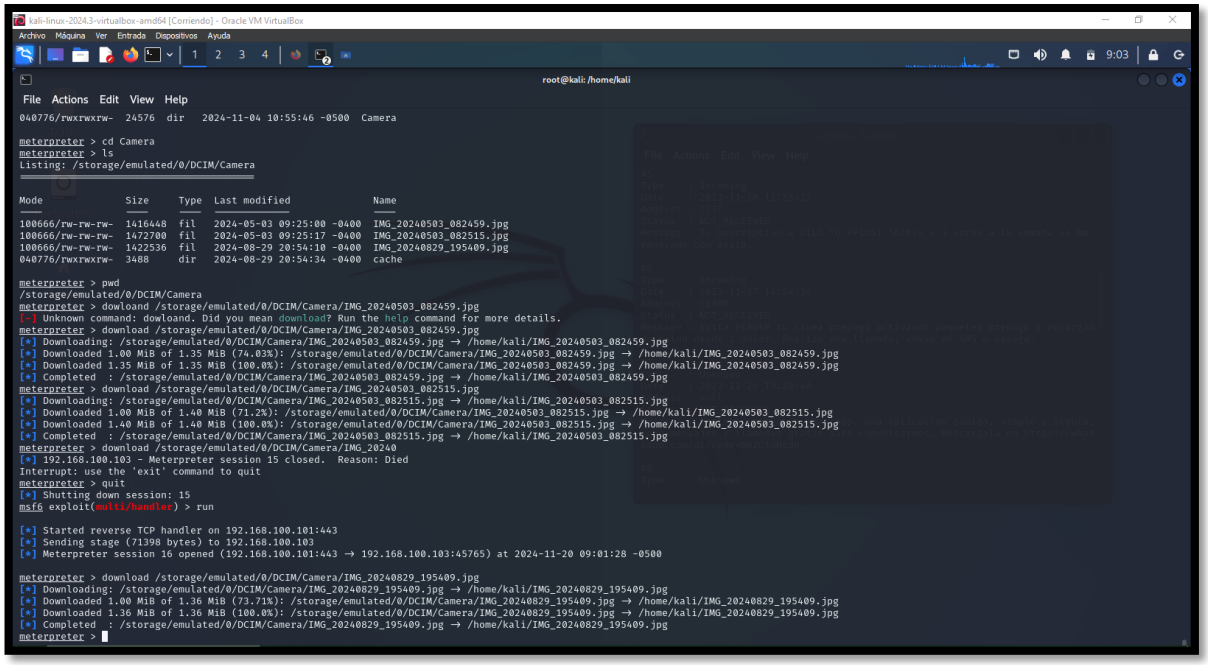


Imagen 123: Descargar las imágenes del dispositivo comprometido

26. Cómo resultado final se obtuvo los contactos, los sms, imágenes, geolocalización, y una prueba de sonido.

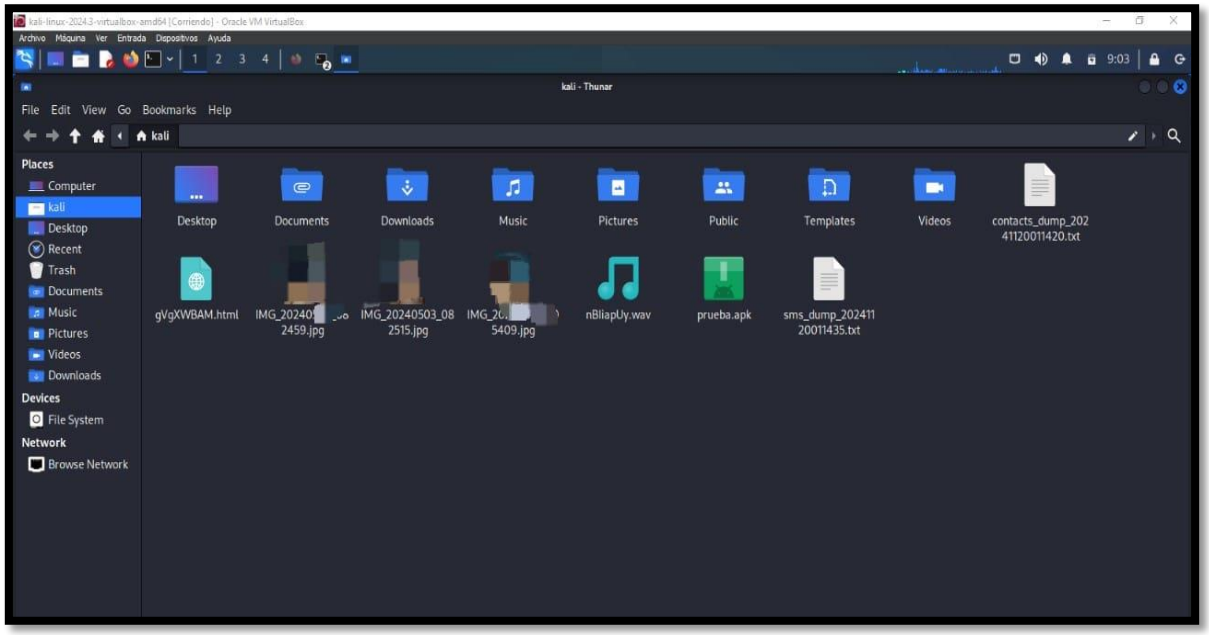


Imagen 124: Información general de lo hallado en la computadora.

MALWARE – ANDROID_KEYLOGGER

Escenario: _2: Capturar pulsaciones de teclado del Android

Objetivo: Capturar todas las acciones del teclado Android para encontrar información relevante

Tiempo: 8 minutos

1. Descargar la apk desde el repositorio Github del siguiente link "https://github.com/IceWreck/LokiBoard-Android-Keylogger?tab=readme-ov-file" aplicación prebuilt apks

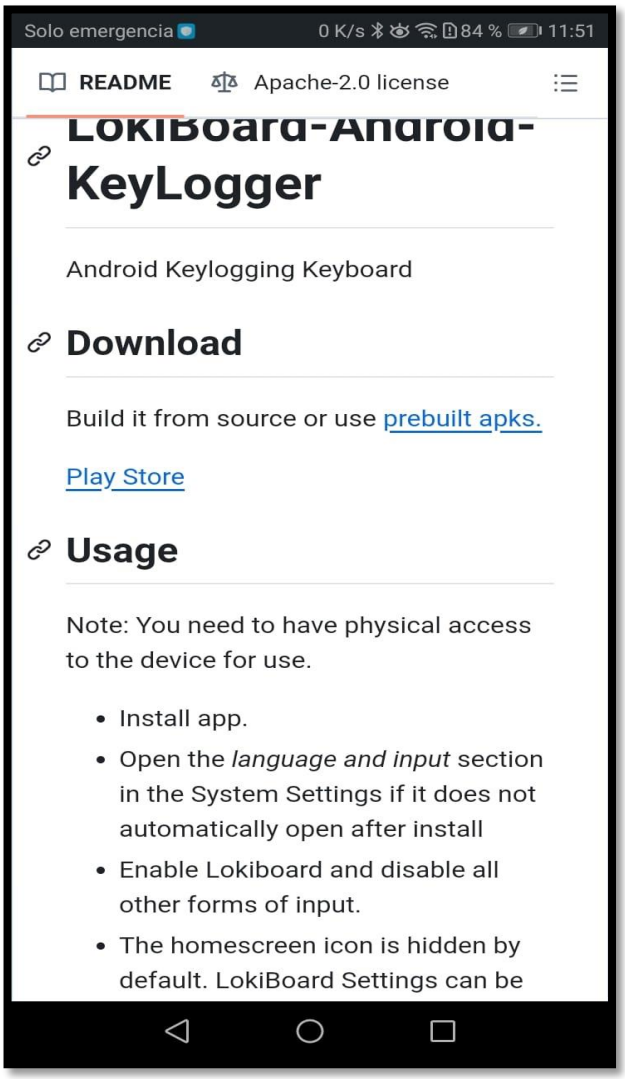


Imagen 125: Descargar la apk desde github

2. Al realizar clic en prebuilt apks comienza la descarga de la aplicación para su instalación.

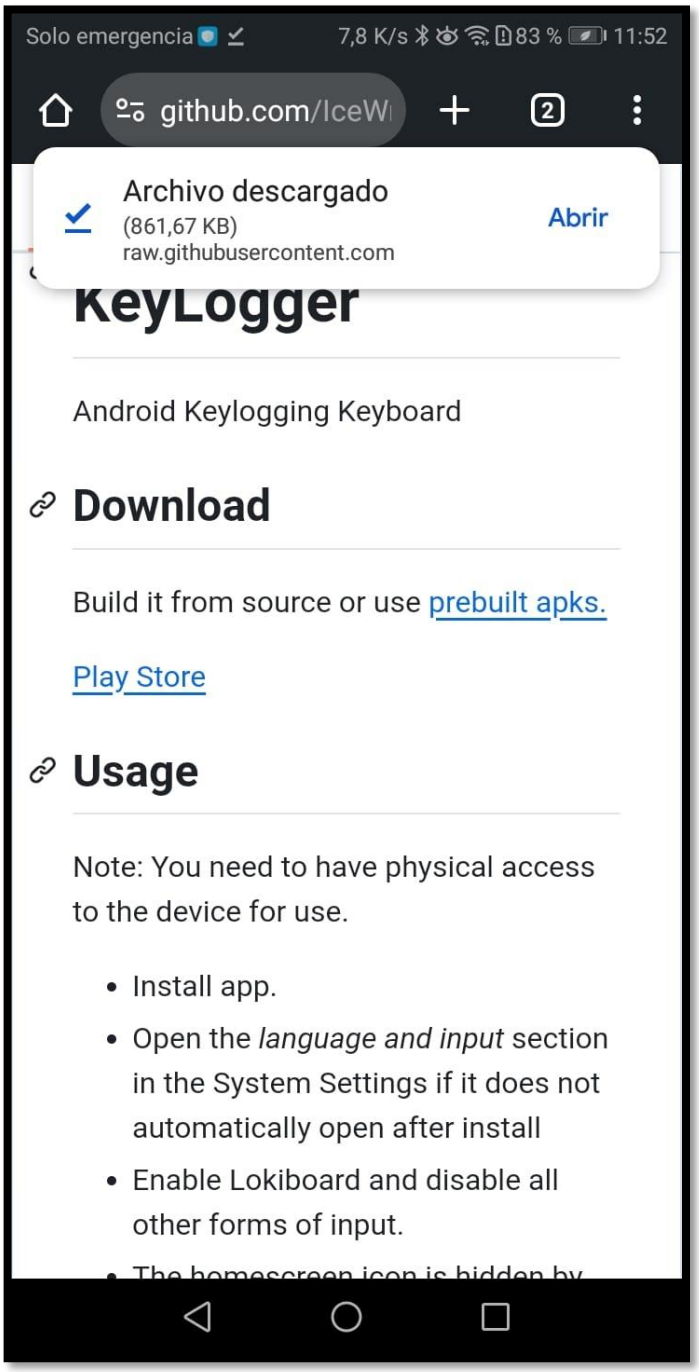


Imagen 126: Abrir la apk prebuilt.apks

3. Una vez finalizado la descarga, se procede abrir el archivo para instalar la apk en el dispositivo móvil.

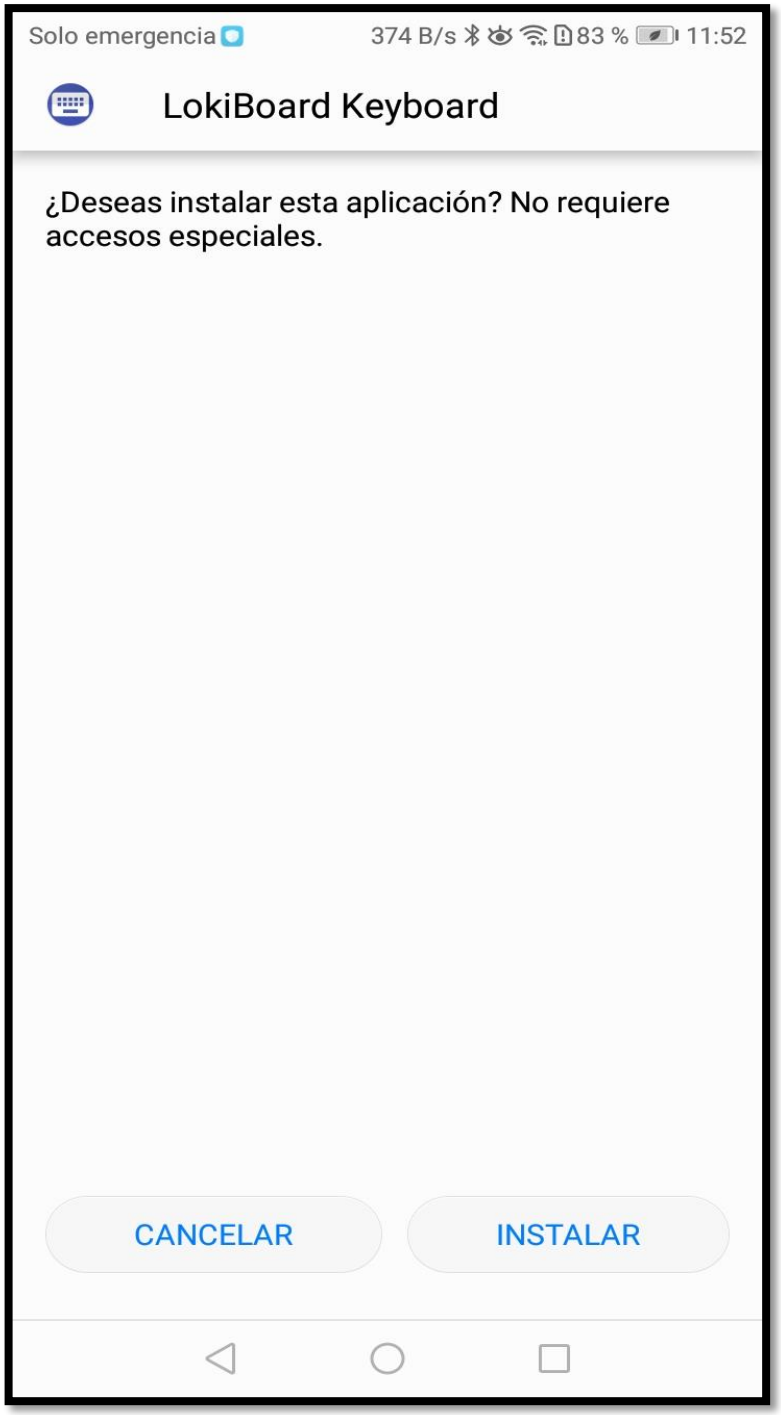


Imagen 127: Portal de instalación – Instalar

4. Aplicación instalada correctamente

Imagen 128: Instalación correcta

5. Una vez se ejecuta la aplicación se representa una notificación de configuración para poder interactuar con el LokiBoard Keyboard, y dar en OK

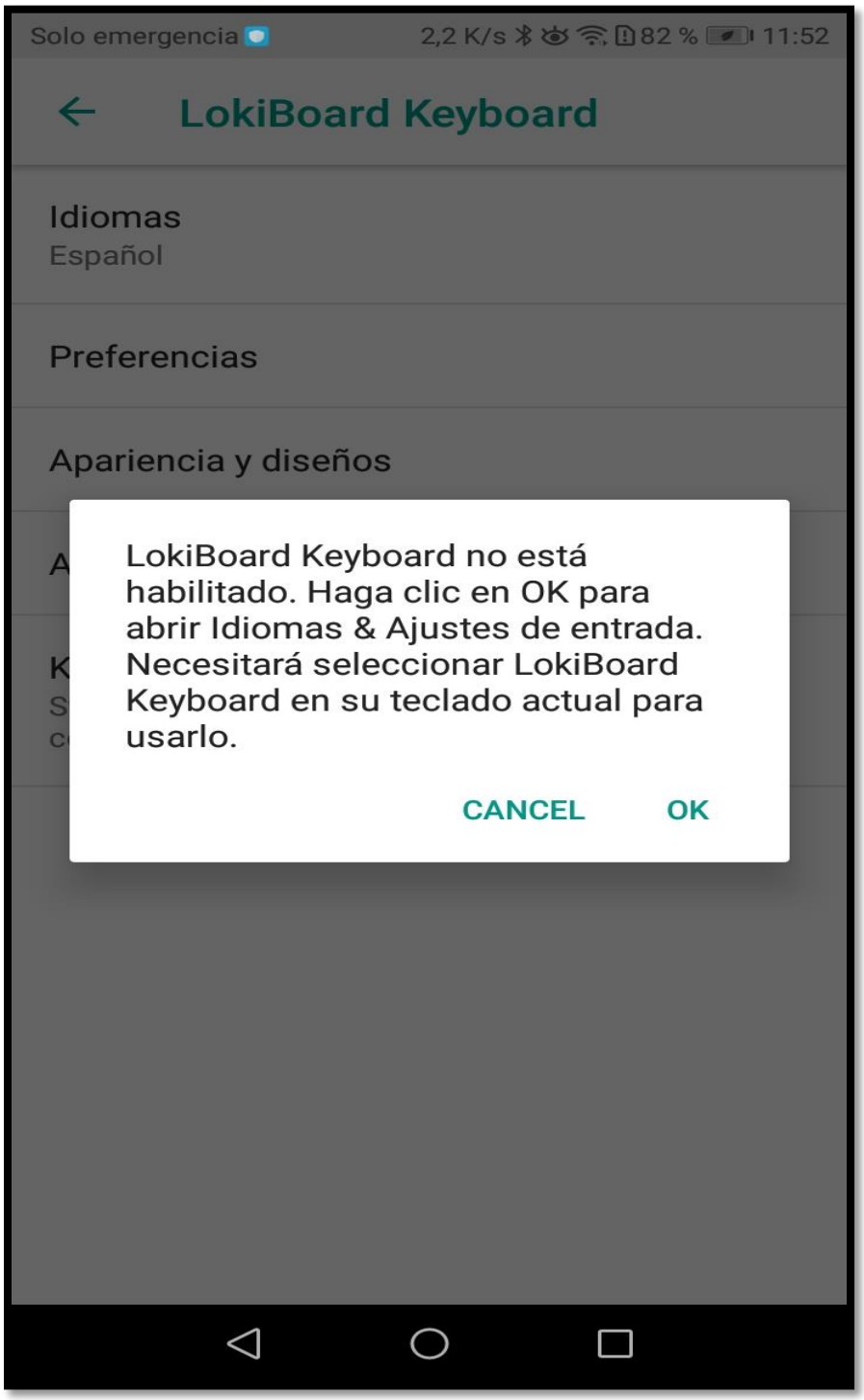


Imagen 129: Ejecutar la aplicación y dar OK

6. Al dar OK nos direcciona a la configuración del teclado virtual del dispositivo móvil y aparece el de la aplicación instalada, se lee la notificación de manera detenida y se observa que información se puede obtener, dar en OK

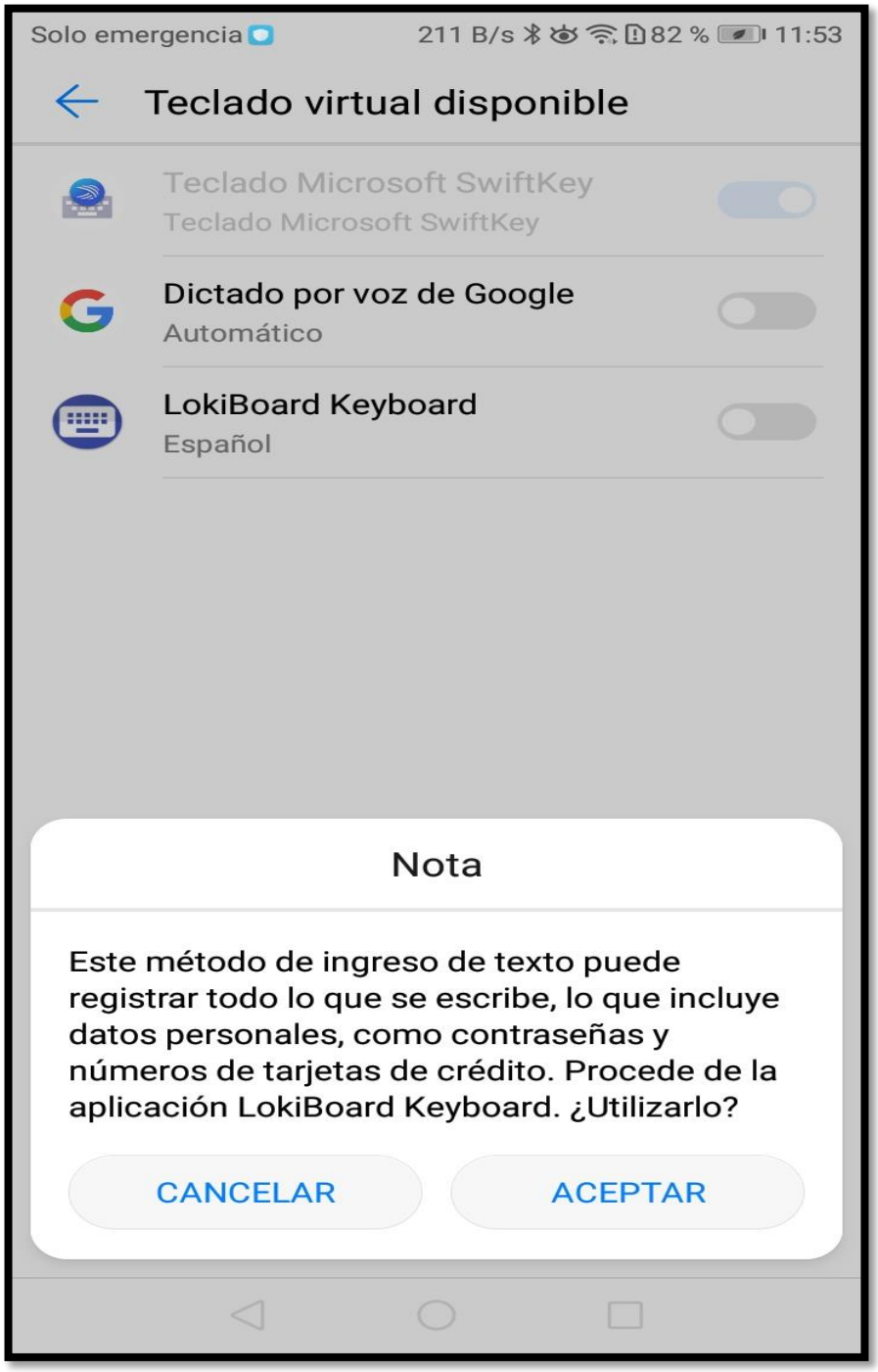


Imagen 130: Aceptar el método de ingreso de texto de la apk

7. Aparece otra notificación a tomar en cuenta, desde que el dispositivo sea reiniciado o bloqueado a desbloquear la aplicación reinicia el registro y luego de leer lo establecido dar en “Aceptar”

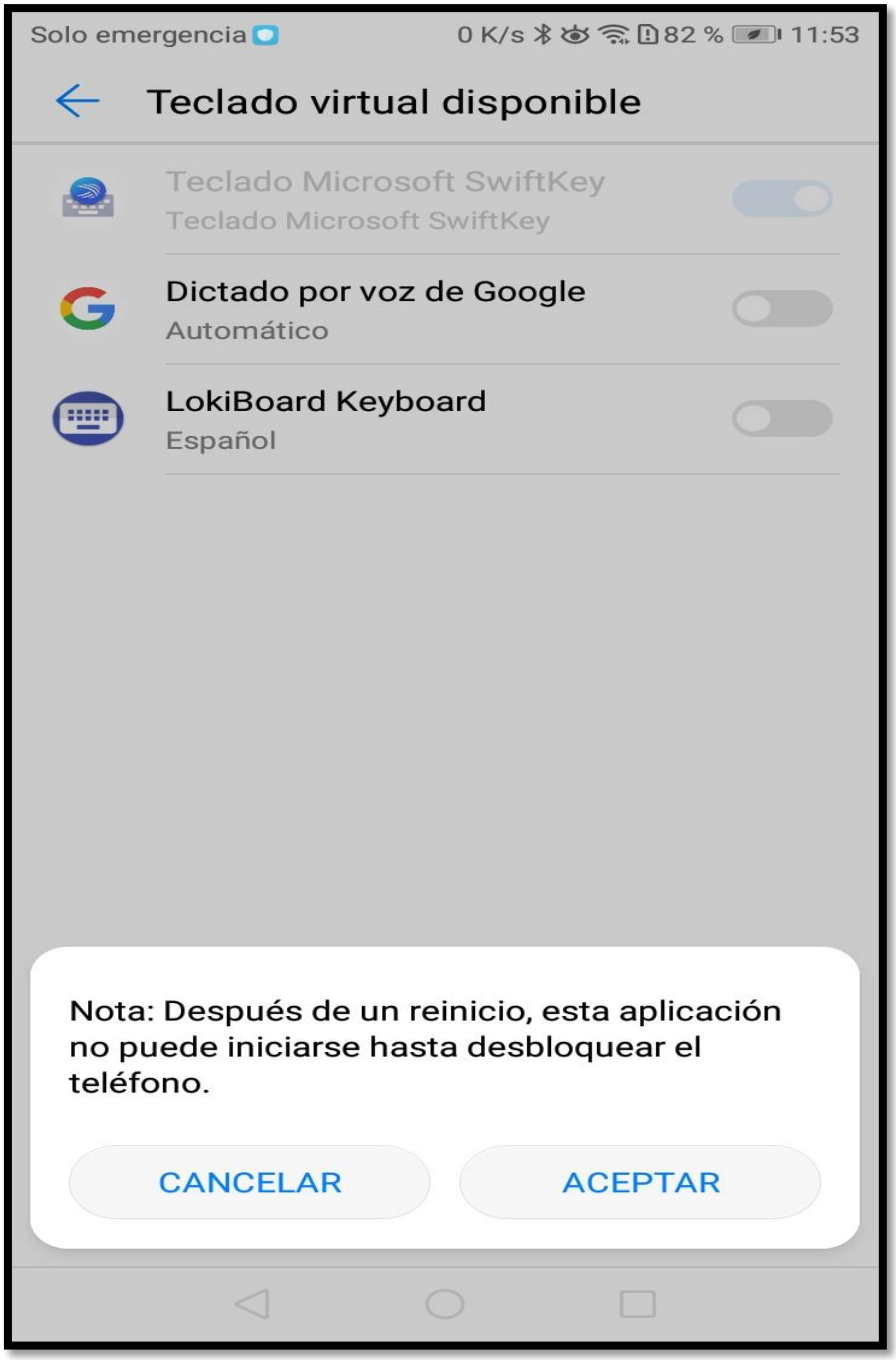


Imagen 131: Aceptar la notificación de desbloqueo

8. Una vez leído las notificaciones se selecciona el lokiboard keyboard a usar

Imagen 132: Activar el Lokiboard Keyboard

9. Otra parte esencial es seleccionar el teclado por determinado en el sistema del dispositivo móvil, seleccionar en Idioma e ingreso de texto

Imagen 133: Cambiar teclado predeterminado en sistema

10. En la opción de teclado predeterminado aparece el teclado por defecto del sistema operativo, seleccionar aquello y cambiar al Lokiboard Keyboard

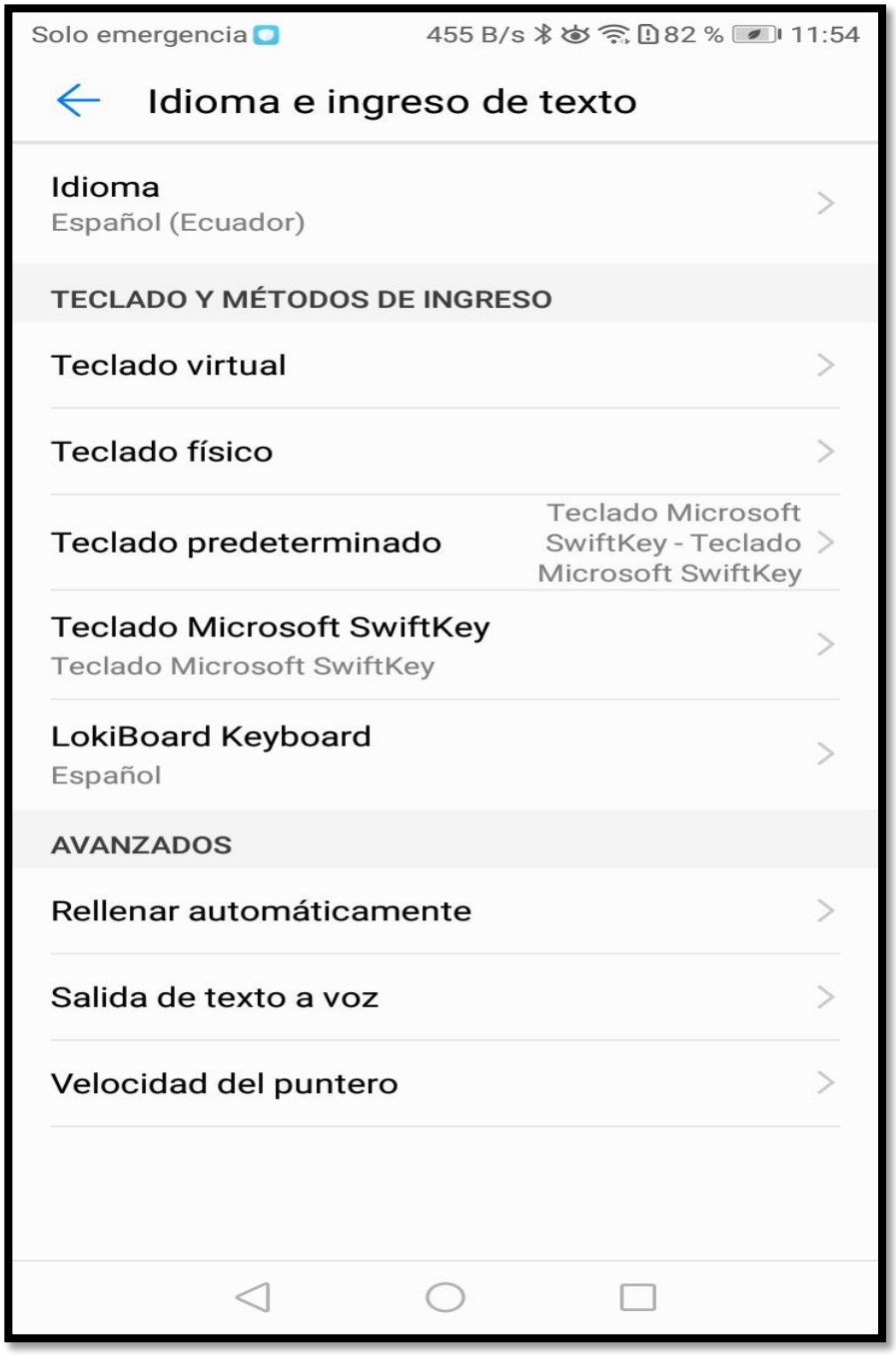


Imagen 134: Teclado predeterminado

11. Seleccionar Lokiboard keyboard

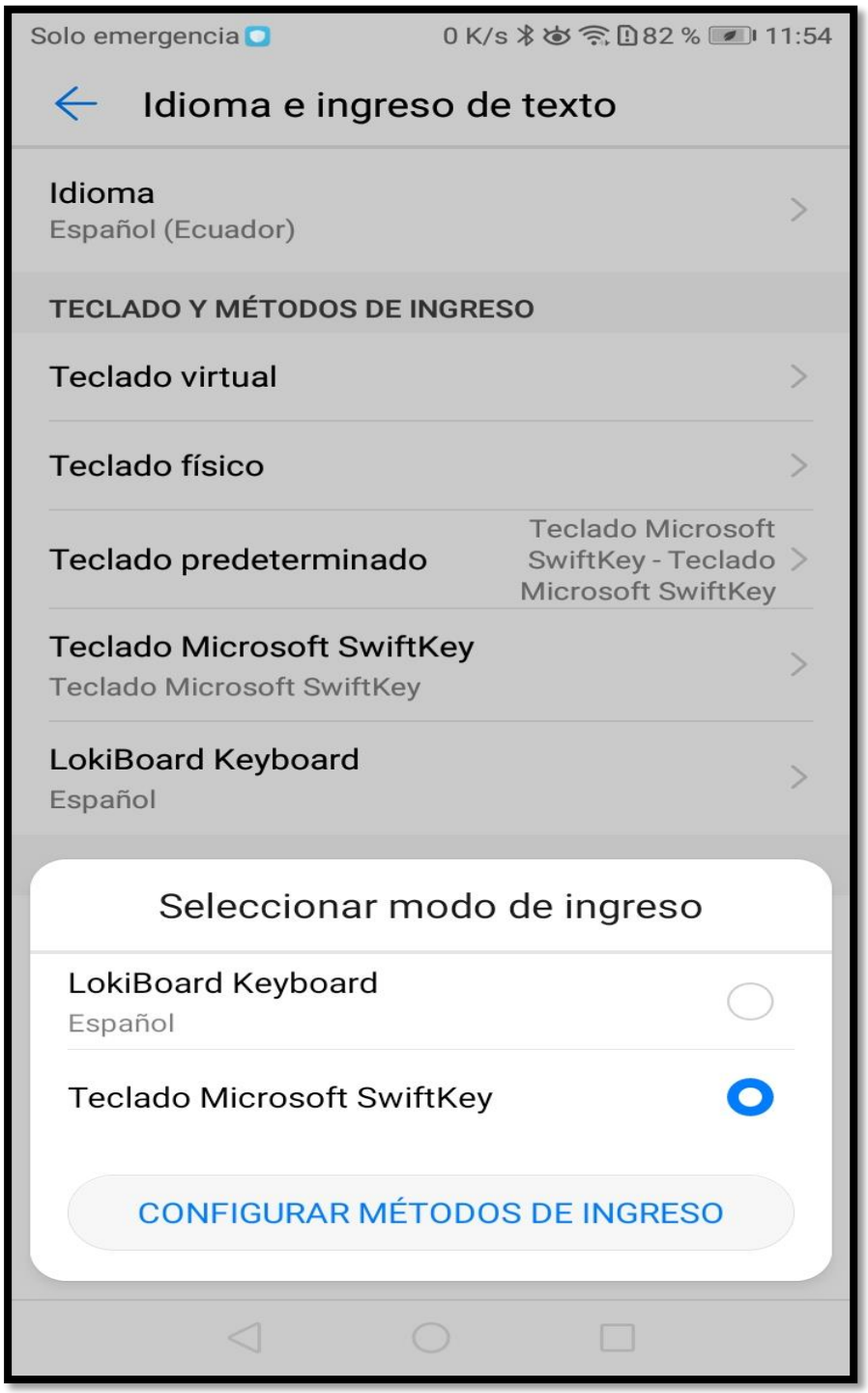


Imagen 135: Seleccionar Lokiboard como predeterminado

12. Configuración terminada exitosamente, ahora aprobar que captura.

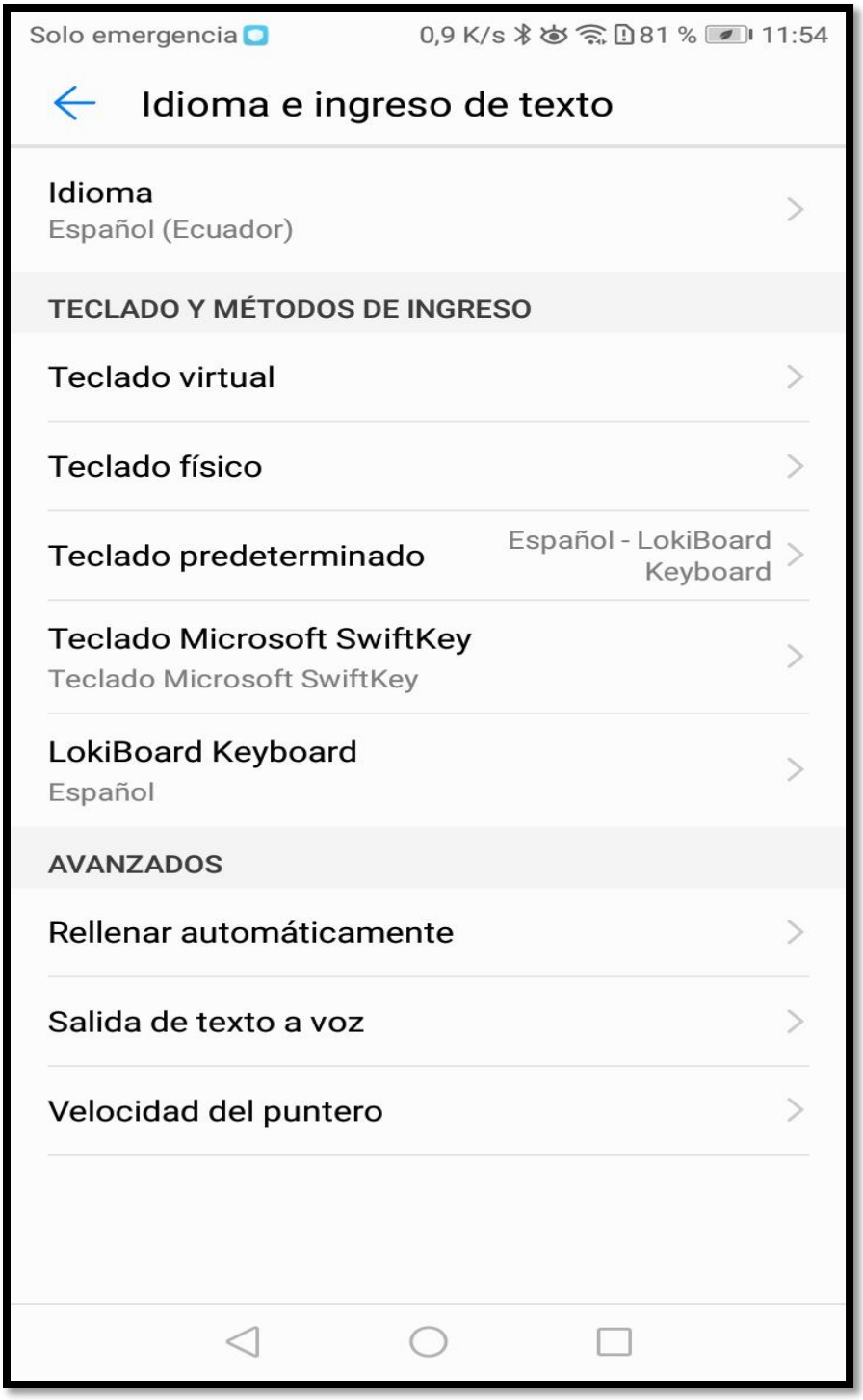


Imagen 136: Cambio exitosos del teclado

13. Realizamos la primera captura al insertar en el cuadro de búsqueda de las aplicaciones

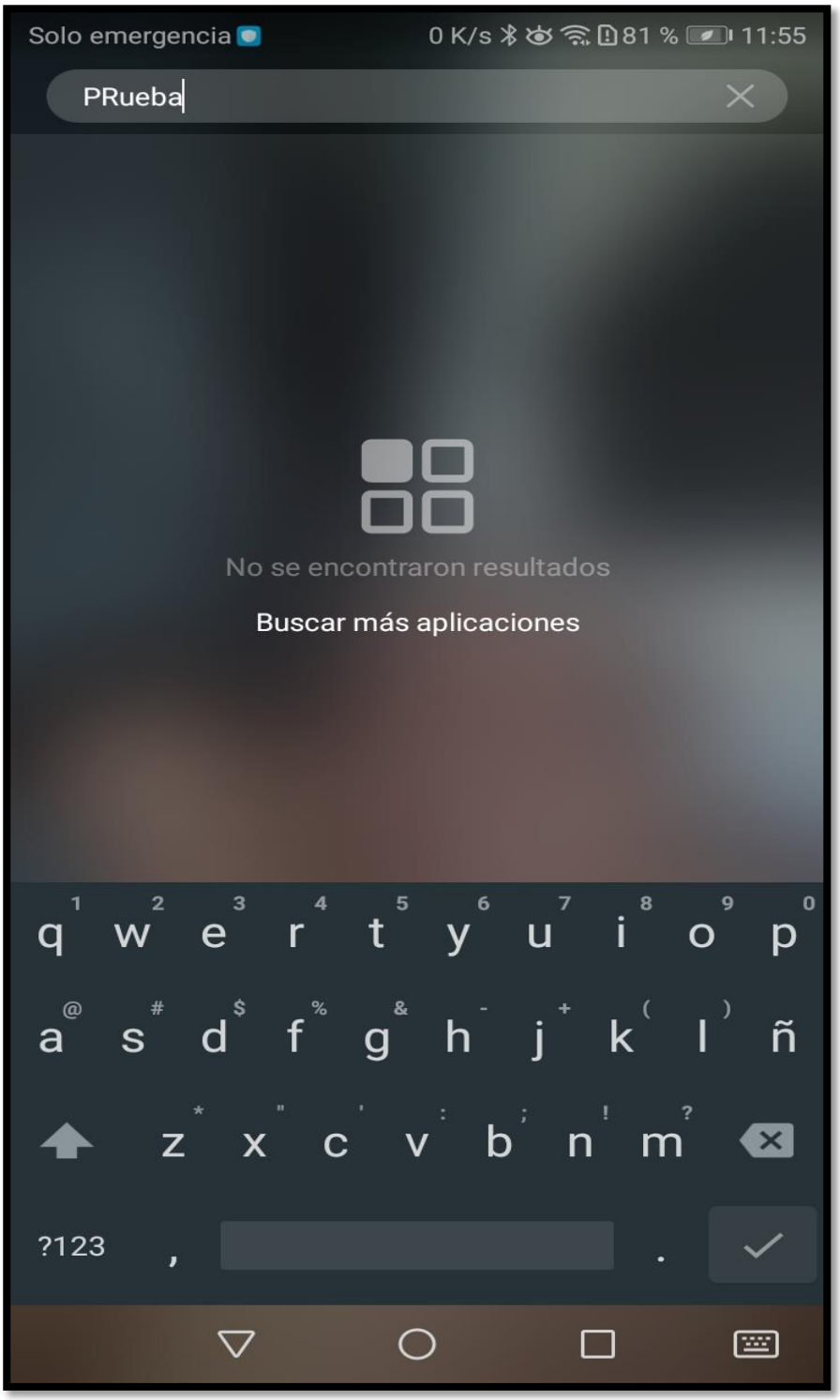


Imagen 137: Prueba demostrativa

14. En la aplicación archivos se busca el package de la aplicación en la ruta “Android/data” el com.abifog.lokiboard – files – lokiboard-files.txt.

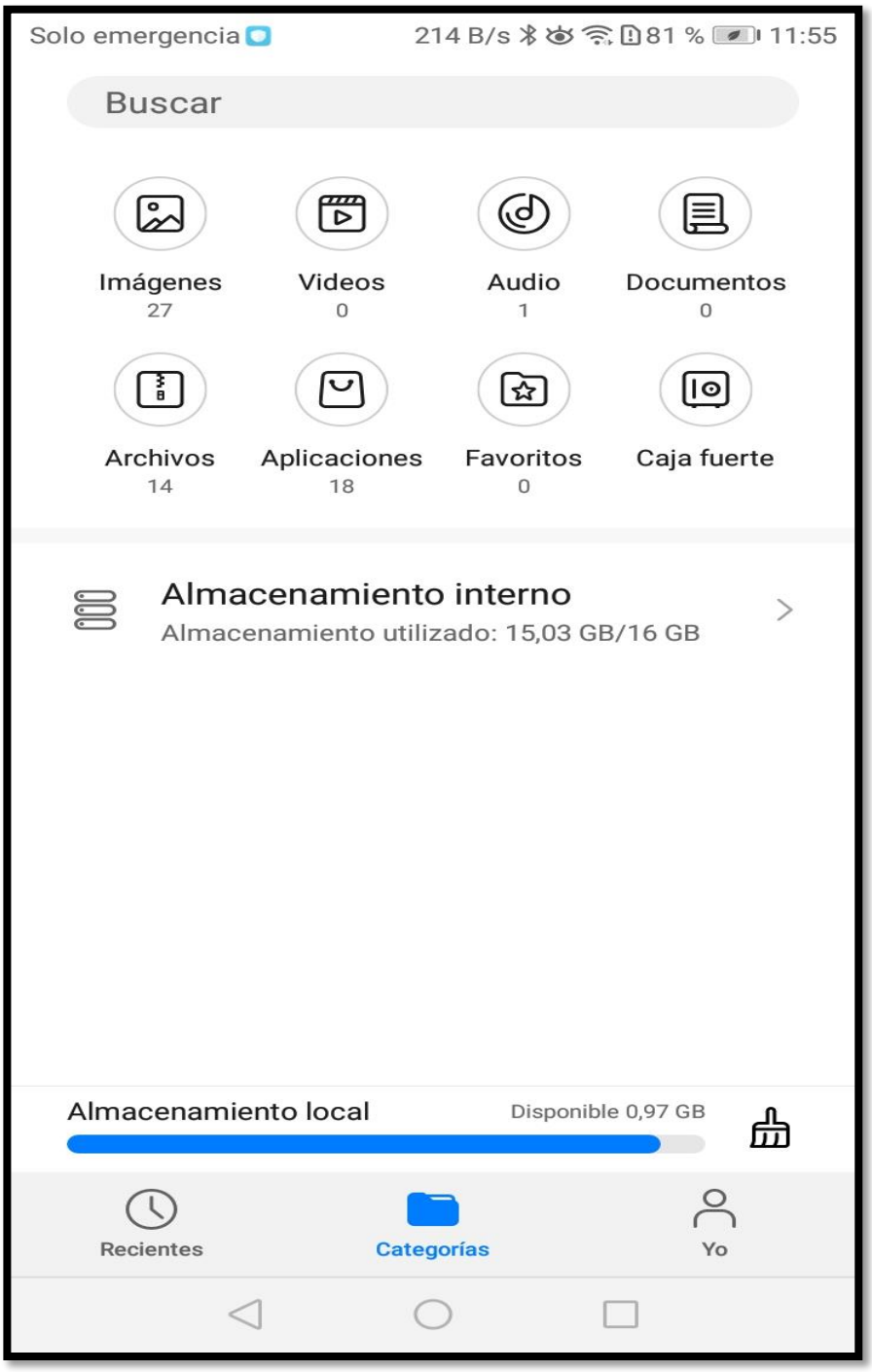


Imagen 138: Buscar El file.txt del almacenamiento Interno

15. Se encuentra el packages de la aplicación “com.abifog.lokiboard”

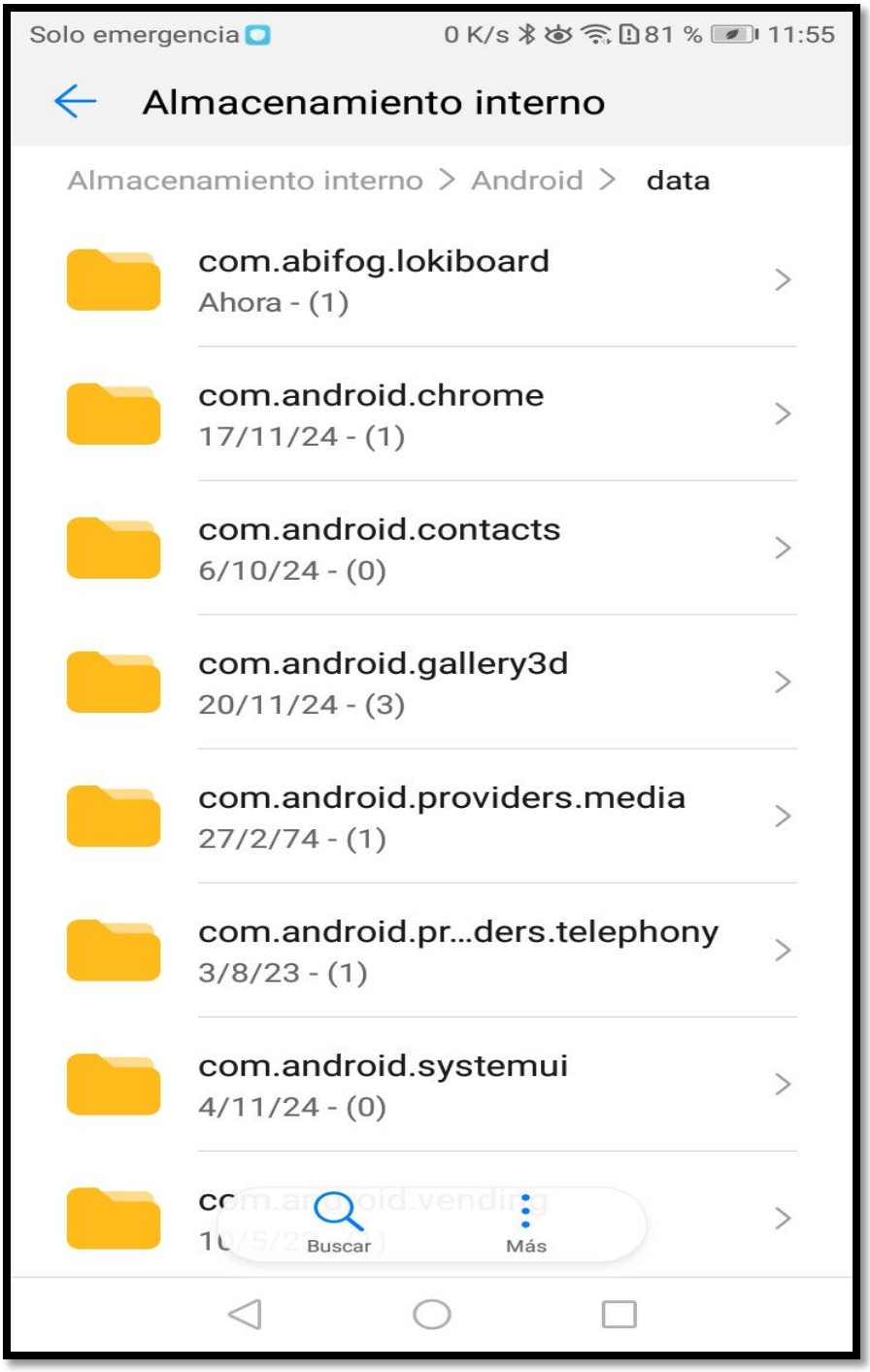


Imagen 139: Package de Lokiboard – com.abifog.lokiboard

16. Se halla en el archivo de almacenamiento lo tecleado por el usuario.

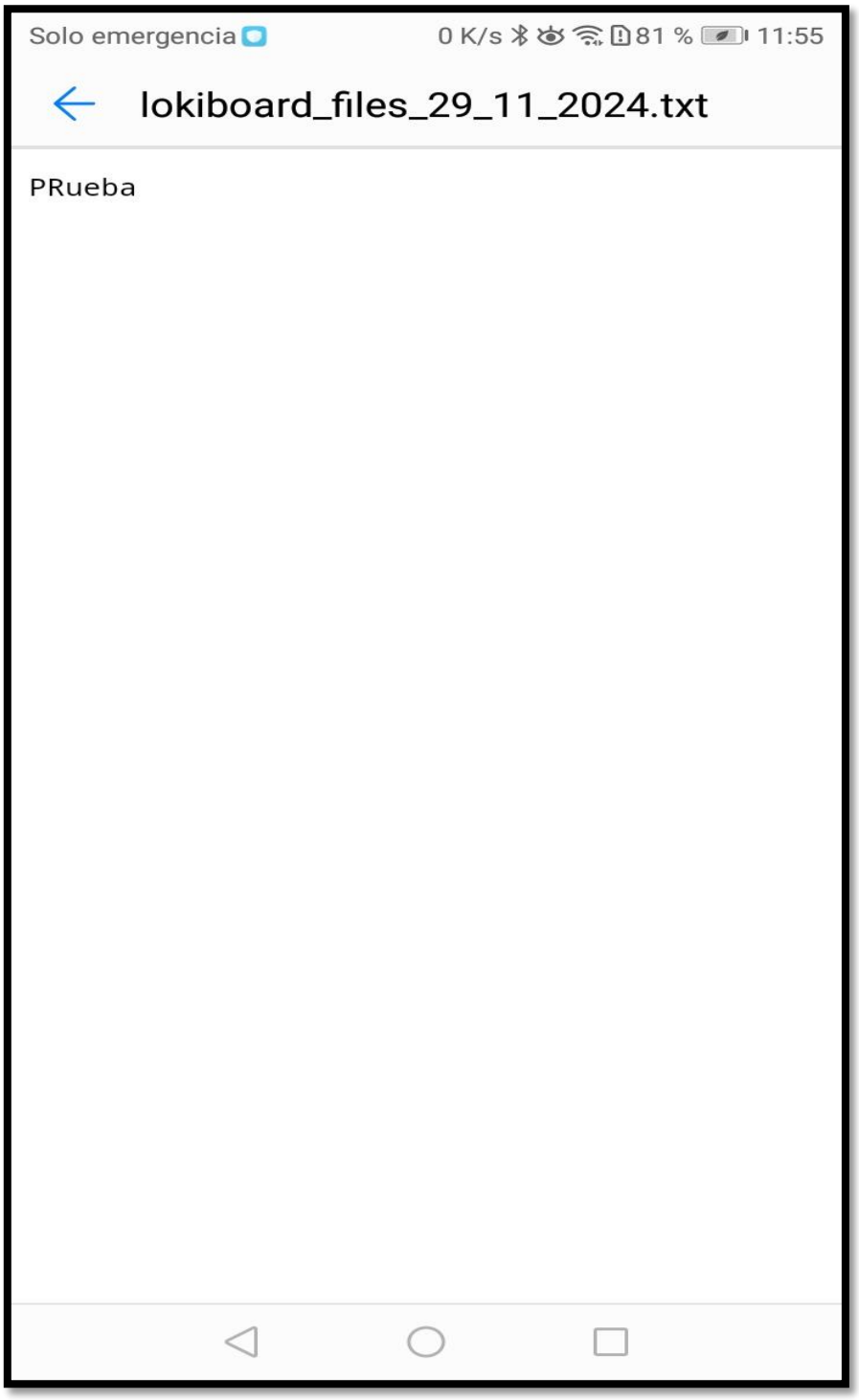


Imagen 140: Captura exitosamente

17. Se procede a ejercer un envió de información a un persona X para conocer que información se almacena en el archivo txt

Imagen 141: Prueba dos – sms a otro usuario

18. Se observa que guardar información relevante precisamente que cualquier atacante informático puede indagar a través de una persistencia o con el simple hecho de indagar los packages de las aplicaciones.

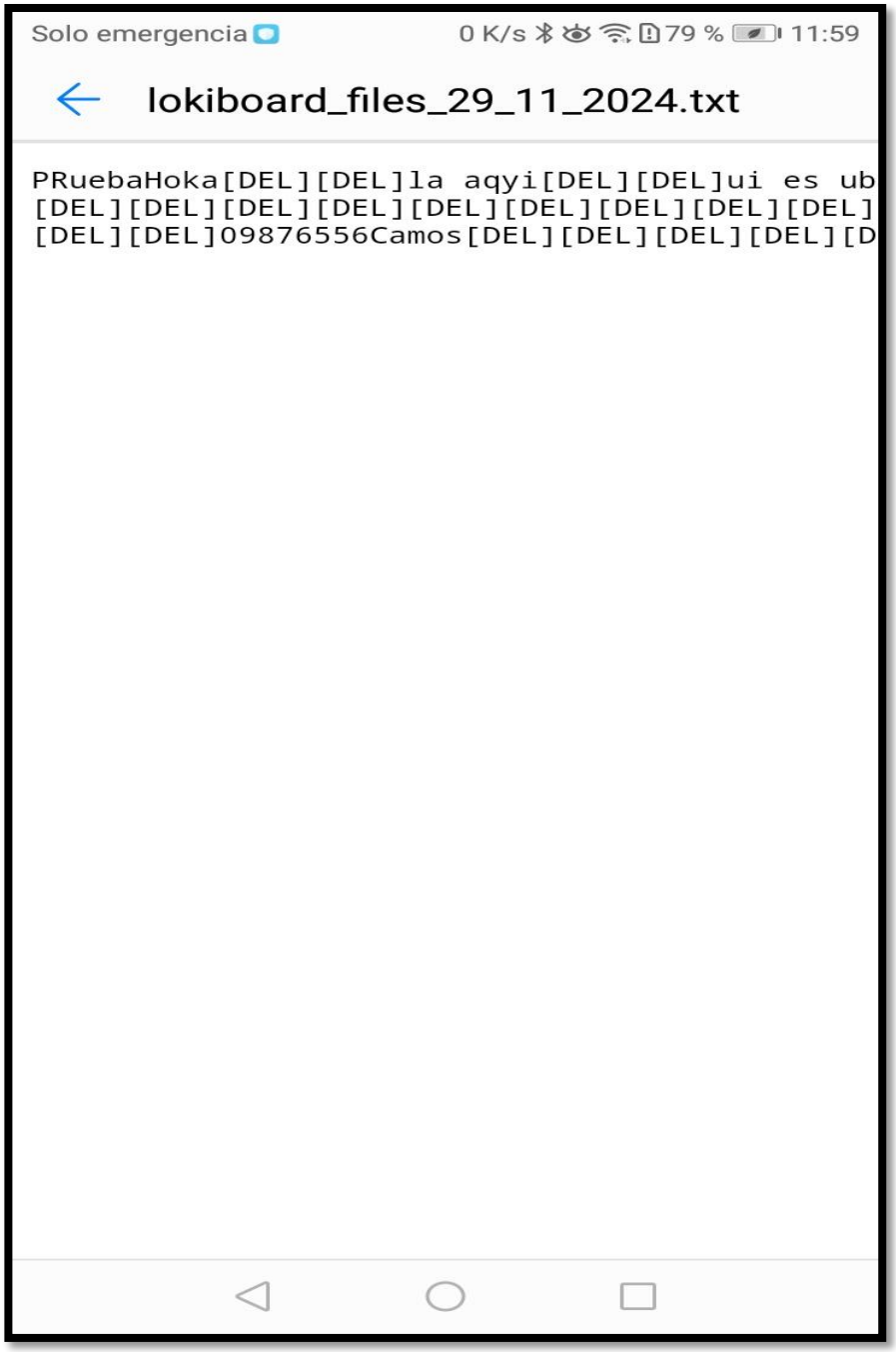


Imagen 142: Captura exitosamente en varias situaciones

ANEXO #6
FASE V: REPORTE

REPORTE#1: ANALISIS Y RESULTADO DE PRUEBA.APK – PERSISTENCIA

Análisis de Malware Rat de la Aplicación Prueba Apk			
REPORTE DE EVALUACIÓN DE VULNERABILIDADES EN DISPOSITIVO ANDROID			
DATOS DEL EXPERIMENTO			
TÍTULO DEL EXPERIMENTO:	Malware Virus – Prueba.apk	**REALIZO POR:**	Arelys Chancay
NO. PRUEBA:	01	**FECHA INICIO**	21/10/2024
TIPO DE PRUEBA:	Explotación	**FECHA FIN:**	10/11/2024
DETALLES DEL EXPERIMENTO			
OBJETIVO DEL EXPERIMENTO	Elevar privilegios para evadir seguridad y encontrar activos informáticos	**FASE:**	Pruebas de Escenarios
NIVEL DE COMPLEJIDAD PRUEBA:	Media	**TIEMPO EJECUCIÓN**	3 semanas
VULNERABILIDAD A EXPLOTAR	Reverse_TCP	**TÉCNICA HACKING**	Desarrollo de malware
HERRAMIENTAS APLICADAS			
HARDWARE	Android – Móvil	**VIRTUALIZACIÓN**	Kali Linux
SOFTWARE	Msfvenom. Msfconsole	**REDES**	Adaptador Puente , Red Movil
EVALUACION DE NIVEL DE SEGURIDAD			

- Análisis de Riesgos: El dispositivo Android presenta vulnerabilidades significativas que permiten la ejecución de un payload malicioso, lo que compromete la privacidad y seguridad del usuario.
- Impacto de la Vulnerabilidad: La explotación de la vulnerabilidad Reverse_TCP en combinación con la técnica de desarrollo de malware puede dar acceso total al dispositivo, lo que permite la exfiltración de datos sensibles, como contactos, mensajes, imágenes y grabaciones.
- Niveles de protección: Los sistemas de seguridad del dispositivo, como las restricciones de acceso y la verificación de aplicaciones, no son suficientes para prevenir este tipo de ataques, especialmente cuando el usuario permite la instalación de aplicaciones de fuentes no confiables.

- Contramedidas aplicadas: El uso de herramientas de ofuscación para evitar la detección del malware puede hacer más difícil la identificación de la amenaza, aumentando la posibilidad de éxito en el ataque.
- Recomendaciones de mitigación: Es crucial que los dispositivos Android cuenten con mecanismos de protección más robustos, como sistemas de autenticación de múltiples factores y políticas de restricción más estrictas sobre la instalación de aplicaciones de fuentes no confiables.

DISEÑO EXPERIMENTO

PROCEDIMIENTOS:

- Utilizas msfvenom para generar un payload malicioso que ejecute una acción específica en el dispositivo Android.
- Compilas el payload en un archivo APK, integrando el código malicioso dentro de la aplicación.
- Si es necesario, aplicas técnicas de ofuscación para evitar que herramientas de seguridad detecten el archivo APK como malicioso.
- Luego, firmas digitalmente el APK con un certificado, permitiendo que pueda ser instalado en el dispositivo Android.
- Distribuyes el archivo APK a los usuarios, generalmente a través de enlaces maliciosos o aplicaciones falsas.
- El usuario instala el APK en su dispositivo, habilitando la opción de instalar aplicaciones desde fuentes desconocidas.
- Una vez instalado, el payload se ejecuta en segundo plano, dando al atacante acceso a los datos del dispositivo o control sobre el mismo.

RESULTADOS ESPERADOS	**RESULTADOS OBTENIDOS**
• Romper seguridad del Android • Acceder al dispositivo por console • Verificar carpetas y raíz de usuario	• Lista de contactos • Lista de mensajes • Descargar imagines • Grabar audios

CONCLUSIONES

- El experimento confirma que los dispositivos Android son vulnerables a la explotación mediante payloads maliciosos, como el Reverse_TCP, que permiten a un atacante obtener acceso completo al dispositivo.
- La falta de medidas de seguridad adicionales en muchos dispositivos, como la falta de protección contra la instalación de aplicaciones de fuentes no verificadas, facilita la infección por malware.
- El ataque demuestra cómo los usuarios pueden estar expuestos a riesgos significativos, incluso con un sistema operativo como Android, que, aunque robusto, sigue siendo susceptible a técnicas avanzadas de hacking.

RECOMENDACIONES

- Educación del usuario: Se recomienda que los usuarios de Android estén conscientes de los riesgos asociados con la instalación de aplicaciones desde fuentes desconocidas y desactiven esta opción cuando no sea necesario.
- Refuerzo de políticas de seguridad: Es importante que las empresas y organizaciones implementen políticas de seguridad más estrictas en dispositivos móviles, como el uso de

soluciones de administración de dispositivos móviles (MDM) y el control de aplicaciones instaladas. • Actualizaciones constantes: Asegurarse de que los dispositivos reciban actualizaciones periódicas del sistema operativo y de las aplicaciones, para corregir vulnerabilidades conocidas. • Uso de software de seguridad: Implementar soluciones de seguridad, como antivirus y aplicaciones de protección contra malware, que puedan detectar e impedir la instalación de software malicioso en dispositivos móviles.

Tabla 7: Reporte de análisis y resultados – Prueba.apk

REPORTE#2: ANALISIS Y RESULTADO DE LOKIBOARD.APK – PERSISTENCIA

Análisis De Malware Rat del Lokiboard Apk			
REPORTE DE EVALUACIÓN DE VULNERABILIDADES EN DISPOSITIVO ANDROID			
DATOS DEL EXPERIMENTO			
TÍTULO DEL EXPERIMENTO:	Malware Virus – Lokiboard.apk	**REALIZO POR:**	Arelys Chancay
NO. PRUEBA:	02	**FECHA INICIO**	11/11/2024
TIPO DE PRUEBA:	Explotación	**FECHA FIN:**	27/11/2024
DETALLES DEL EXPERIMENTO			
OBJETIVO DEL EXPERIMENTO	Elevar privilegios para evadir seguridad y encontrar activos informáticos	**FASE:**	Pruebas de Escenarios
NIVEL DE COMPLEJIDAD PRUEBA:	Medio	**TIEMPO EJECUCIÓN**	2 semanas
VULNERABILIDAD A EXPLOTAR	Acceso no autorizado a entradas del teclado (keylogging)	**TÉCNICA HACKING**	Uso de malware tipo Keylogger (Lokiboard)

HERRAMIENTAS APLICADAS			
HARDWARE	Android – Móvil	**VIRTUALIZACIÓN**	Kali Linux
SOFTWARE	Lokiboard Keyboard	**REDES**	Adaptador Puente , Red Móvil

EVALUACION DE NIVEL DE SEGURIDAD

- Análisis de Riesgos: La instalación de un teclado malicioso como Lokiboard permite al atacante registrar las pulsaciones de teclas sin el conocimiento del usuario, lo que expone a la víctima a la fuga de información sensible como contraseñas y datos personales.
- Impacto de la Vulnerabilidad: La ejecución del keylogger en segundo plano facilita el acceso no autorizado a datos privados, lo que puede resultar en robo de credenciales, acceso a cuentas bancarias y otros riesgos de privacidad.
- Niveles de Protección: La seguridad del dispositivo no es suficiente para detectar este tipo de malware si el usuario permite la instalación de aplicaciones desde fuentes desconocidas. Además, el malware se ejecuta en segundo plano y se oculta en las configuraciones del sistema.
- Contramedidas Aplicadas: El malware se oculta en el sistema de entrada, lo que dificulta la detección. Es necesario implementar medidas de protección como sistemas de seguridad adicionales y la verificación de permisos de aplicaciones para evitar esta amenaza.

DISEÑO EXPERIMENTO

PROCEDIMIENTOS:

- Compilación del APK: El malware Lokiboard se compila utilizando el código fuente o descargando un APK preconstruido.
- Instalación en el dispositivo: Se instala el APK en el dispositivo Android después de habilitar la opción de instalación desde fuentes desconocidas.
- Activación del Keylogger: En el dispositivo, se abre la sección de "Idioma y entrada" en la configuración del sistema y se habilita Lokiboard como el teclado predeterminado, deshabilitando todos los demás métodos de entrada.
- Acceso a los registros de pulsaciones: Una vez que el keylogger está activo, las pulsaciones de teclas se registran en un archivo dentro del almacenamiento interno del dispositivo (Ruta: Internal Storage > Android > Data > com.abifog.lokiboard > files > lokiboard-files.txt).
- Obtención de datos: El atacante puede acceder al archivo de registro de pulsaciones de teclas a través de un administrador de archivos.

RESULTADOS ESPERADOS	**RESULTADOS OBTENIDOS**
• Captura de las pulsaciones de teclas • Acceso a información sensible • Obtención de registros del Keylogger	• Lista de pulsaciones de teclados • Acceso a datos sensibles • Archivos de registro
CONCLUSIONES	
• Confirmación de la vulnerabilidad: La explotación de la vulnerabilidad de acceso a las entradas de teclado a través de aplicaciones maliciosas como Lokiboard demuestra que los dispositivos Android siguen siendo susceptibles a ataques de keylogging. • Eficiencia del malware: Lokiboard es efectivo para registrar pulsaciones de teclas sin ser detectado, lo que implica una alta probabilidad de filtración de datos sensibles. • Limitaciones de seguridad: La falta de restricciones más estrictas en el sistema operativo, como la verificación de aplicaciones en segundo plano y el monitoreo de teclados predeterminados, aumenta la vulnerabilidad del dispositivo.	
RECOMENDACIONES	
• Educación del usuario: Es fundamental educar a los usuarios sobre los peligros de instalar aplicaciones de fuentes desconocidas y la importancia de verificar los permisos antes de permitir la instalación de cualquier aplicación. • Refuerzo de seguridad del sistema operativo: Android debe implementar medidas de seguridad adicionales, como detección de aplicaciones maliciosas que alteren el sistema de entrada, y advertencias sobre cambios en los teclados predeterminados. • Control de permisos de aplicaciones: Se recomienda usar soluciones de seguridad que bloqueen aplicaciones de keylogging o que permitan el monitoreo de actividad de teclados para prevenir la fuga de datos. • Actualizaciones y parches de seguridad: Mantener los dispositivos actualizados con las últimas versiones de Android y parches de seguridad puede ayudar a mitigar vulnerabilidades conocidas.	

Tabla 8: Reporte de análisis y resultados – Lokiboard_Keyboard.apk