PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

TRABAJO DE TITULACIÓN

“ATAQUE DE DICCIONARIO MEDIANTE EL ANÁLISIS Y RECOLECCION DE DATOS DE LAS REDES SOCIALES”

AUTOR:

YANEZ TAPIA ARIEL FERNANDO

DIRECTOR:

GUAÑA MOYA EDISON JAVIER

QUITO DM, 2023

DEDICATORIA

Primero, agradezco infinitamente a Dios por ser mi guía y fortaleza en cada paso de este viaje. Su luz me ha acompañado en los momentos más desafiantes, iluminando mi camino hacia el conocimiento y la sabiduría.

A mi querida madre, Irina Tapia, cuyo amor, sacrificio y apoyo incondicional han sido el faro que me guio a través de esta travesía. Tus enseñanzas y tu fuerza han sido mi mayor inspiración.

A mi padre, Diego Yáñez, por su constante estímulo y por creer en mí incluso cuando dudaba de mis propias capacidades. Tu apoyo y sabiduría han sido fundamentales en mi formación académica y personal.

Esta tesis es el fruto de sus enseñanzas, amor y fe en mí. A todos ustedes, mi más profundo agradecimiento.

AGRADECIMIENTO

Quisiera expresar mi más profundo agradecimiento a quienes han sido pilares fundamentales en este significativo viaje académico.

A mi amada madre, Irina Tapia, cuyo amor incondicional, apoyo y sacrificios han sido la base sobre la cual he construido mis sueños. Tu fuerza y dedicación son el ejemplo que me guía cada día.

A mi padre, Diego Yáñez, por su constante aliento y apoyo en cada etapa de mi educación. Tu creencia en mis capacidades ha sido un regalo invaluable que me ha impulsado a alcanzar mis metas.

También extiendo mi gratitud a mi tutor, Javier Guaña, por su invaluable ayuda y orientación durante la realización de esta tesis. Su sabiduría, paciencia y guía han sido esenciales en la culminación exitosa de este proyecto.

A todos ustedes, gracias por su amor, apoyo y confianza. Este logro es también suyo.

RESUMEN

El presente trabajo plantea una metodología general a seguir para realizar un ataque de diccionario dirigido, de forma controlada ya que no se puede vulnerar ni realizar accesos no autorizados a una cuenta se lo realiza un ataque a un perfil propio para evitar cualquier tipo incumplimientos de la ley, esto se lo hace con un único fin educativo y académico.

El ataque realizado se divide en cuatro fases que van acorde a la metodología planteada, en las cuales consta una primera fase de extracción de información en la cual el atacante busca las redes sociales de la víctima y realiza capturas de la información que considere relevante para ingresarlas en un script desarrollado en Python que trascribe el texto de una fotografía y la almacena en un repositorio.

Una segunda fase para depurar la información que se encuentra ya transcrita en archivos de texto para eliminar las palabras que no sean relevantes en la creación de la contraseña, la tercera fase es la de la creación de los diccionarios de datos utilizando la información que se encuentra en las redes sociales de Facebook, Instagram y X(Twitter) con la ayuda de herramientas de una distribución libre de Linux “Kali Linux” como Cupp y Crunch para la creación de diccionarios de contraseñas.

Y por último la cuarta fase que consiste en comprobar las contraseñas generadas para lo cual se utilizó la herramienta de Hydra que permite realizar ataques de fuerza bruta, pero con las nuevas seguridades implementadas por los destinos sitios web solo se puede comprobar 20 contraseñas en un lapso de una hora para que los servidores no bloquen la conexión que se tiene e inhabilite poder seguir intentando más contraseñas.

INDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPITULO I: INTRODUCCION

1. Marco de referencia

1.1. Tema

Ataque de diccionario mediante el análisis y recolección de datos de las redes sociales.

1.2.Justificación

La información personal puede ser utilizada por ciberdelincuentes para cometer delitos financieros o de identidad, como el robo de contraseñas, la clonación de tarjetas de crédito y la suplantación de identidad. Además, la información personal en las redes sociales puede ser recopilada y utilizada por empresas para dirigir anuncios publicitarios específicos y para crear perfiles detallados de los usuarios, lo que puede afectar su privacidad y seguridad en línea.

Por lo tanto, es crucial que los usuarios protejan su información personal en las redes sociales configurando adecuadamente la privacidad y la seguridad de sus cuentas, no compartiendo información personal sensible en línea y evitando interactuar con usuarios desconocidos o sospechosos. También deben tener precaución al hacer clic en enlaces o descargar archivos de fuentes no confiables en las redes sociales. Esta vulnerabilidad encontrada sobre la información personal de los usuarios dentro de las redes sociales permite recopilar y analizar sus datos personales con el fin de ensamblar esa información en diccionarios y tener posibles contraseñas de los usuarios combinando la información extraída de sus redes sociales.

1.3.Planteamiento del problema

El avance de la tecnología conlleva el desarrollo de nuevas aplicaciones como lo son las redes sociales donde las personas pueden interactuar y compartir datos a través del internet, pero no todos los usuarios de las redes sociales realizan publicaciones de manera consciente tomando en cuento los riesgos que puede conllevar compartir su información personal en este tipo de aplicaciones.

Con esta vulnerabilidad presente se pueden realizar diferentes tipos de ataques, en este caso se opta por realizar un ataque de diccionario con el fin de dar a conocer al usuario atacado que el compartir su información personal por estos medios digitales los hace blancos de ataques y en su gran mayoría con fines malísimos y perjudiciales.

Esta problemática da como resultado la siguiente pregunta inicial de investigación:

- ¿Como se puede recopilar la información personal compartida por los usuarios en de las redes sociales para generar un diccionario con los datos obtenidos?
- Y las siguientes preguntas secundarias:
- ¿Las contraseñas de los usuarios en las redes sociales están compuestas por su información personal?
- ¿Los usuarios de las redes sociales conocen los riesgos sobre compartir su información personal en redes sociales?
- ¿Se puede realizar diccionarios de datos con la información personal de las redes sociales?

1.4.Objetivo general

Realizar un ataque de diccionario mediante herramientas Open Source utilizando datos de diferentes redes sociales de un usuario.

1.5.Objetivo especifico

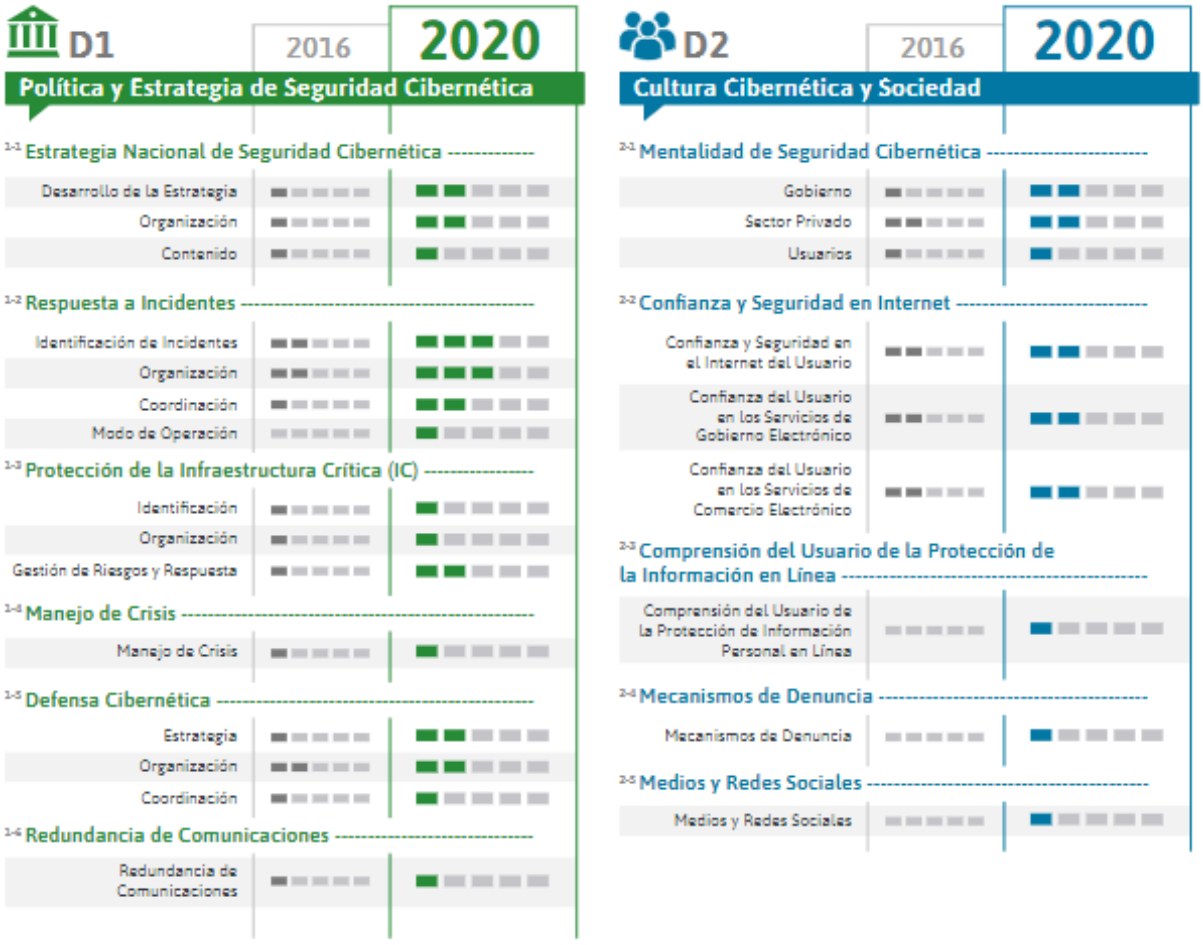
- Analizar diferentes aplicaciones Open Source que son utilizadas para la generación de ataques de diccionario.
- Recopilar datos de un usuario en redes sociales para la generación de una base de datos con la información obtenida.
- Realizar combinaciones con la información compilada en los diccionarios para generar posibles claves de usuario.
- Implementar un ataque de diccionario con la información recopilada de diferentes redes sociales.

1.6.Antecedentes

Aunque Ecuador todavía no ha desarrollado una estrategia integral de seguridad cibernética, ha realizado progresos notables en el fortalecimiento de sus habilidades en este campo y en la gestión de riesgos relacionados como se muestra en los indicadores del reporte de "Ciberseguridad riesgos, avances y el camino a seguir en américa latina y el caribe" realizado por el Banco Interamericano de Desarrollo y la Organización de Estados Americanos. En el cual contrasta, la política y estrategia de seguridad y la cultura cibernéticas y sociedad, los avances logrados del año 2016 al año 2020 (Ver figura 1).

Figura 1

Política y estrategia de seguridad, cultura y sociedad cibernética.



Nota. La figura muestra el crecimiento cibernético del Ecuador en aspectos de política, estrategia, cultura y sociedad entre los años 2016 y 2020. Adaptado de *Riesgos, avances y el camino a seguir en América latina y el caribe,* de BID y OEA, 2020, Reporte Ciberseguridad 2020.(https://publications.iadb.org/publications/spanish/viewer/Reporte-Ciberseguridad-2020-riesgos-avances-y-el-camino-a-seguir-en-America-Latina-y-el-Caribe.pdf). CC-IGO 3.0

Como se puede apreciar en la Figura 1 la compresión del usuario de la protección de información en línea en el Ecuador en 2016 es nula mientras que en el año 2020 apenas a tendió un crecimiento mínimo lo que denota que los usuarios no contemplan una seguridad de sus datos al momento de ponerlos en internet como en redes sociales, blogs, páginas web, etc.

El uso de una misma contraseña para varios sitios es una mala práctica que se lleva a cabo por muchos usuarios que navegan por internet como lo muestra el informe "Psicología de las contraseñas" realizado por LastPass en el año 2022, el cual muestra un estudio realizado a 3750 profesionales para conocer sus prácticas en el campo de las contraseñas, en el que se evidencio que el 12% de los encuestados admitieron que usan contraseñas únicas para cada sitio web, mientras que el 89% usa contraseñas similares o reutilizadas, representando que es un riesgo para la seguridad.

En el último año, la cantidad de personas que utilizan redes sociales aumentó en 227 millones, llegando a 4,700 millones a principios de julio de 2022. Esto representa un incremento de más del 5% en el uso de redes sociales a nivel mundial en los últimos 12 meses. Actualmente, esta cifra equivale al 59% de la población total del planeta. (Kemp, 2022)

Personas jóvenes de 16 a 34 años revelan más del 60% de su información personal y más del 42% de sus detalles financieros. En contraste, individuos mayores de 55 años comparten solo alrededor del 38% de su información personal, indicando una mayor precaución. Esto sugiere que la población más joven es más vulnerable a ataques cibernéticos. (Semymas, 2019)

Con estos antecedentes el trabajo realizado se basa en extraer información que se encuentra de manera pública en las redes sociales, para realizar un ataque de diccionario personalizado, que en la mayoría de los casos es la misma información la que compone la contraseña en uno o varios sititos webs del usuario.

1.7.Alcance

El trabajo de desarrollo tiene como alcance el recopilar la información de usuarios de las redes sociales Facebook, Twitter e Instagram usando técnicas similares al Web Scraping con el fin de crear un diccionario de cada usuario y realizar combinaciones con cada dato, ataque de diccionario personalizado, con el fin de obtener una posible combinación de acuerdo con los requerimientos de cada red social al crear una contraseña (número de caracteres, mayúsculas, minúsculas, números, caracteres). Se usará estos parámetros para poder realizar las combinaciones una vez obtenidos los datos.

El número de usuarios que se utilizaran para realizar el ataque será limitado por el rendimiento, procesamiento y almacenamiento con el que el CPU que se utilizara lo permita donde se tendrá las herramientas y los sistemas operativos a utilizar que deberán trabajar en conjunto para realizar el ataque de diccionario, junto con las limitaciones que las herramientas Open Source que se utilizaran lo permitan.

El trabajo de titulación se centra en extraer información personal que sea encuentre en redes sociales y sea de acceso público, no se trabajara con información privada que el usuario tenga dentro de sus redes sociales, ya que todas las aplicaciones escogidas alertan al usuario cunado desean subir sus publicaciones o su información que la misma será de libre acceso en el internet y no se responsabilizan por el uso que se le dé.

1.8.Metodología

Para llevar a cabo el desarrollo del trabajo se utilizó una metodología investigativa, ya que el sentido es la búsqueda de las causas o razones de la problemática planteada, respondiendo a cómo podría suceder y por qué podría suceder las situaciones que ocurrieron.

El trabajo parte con la finalidad de comprobar que se pueden identificar la contraseña de una red social realizando un ataque de diccionario con la información que las personas proporcionan en sus redes sociales, si bien no existe una metodología estándar para realizar este tipo de ataque, se basara en las metodologías utilizadas en ataques de diccionario realizados y en técnicas de extracción o raspado de información pública de libre acceso.

Se toma en consideración los recursos de hardware disponibles para la elaboración del trabajo teniendo en cuenta que de ser insuficientes se realizaran cambios en la metodología planteada, así como las limitaciones de las herramientas a utilizar y las políticas de seguridad con las que las redes sociales trabajan, se deberán aplicar cambios con el fin de cumplir el objetivo propuesto en este trabajo.

Para la obtención de información se optó por la técnica de Web Scraping que se la puede aplicar en Python con la ayuda de sus librerías, sin embargo al encontrar que las redes sociales ya cuentan con métodos de protección contra este tipo de técnicas se utilizara un código realizado en Python para que convierta a texto las palabras u oraciones que sean encontradas en una imagen, esto con el fin de crear un repositorio de imágenes de la información del perfil de cada usuario para convertir a texto.

Se examinarán los resultados de textos extraídos de las imágenes para posteriormente analizar qué datos deberán ser ingresados a las aplicaciones que permitirá crear los diccionarios de datos, para su posterior prueba y comprobación de la hipótesis plateada en el trabajo.

CAPITULO II: FUNDAMENTACION TEORICA

2. Marco teórico

La seguridad cibernética se refiere al conjunto de estrategias y prácticas implementadas para salvaguardar redes, sistemas y datos en el ámbito digital. Es una pieza clave en la era de la digitalización, ya que su rol es fundamental para preservar la confidencialidad, integridad y accesibilidad de la información. Esta disciplina utiliza un marco de procedimientos y políticas específicas que buscan mantener la solidez funcional de los sistemas informáticos, protegiéndolos así contra accesos no autorizados y ataques que buscan explotar sus vulnerabilidades.

Aunque a menudo se confunden los términos seguridad de la información y ciberseguridad, cabe destacar que la seguridad de la información se enfoca en proteger la información en todas sus formas, incluyendo la impresa y la transmitida verbal o visualmente. Por otro lado, la ciberseguridad se especializa en la protección integral de los activos digitales, abarcando la información que se procesa, se guarda o se envía a través de redes de sistemas de información, así como de la infraestructura crítica de información. (Innovación, 2022)

Estas bases son complementadas con otros principios como autenticación, no repudio y control de accesos. En la era actual, caracterizada por una creciente interconexión digital, la ciberseguridad no solo se enfoca en prevenir amenazas como programa maligno, phishing o ataques de diccionario, sino también en garantizar la resiliencia y recuperación post ataque, abordando tanto aspectos técnicos como humanos del ecosistema cibernético.

2.1.Ataques cibernéticos

Los ataques cibernéticos son acciones maliciosas llevadas a cabo por personas o grupos con el objetivo de comprometer la seguridad de sistemas y redes informáticas. Existen varios tipos de ataques, como el programa maligno, phishing, ransomware, ataques de fuerza bruta, ataques de denegación de servicio (DDoS) y ataques de diccionario, entre otros.

Los ataques cibernéticos constituyen una modalidad de hostilidad que emplea medios digitales para infligir daño, abarcando desde consecuencias no letales como pérdidas económicas hasta repercusiones potencialmente mortales por la perturbación de infraestructuras vitales. Estos ataques ocupan un punto intermedio entre el progreso tecnológico y el armamento estratégico, presentando motivaciones diversas y a menudo identidades de los agresores no transparentes, lo que complica su atribución y la reacción de las partes afectadas. La forma más extendida de agresión cibernética es el cibercrimen, el cual ha crecido considerablemente y puede provocar pérdidas financieras y de datos personales significativas. La naturaleza cambiante de las amenazas cibernéticas, incluyendo las actividades de ciberterrorismo, ha suscitado inquietudes acerca de su capacidad para paralizar servicios esenciales y menoscabar la seguridad nacional (Snider et al., 2021).

2.2.Técnicas de ataques cibernéticos

El proceso de un ataque de diccionario generalmente implica los siguientes pasos:

- Recopilación de información: El atacante puede recopilar información sobre el objetivo, como nombres de usuario, correos electrónicos, perfiles en redes sociales u otra información personal, para generar una lista de posibles contraseñas.

- Selección del diccionario: Se selecciona un diccionario de palabras o combinaciones de caracteres para realizar las pruebas. Estos diccionarios pueden ser generados por el propio atacante o estar disponibles en línea.
- Prueba de contraseñas: Utilizando un software automatizado, el atacante prueba las contraseñas del diccionario contra el sistema o la red objetivo. El software realiza una serie de intentos de inicio de sesión, probando cada contraseña hasta encontrar una que funcione o hasta agotar el diccionario.
- Éxito o fracaso: Si el ataque tiene éxito, el atacante obtiene acceso no autorizado al sistema o red objetivo. En caso de fracaso, el ataque puede intentar utilizar otros métodos o estrategias para comprometer la seguridad.

2.3.Open source

Open Source, también conocido como código abierto, se define como un modelo de desarrollo de software y tecnología en el que el código fuente está disponible y es compartido de manera pública. Este enfoque permite que los usuarios puedan ver, modificar y redistribuir el software a su antojo, promoviendo así la colaboración y la transparencia dentro de la comunidad (Red Hat, 2023).

Open Source fomenta principalmente la colaboración e innovación en el desarrollo del software, ya que al disponer del código fuente públicamente accesible le permite a cualquier persona poder estudiar, modificar, destruir y mejorar el software lo que conlleva a una mayor diversidad de ideas y soluciones contribuyendo a la transparencia, seguridad, flexibilidad en el desarrollo de software, permitiendo realizar adaptaciones y personalizaciones especificas según sean las necesidades de los usuarios.

2.4.Kali-Linux

Kali Linux es una distribución de Linux basada en Debian, creada por Offensive Security. Se caracteriza por estar específicamente diseñada para pruebas de penetración, seguridad informática y hacking ético. Como una herramienta preinstalada en el sistema operativo Linux, Kali Linux destaca por su alta seguridad y configuraciones avanzadas, y cuenta con múltiples herramientas para proteger y asegurar la información almacenada en el sistema operativo (Castro, 2023).

Kali Linux cuenta con una alta gama de herramientas de seguridad y herramientas de pruebas de penetración, siendo un principal motivo por el cual se escogió este sistema operativo junto a la compatibilidad que ofrece para lograr una interacción con otras herramientas y aplicaciones de su propio entorno o de diferentes sistemas operativos, manteniendo fuertes medidas de seguridad lo que lo hace ideal para trabajos relacionados con la ciberseguridad.

2.5.Python

Python es un lenguaje de programación orientado a objetos, interpretado y de uso general, reconocido por su código claro y estructurado. Se distingue por su licencia de código abierto, lo que permite su uso gratuito en una variedad de aplicaciones. Este lenguaje es ampliamente utilizado en el desarrollo web, la informática científica, el análisis de datos y la automatización de tareas, y es conocido por su versatilidad, capacidad multiplataforma y soporte para múltiples paradigmas de programación. Python es empleado en plataformas de alto tráfico como Google, YouTube o Facebook, destacándose por su eficiencia y facilidad de aprendizaje (ORACLE, 2022).

Las características clave por las cuales se tomó en cuenta a Python en el uso de este trabajo es por su versatilidad al poder usarse en una gran variedad de aplicaciones, su amplia comunidad

lo que aporta una gran cantidad de recursos, bibliotecas y frameworks disponibles para su uso, así como la integración que esta aplicación ofrece al integrarse con otros lenguajes de programación y tecnologías.

2.6.Github

GitHub es una plataforma en línea y un servicio de nube que facilita a los desarrolladores el almacenamiento y la gestión de su código. Utiliza el sistema de control de versiones Git, lo que permite un seguimiento y control detallado de cualquier cambio realizado en el código. Además, GitHub se destaca como una plataforma de desarrollo colaborativo, donde personas de todo el mundo pueden trabajar conjuntamente en proyectos de código fuente para programas de computadora. (KINSTA, 2020).

La herramienta GitHub se integra con una variedad de herramientas y servicios que permiten mejorar el flujo de trabajo en el desarrollo de software el cual después de ser concluido la misma herramienta de GitHub proporciona un lugar seguro y confiable, en el cual se tiene un control de versiones, seguimiento de problemas y una documentación del proyecto para mejorar la calidad del software, el cual es accesible desde cualquier lugar.

2.7.Web Scraping

El web scraping, o raspado web, es un proceso utilizado para recolectar datos de sitios web y guardarlos para investigaciones futuras o su preservación a largo plazo. A diferencia del archivado web, que busca preservar la apariencia y estructura de los sitios web, el web scraping se enfoca principalmente en la recolección de datos textuales. Las herramientas de web scraping permiten estructurar los datos recopilados, transformándolos de archivos de texto masivos y desestructurados a formatos como hojas de cálculo, CSV o bases de datos, facilitando su análisis

y uso en investigaciones. Este proceso se utiliza en una variedad de aplicaciones, desde la investigación de mercado y precios por parte de empresas, hasta el seguimiento de información meteorológica y datos inmobiliarios. Además, los investigadores utilizan el web scraping para estudios en foros web o redes sociales como Twitter y Facebook, así como para el monitoreo de cambios en páginas web a lo largo del tiempo (Research Data Services, 2019).

Usando la técnica de Web Scraping se logra una recopilación eficiente de datos ya que es un recolección automatizada de grandes cantidades de datos de la web de manera rápida y eficiente que posteriormente usualmente se los puede utilizar para estudios de mercado, análisis de precios o como en este caso investigación y fines académicos, los resultados de datos obtenidos se los puede presentar como datos estructurados o no estructurados y analizables en CVS, hojas de cálculos, TXT o en bases de datos.

Los derechos de autor y de propiedad intelectual no siempre se encuentran bien definidos en relación a si los datos extraídos de un sitio web infringen los derechos de autor o las leyes de propiedad intelectual mucho más al utilizar esta técnica en redes sociales donde la información que proporciona cada usuario a su perfil es de libre acceso al estar en la web, por estos inconvenientes algunas redes sociales prohíben explícitamente el scrpaing en sus términos de servicio y otras han implementado herramientas para evitar que se realicen este tipo de técnicas de forma automática y masiva.

2.8.Redes sociales

En el inicio del milenio, emergió la noción de Web 2.0, vinculada al avance de portales web que facilitan la interacción y cooperación entre usuarios, además de permitirles generar su propio contendió, esta capacidad de participación directa ha hecho que la Web 2.0 sea denominada web participativa y bajo estos cambios los internautas comienzan a involucrarse en comunidades

de discusión, a narrar y divulgar sus vivencias de adquisiciones y a ofrecer sugerencias a otros internautas de la red (Piñero et al., 2021).

Con el avance en la tecnología se ha podido presenciar el desarrollo de las aplicaciones para la comunicación global entre personas y la difusión en tiempo real de información, logrando que con estas características se desarrollen aplicaciones específicas en las cuales las personas puedan compartir la información que deseen y esta sea publica de libre acceso para los usuarios de internet.

2.9.Contraseña

Las contraseñas son una forma de seguridad de la información que exige a los usuarios cumplir con políticas que a menudo requieren la creación de contraseñas largas y complejas, que pueden ser difíciles de recordar, pero son esenciales para proteger contra el acceso no autorizado. La fortaleza de una contraseña a menudo se mide por su longitud, complejidad e imprevisibilidad. Sin embargo, las políticas de contraseñas muy estrictas pueden llevar a una seguridad debilitada si los usuarios eligen contraseñas fáciles de recordar o las almacenan de manera insegura, dejándolas vulnerables al robo (Trepp, 2021).

Una forma efectiva de fortalecer una contraseña es usar frases largas, que, incluso llegando a colocar el hash de la contraseña valido, se volvería difícil de adivinar para una computadora de cracking, de igual forma el uso de un administrador de contraseñas que genere cadenas largas y pseudoaleatorias de caracteres en conjunto con autentificadores de múltiples factores.

Los atacantes tienen numerosas herramientas para descifrar las contraseñas, como la interceptación de credenciales y la explotación de protocolos de permisos, sin embargo, la búsqueda y captura de contraseñas por parte de atacantes puede resultar sorprendentemente

exitosa, especialmente cuando las credenciales están debidamente protegidas y almacenadas en lugares accesibles o en teto claro (Trepp, 2021).

CAPITULO III: ATAQUES DE DICCIONARIO

3. Ataque de diccionario

Un ataque de diccionario es una técnica de fuerza bruta en la cual los atacantes recorren palabras y frases comunes, como las que se encuentra en un diccionario, con el finde adivinar contraseñas. A diferencia de los ataques de fuerza bruta tradicionales que intentan todas las combinaciones posibles de manera sistemática para romper los controles de autentificación los ataques de diccionario utilizan un gran número de palabras y frases preseleccionadas, eso para reducir la probabilidad de adivinar correctamente una contraseña fácil, pero un ataque de diccionario requiere menos tiempo y recurso para ejecutarse (Dan, CSOONLINE, 2020).

3.1.Técnicas de ataque de diccionario

El proceso de un ataque de diccionario generalmente implica los siguientes pasos:

- Recopilación de información: El atacante puede recopilar información sobre el objetivo, como nombres de usuario, correos electrónicos, perfiles en redes sociales u otra información personal, para generar una lista de posibles contraseñas.
- Selección del diccionario: Se selecciona un diccionario de palabras o combinaciones de caracteres para realizar las pruebas. Estos diccionarios pueden ser generados por el propio atacante o estar disponibles en línea.

- Prueba de contraseñas: Utilizando un software automatizado, el atacante prueba las contraseñas del diccionario contra el sistema o la red objetivo. El software realiza una serie de intentos de inicio de sesión, probando cada contraseña hasta encontrar una que funcione o hasta agotar el diccionario.
- Éxito o fracaso: Si el ataque tiene éxito, el atacante obtiene acceso no autorizado al sistema o red objetivo. En caso de fracaso, el ataque puede intentar utilizar otros métodos o estrategias para comprometer la seguridad.

3.2.Medidas de protección contra ataques de diccionario

Un ataque de diccionario implica probar múltiples posibles contraseñas, a menudo basadas en palabras comunes o listas de contraseñas previamente filtradas, en un intento de adivinar una contraseña. Estos ataques son particularmente efectivos contra contraseñas débiles o comunes.

- Uso de contraseñas fuertes y únicas: Las contraseñas deben ser lo suficientemente largas y contener una combinación de letras mayúsculas y minúsculas, números y caracteres especiales. Evitar el uso de palabras completas, nombres, fechas de nacimiento, o cualquier otra información fácilmente adivinable.
- Limitación de intentos fallidos: Limitar el número de intentos de inicio de sesión fallidos. Tras cierto número de intentos, se puede bloquear temporalmente el acceso e imponer un retraso antes de intentar nuevamente.
- Autenticación de dos factores (2FA): Incluso si un atacante adivina la contraseña, la 2FA requiere un segundo método de autenticación, como un código temporal enviado a un teléfono móvil.
- Detección y respuesta: Implementar sistemas de detección que identifiquen intentos de inicio de sesión anómalos o patrones que se asemejen a un ataque de diccionario.

- Salado y hashing de contraseñas: Almacenar contraseñas de forma segura utilizando técnicas de salting y hashing para hacer que las contraseñas filtradas sean inútiles para un atacante.
- Educación y concientización del usuario: Capacitar a los usuarios sobre la importancia de usar contraseñas fuertes y cambiarlas regularmente.
- Uso de CAPTCHAs o desafíos similares: Requerir que los usuarios completen un desafío, como un CAPTCHA, después de varios intentos fallidos puede ayudar a prevenir ataques automatizados.

Implementando estas medidas y concientizando a los usuarios sobre las amenazas de seguridad, es posible reducir significativamente el riesgo de sufrir un ataque de diccionario. Es esencial mantenerse actualizado con las mejores prácticas de seguridad y revisar regularmente las políticas y sistemas de seguridad.

3.3.Tipos de ataque de diccionario

3.3.1. Ataque de diccionario básico

Los ataques de diccionario básicos son una forma de ataque de fuerza bruta que emplea listas de palabras comunes y frases, con las de un diccionario, para adivinar contraseñas. Estos ataques se benefician del uso frecuente de contraseñas simples y fáciles de recordad por parte de las personas, lo que significa que pueden ser exitosos mientras requieren menos recursos para ejecutarse (Swinhoe, 2020).

El uso de palabras comunes que sea fáciles de recordar permite los ataques de fuerza bruta realizando listados de palabras comunes o agrupando palabras muy utilizadas para tratar de lograr la mayor cantidad de combinaciones posibles con el fin de descifrar la contraseña utilizada por un

usuario, ya que los ataques de diccionario utilizan un numero grande pero limitado de palabras o frases preseleccionadas para reducir el consumo de tiempo y de recursos.

Muchas herramientas utilizadas para ataques de diccionario incluyen contraseñas comunes filtradas en brechas de seguridad en línea y variantes comunes de ciertas palabras y frases. El hecho de que las palabras a menudo reutilicen contraseñas y no las cambien después de las brechas significa que este tipo de ataque puede ser fácil de ejecutar y es probable que tenga éxito con suficiente tiempo e intentos (Swinhoe, 2020).

3.3.2. Ataques basados en reglas

En los ataques de diccionario basados en reglas el atacante conoce información sobre la contraseña que desea descifrar, parámetros como el número de caracteres, si usa números o símbolos, e incluso una de las palabras de la cual está conformada la contraseña a descifrar y con el conocimiento de estos parámetros el atacante puede mejorar las posibles combinaciones logrando que sean más especificadas y reduciendo el número de combinaciones a realizar por la herramienta que se use (Livejournal, 2019).

Cuando el atacante posee un mayor conocimiento sobre la estructura de la contraseña, facilita su tarea al momento de depurar los datos que usara, como el posicionamiento especifico de un carácter o un posible orden a seguir, logrando que sean más precisos al momento de realizar las combinaciones exigiendo un menor procesamiento de computadora y tiempo de espera para el atacante.

3.3.3. Ataques híbridos

En este tipo de ataques se mezclan los ataques de diccionario básicos y ataques de diccionarios basados en reglas, para que las contraseñas habituales se combinan con caracteres aleatorios en diferentes posiciones de las palabras, para crear una base de datos más amplia abarcando más posibles combinaciones de posibles contraseñas, que posteriormente se podrán poner a prueba (Hackwise, 2021).

Con el fin de lograr obtener la contraseña de un usuario un atacante puede combinar los ataques de diccionario básicos y basados en reglas para lograr una mayor cantidad de posibles combinaciones, ya que a las palabras ya definidas por quien realiza el diccionario de datos se le agrega números o caracteres comunes de forma aleatoria en diferentes lugares, para abarcar muchas más posibilidades de contraseñas.

3.3.4. Ataques de diccionario dirigido

Los ataques de diccionario dirigidos son una forma de ciberataques en las que se utiliza un listado de frases o palabras específicamente diseñadas para una persona u organización en particular, estos listados se basan en información que es relevante para el objetivo como nombres suyos o de familiares, fechas importantes, términos asociados o gustos del usuario, y estos datos son obtenidos de investigaciones, obtenciones de información disponible públicamente o filtraciones (Leighton, 2019).

3.4. Comparación ataques de diccionario

Una vez revisados los diferentes tipos de ataques de diccionario existentes con sus características principales y en qué consisten cada uno, se plantea una comparativa de los mismo para enfatizar en el objetivo que tiene cada uno de los ataques.

Tabla 1

Comparativa de ataques de diccionario

Tipo de Ataque de Diccionario	**Descripción**	**Eficacia**	**Vulnerabilidades Objetivo**
Básico	Utiliza palabras o frases comunes. Basado en diccionarios de palabras comunes o listas de contraseñas filtradas.	Más eficiente que la fuerza bruta.	Efectivo contra contraseñas débiles o basadas en palabras comunes.
Dirigido	Crea un diccionario basado en información personal de la víctima (nombres, fechas de nacimiento, etc.).	Muy efectivo cuando se usa información personal en contraseñas.	Dirigido a víctimas que incorporan datos personales en sus contraseñas.
Híbrido	Combina palabras comunes con números, caracteres	Mejora las posibilidades de éxito comparado con el ataque simple.	Efectivo contra contraseñas que incluyen caracteres especiales o números,

	especiales y otras variaciones.		pero aún son predecibles.
Basado en reglas	Usa algoritmos y técnicas sofisticadas para generar combinaciones. Incluye reglas gramaticales y sustituciones inteligentes.	Más rápido y preciso que los métodos anteriores.	Puede superar contraseñas más complejas con variados patrones y algoritmos.

Al analizar la tabla anterior, se enfatizó en la eficiencia y en las vulnerabilidades objetivo que tiene cada tipo de ataque, si bien los ataques de diccionario basado en reglas pueden llegar a obtener contraseñas más complejas los ataques de diccionario dirigidos incorporan los datos de la víctima lo que lo hace más precioso al determinar una posible contraseña por lo cual es el que se plantea para el desarrollo de este trabajo.

Se puede determinar que para el trabajo realizado se trabajó con un ataque de diccionario dirigido ya que no se usan bases de datos de contraseñas ya existentes, se crea un repositorio propio con la información del usuario al que se está realizado el ataque.

CAPITULO IV: METODOLOGIA PARA ATAQUE DE DICCIONARIO

4. Planteamiento para ataque de diccionario

4.1.Herramientas para crear diccionarios de datos

En el ámbito de la gestión de datos, la creación de diccionarios de datos es un paso crucial para garantizar la coherencia y la calidad de la información. Las herramientas modernas para desarrollar estos diccionarios ofrecen una variedad de funciones que facilitan la documentación de metadatos, la definición de relaciones entre datos y la estandarización de términos. Estas herramientas no solo proporcionan una plataforma para mantener un registro detallado de los elementos de datos, sino que también mejoran la comunicación entre los miembros del equipo al proporcionar una terminología común. Al integrar tecnologías como la inteligencia artificial y el aprendizaje automático, estas herramientas avanzan hacia un enfoque más dinámico y adaptativo en la gestión de datos, permitiendo una actualización y un mantenimiento más eficientes de los diccionarios de datos. Esta evolución es fundamental para afrontar los retos de la creciente complejidad y volumen de datos en diversas industrias.

4.1.1. CUPP

La aplicación Common User Passwords Profiler (Cupp) es una herramienta diseñada para la creación de listas de contraseñas personalizadas en base a diferentes parámetros. Esta herramienta es ampliamente utilizada en el ámbito de la seguridad informática, más específicamente en las actividades de pentesting (pruebas de penetración) y auditorías de seguridad. El propósito de Cupp es generar posibles contraseñas que un usuario específico podría utilizar, aprovechando la

información personal que a menudo se incluye en las contraseñas, como fechas de nacimiento, nombres de mascotas y fechas importantes (KALI.TOOLS, 2021).

Cupp puede ser empleado por profesionales de la seguridad informática para evaluar la robustez de las políticas de contraseñas dentro de una organización. Mediante el uso de perfiles personalizados y la integración de información específica de los usuarios objetivo, se pueden crear listas de contraseñas que reflejen los hábitos comunes de creación de contraseñas. Esto permite a los auditores realizar ataques de fuerza bruta o de diccionario de manera más eficiente al reducir el conjunto de contraseñas a aquellas que tienen una mayor probabilidad de ser utilizadas.

El uso ético de Cupp se limita al ámbito de las pruebas autorizadas y los ejercicios de seguridad cibernética, donde los profesionales tienen permiso para realizar tales acciones como parte de una evaluación de seguridad. Su uso contra sistemas o usuarios sin consentimiento explícito es ilegal y contrario a las prácticas éticas en ciberseguridad.

4.1.2. Crunch

La herramienta Crunch es una aplicación diseñada para sistemas Unix y Linux, que sirve para la generación de listas de palabras “wordlists” que se utilizan en pruebas de penetración y auditorías de seguridad. Estas listas de palabras son útiles para realizar ataques de fuerza bruta o de diccionario contra sistemas de autenticación para probar su resistencia ante intentos de acceso no autorizado.

La función principal de Crunch es permitir a los profesionales de la seguridad crear listas de palabras a medida, basadas en criterios específicos proporcionados por el usuario. Por ejemplo, Crunch puede generar todas las combinaciones posibles de caracteres alfanuméricos que cumplan con un conjunto de parámetros definidos, como longitud mínima y máxima de la contraseña,

juegos de caracteres a incluir (por ejemplo, solo letras minúsculas y solo números incluso patrones con los cuales las contraseñas deben alinearse (Ehacking, 2019).

Además, Crunch puede ser usado para crear listas de palabras que incluyan o excluyan combinaciones específicas de caracteres basándose en reglas o patrones, lo que permite a los auditores y testers de seguridad simular ataques más dirigidos y efectivos. Esto es particularmente útil cuando se ha recopilado información previa sobre las posibles contraseñas a través de técnicas de ingeniería social o durante la fase de reconocimiento de una prueba de penetración.

4.1.3. Herramientas “Cupp” y “Crunch”

Después de revisar las herramientas Cupp y Crunch, para poder diferenciar las características de cada aplicación a continuación se plantea una comparativa en formato de tabla, aunque se utilizaran ambas herramientas para crear un diccionario de datos ya que cada una procesa la información ingresada de manera diferente.

Tabla 2

Tabla comparativa entre las herramientas Cupp y Crunch.

Característica	Cupp	Crunch
Tipo de Herramienta	Generador de perfiles de contraseñas	Generador de wordlists
Lenguaje de Programación	Python	C

Uso Principal	Crear diccionarios basados en datos personales	Crear diccionarios basados en patrones
Personalización	Media; basada en perfiles de usuarios	Alta; basada en parámetros completos
Combinatoria	Limitada a datos personales	Amplia; genera todas las combinaciones posibles
Intuitividad	Alta; interfaz sencilla y preguntas guiadas	Media; requiere conocimiento de opciones de línea de comandos
Especificidad	Alta en contexto personal	Alta en contexto de patrones de caracteres
Versatilidad	Menos versátil; más enfocada en el perfil de un usuario	Más versátil; crea wordlists para cualquier escenario de fuerza bruta o diccionario
Velocidad	Depende de la complejidad del perfil	Muy rápida; optimizada en C
Integración con Herramientas de Pentesting	Puede integrarse con otras herramientas de pentesting	Ampliamente integrable con herramientas como John the Ripper, Hydra, etc.
Facilidad de Instalación	Fácil; suele ser un script de Python simple	Fácil a moderada; puede requerir compilación en algunos sistemas

Contribución a la Seguridad	Ayuda a entender la seguridad desde la perspectiva del usuario	Ayuda a entender la seguridad desde la perspectiva de la complejidad y patrones de contraseñas
Capacidad de Predicción	Basada en el perfil psicolingüístico de usuarios	Basada en matemáticas y probabilidad de combinaciones

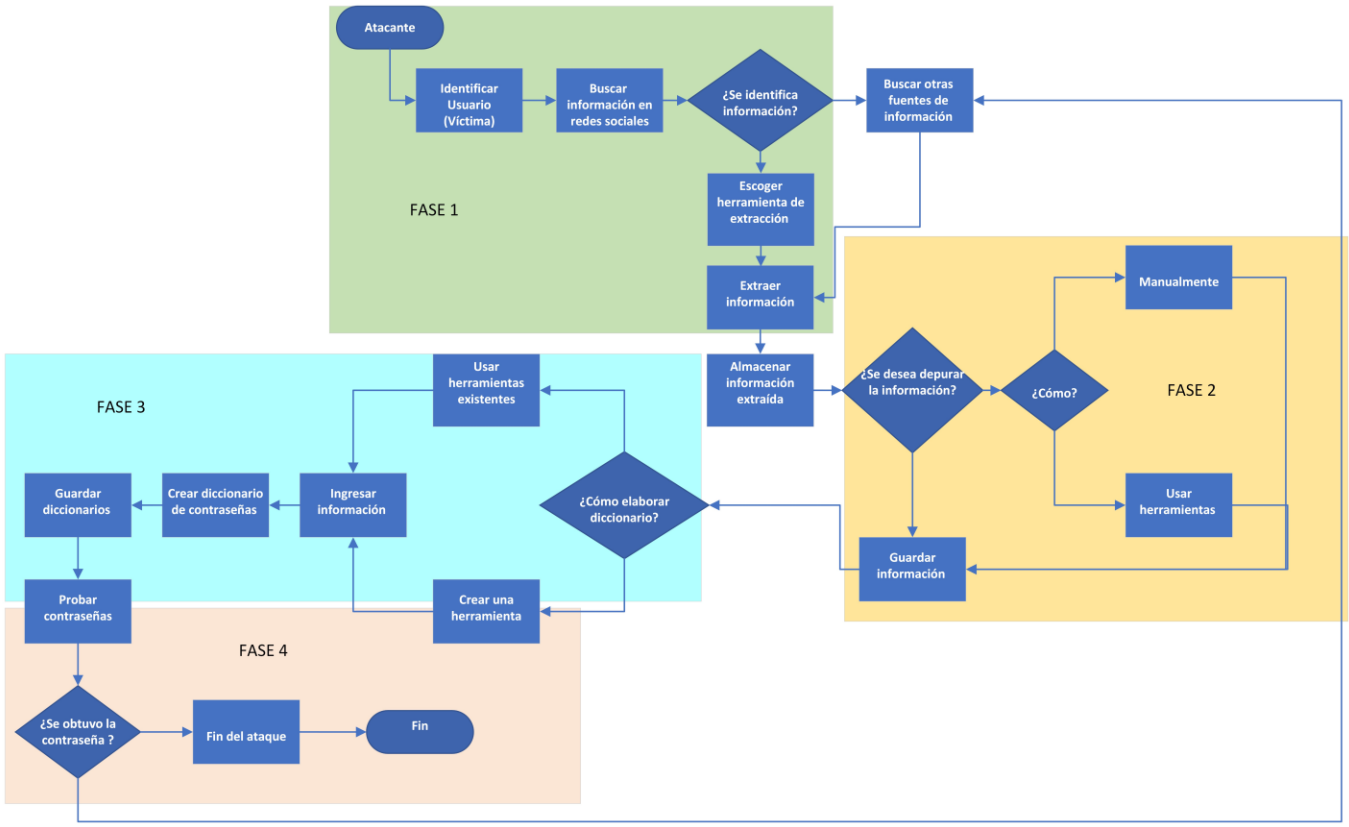
Esta tabla compara aspectos clave que se deben tener en cuenta al elegir una herramienta sobre otra para pruebas de seguridad específicas. Cupp es más útil cuando se dispone de información detallada sobre un usuario objetivo y se desea crear un diccionario de contraseñas personalizado que podría utilizar esa persona. Crunch, por otro lado, es más adecuado para generar grandes listas de contraseñas posibles basadas en patrones conocidos o posibles estructuras de contraseñas sin necesitar conocimiento previo sobre un usuario específico.

4.2.Metodología para ataque de diccionario

Para poder realizar un ataque de diccionario no existe una metodología o un paso especifico, por lo cual para poder llevar a cabo el ataque se planteó un proceso que se seguirá, se proponen acciones y decisiones a tomar con el fin de realizar un ataque de diccionario, con información pública que se encuentre en las redes sociales, dividido en 4 fases (Ver figura 2).

Figura 2

Metodología para realizar ataque de diccionario.



El diagrama de flujo que describe una propuesta de un proceso para realizar un ataque de diccionario con el fin de obtener la contraseña de un usuario "víctima" por medio de la recopilación de información extraída principalmente de sus redes sociales. A continuación, un análisis paso a paso del diagrama propuesto.

- **Fase 1: Recopilación de Información**

    1. Buscar información en redes sociales.
    2. Si se identifica información, se escoge una herramienta de extracción para obtener datos del usuario "víctima".
    3. Extraer la información y almacenar la información extraída.
    4. Si no se identifica información, buscar otras fuentes de información.

- **Fase 2: Depuración de Información**
    1. Decidir si se desea depurar la información extraída.
    2. Si se decide depurar, se puede hacer manualmente o usando herramientas específicas.
    3. Guardar la información depurada.
- **Fase 3: Creación de un Diccionario de Contraseñas**
    1. Usar herramientas existentes o crear una nueva herramienta para ingresar información.
    2. Crear un diccionario de contraseñas con la información obtenida.
    3. Guardar los diccionarios de contraseñas creados.
- **Fase 4: Prueba de Contraseñas**
    1. Probar las contraseñas del diccionario creado.
    2. Si se obtiene la contraseña, el ataque finaliza con éxito.
    3. Si no se obtiene la contraseña, se da por terminado el ataque.

Este diagrama muestra un esquema general de cómo un atacante puede recopilar y utilizar información obtenida de redes sociales y posiblemente otras fuentes para comprometer la seguridad de alguien. Es un ejemplo claro de las tácticas que pueden emplear los atacantes para un ataque de diccionario y otros tipos de ataques cibernéticos. Es importante destacar que realizar tales acciones sin consentimiento es ilegal y éticamente incorrecto. Este análisis se proporciona únicamente con fines educativos y no debe utilizarse para realizar actividades ilegales.

4.3.Extracción de información de redes sociales

Para poder extraer información de las redes sociales, Facebook, Instagram y X (Twitter) se planteó realizar con la técnica de Web Scraping que consiste en extraer información de páginas

web, realizando un script que simule la navegación en una persona en una página web recopilando datos.

Al momento de realizar el script en Python para realizar Web Scraping se evidencio que, las redes sociales escogidas para extraer la información cuentan en sus políticas de protección con la implementación de robots.txt que es un archivo para gestionar el tráfico y a que URL's se puede acceder en un sitio web, por lo cual manteniendo el objetivo central de realizar un ataque de diccionario con información de redes sociales se planteó un proceso diferente para la extracción de información.

Se creó un script en Python, para que el atacante realice capturas de pantalla del perfil de usuario donde se muestre la información que es de libre acceso en la web de la persona a la que decide atacar, para agrupar las imágenes obtenidas por el atacante y agruparlas en un repositorio el cual el código realizado, leerá cada imagen que deberá estar en formato .png, transcribirá el texto encontrado en todas las imágenes que se encuentren dentro del repositorio.

Figura 3

Código realizado en Python para transcripción de imágenes.

```python
from PIL import Image
import pytesseract
import os

# Ruta al ejecutable de Tesseract
pytesseract.pytesseract.tesseract_cmd = r"C:\Users\ariel\OneDrive\Escritorio\Tesseract-OCR\tesseract.exe"

# Ruta de la carpeta que contiene las imágenes
ruta_carpeta = r"C:\Users\ariel\PycharmProjects\pythonProject\imagenes\capturas"

# Ruta de la carpeta para guardar los archivos de texto
ruta_guardado = r"C:\Users\ariel\PycharmProjects\pythonProject\imagenes\Usuarios"

# Recorrer los archivos en la carpeta
for archivo in os.listdir(ruta_carpeta):
    # Construir la ruta completa del archivo
    ruta_completa = os.path.join(ruta_carpeta, archivo)

    # Comprobar si el archivo es una imagen PNG
    if archivo.lower().endswith('.png'):§
        # Abrir la imagen
        imagen_abierta = Image.open(ruta_completa)

        # Convertir la imagen a texto
        texto = pytesseract.image_to_string(imagen_abierta)
```

Figura 4

Código realizad o en Python para transcripción de imágenes parte 2.

```python

        # Construir el nombre base del archivo de texto
        nombre_base = "usuario"
        contador = 1
        nombre_archivo = f"{nombre_base}{contador}.txt"

        # Verificar si el archivo ya existe y actualizar el nombre consecutivamente
        while os.path.exists(os.path.join(ruta_guardado, nombre_archivo)):
            contador += 1
            nombre_archivo = f"{nombre_base}{contador}.txt"

        # Guardar el texto en un archivo TXT
        with open(os.path.join(ruta_guardado, nombre_archivo), 'w') as file:
            file.write(texto)

        print(f"Texto extraído de {archivo} guardado en {nombre_archivo}")
```

Para un mejor entendimiento del script realizado se explica a continuación cada línea escrita y que realiza cada una de ellas enumerándolas, pero sin tomar en cuenta las líneas que ya están comentadas dentro del script.

1. `from PIL import Image`

- Importa la clase `Image` del módulo `PIL`, que se utiliza para abrir y manipular imágenes.

2. `import pytesseract`

- Importa el módulo `pytesseract`, que permite la utilización de Tesseract-OCR en Python para convertir imágenes en texto.

3. `import os`

- Importa el módulo **os**, que proporciona funciones para interactuar con el sistema operativo, como listar archivos en un directorio.

4. `pytesseract.pytesseract.tesseract_cmd = r"C:\Users\ariel \OneDrive\Escritorio\Tesseract-OCR\tesseract.exe"`

- Asigna la ruta del ejecutable de Tesseract a una variable dentro de `pytesseract`, que es necesaria para que la librería sepa dónde ejecutar el OCR.

5. `ruta_carpeta =  r"C:\Users\ariel\PycharmProjects\ pythonProject \imagenes\capturas"`

- Establece una variable con la ruta de la carpeta que contiene las imágenes a procesar.

6. `ruta_guardado = r"C:\Users\ariel\PycharmProjects \pythonProject\imagenes\Usuarios"`

- Define una variable con la ruta de la carpeta donde se guardarán los archivos de texto resultantes del OCR.

7. `for archivo in os.listdir(ruta_carpeta):`

- Inicia un bucle que itera sobre cada archivo en el directorio de imágenes.

8. `ruta_completa = os.path.join(ruta_carpeta, archivo)`

- Construye la ruta completa hacia un archivo específico combinando la ruta de la carpeta con el nombre del archivo.

```
9.  if archivo.lower().endswith('.png'):
```

- Comprueba si el nombre del archivo termina en '.png' (en minúsculas), indicando que es una imagen PNG.

```
10. imagen_abierta = Image.open(ruta_completa)
```

- Abre la imagen usando la clase **Image** importada de **PIL**.

```
11. texto = pytesseract.image_to_string(imagen_abierta)
```

- Utiliza **pytesseract** para convertir la imagen abierta en texto.

```
12. nombre_base = "usuario"
```

- Establece una cadena base para los nombres de archivos de texto que se crearán.

```
13. contador = 1
```

- Inicializa un contador en 1.

```
14. nombre_archivo = f"{nombre_base}{contador}.txt"
```

- Crea un nombre de archivo de texto usando el nombre base y el contador.

```
15. while os.path.exists(os.path.join(ruta_guardado,
nombre_archivo)):
```

- Entra en un bucle que verifica si un archivo con el nombre generado ya existe en la ruta de guardado.

```
16. contador += 1
```

- Incrementa el contador para evitar sobrescribir archivos existentes.

```
17. nombre_archivo = f"{nombre_base}{contador}.txt"
```

- Actualiza el nombre del archivo con el nuevo valor del contador.

```
18. with open(os.path.join(ruta_guardado, nombre_archivo), 'w')
as file:
```

- Abre (o crea) un archivo de texto para escribir en la ruta de guardado con el nombre de archivo actualizado.

19. `file.write(texto)`

- Escribe el texto extraído en el archivo de texto.

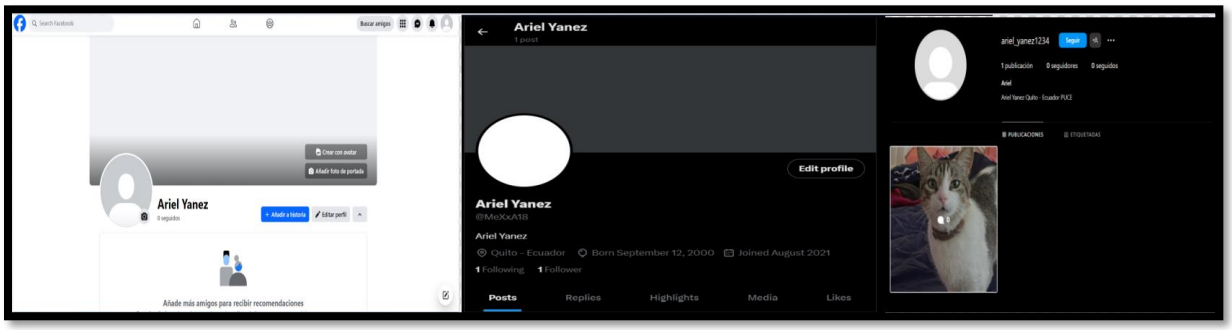
20. `print(f"Texto extraído de {archivo} guardado en {nombre_archivo}")`

- Imprime un mensaje en la consola que indica que el texto de la imagen ha sido extraído y guardado en un archivo de texto.

Una vez obtenido el código que realizara la extracción de la información se identificara a la persona a la cual se realizara el ataque, para después buscar sus redes sociales, en este caso se encontró los perfiles del usuario en Facebook, Instagram y X(Twitter).

Figura 5

Redes sociales del usuario escogido.

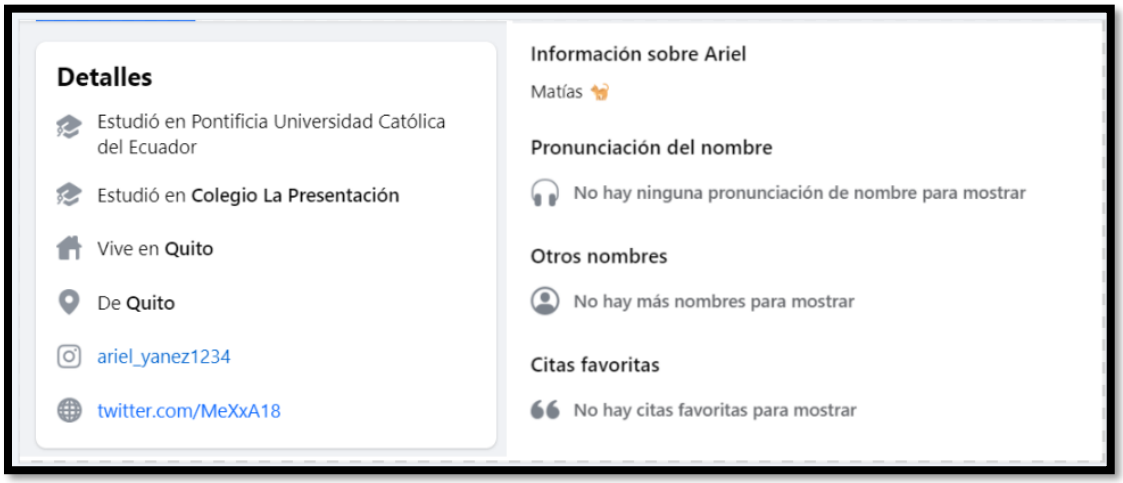


Después de encontrar las redes sociales del usuario escogido se tomó capturas de pantalla de la información compartida por el mismo, en sus perfiles la cual es de acceso público para cualquier cibernauta, el número de capturas y de que sección realizarlas es de criterio propio de quien esté realizando el ataque.

De la red social Facebook, se encontró la siguiente información, la cual a criterio propio se cree es relevante para la creación del diccionario, en el perfil del usuario dentro del apartado de inflación.

Figura 6

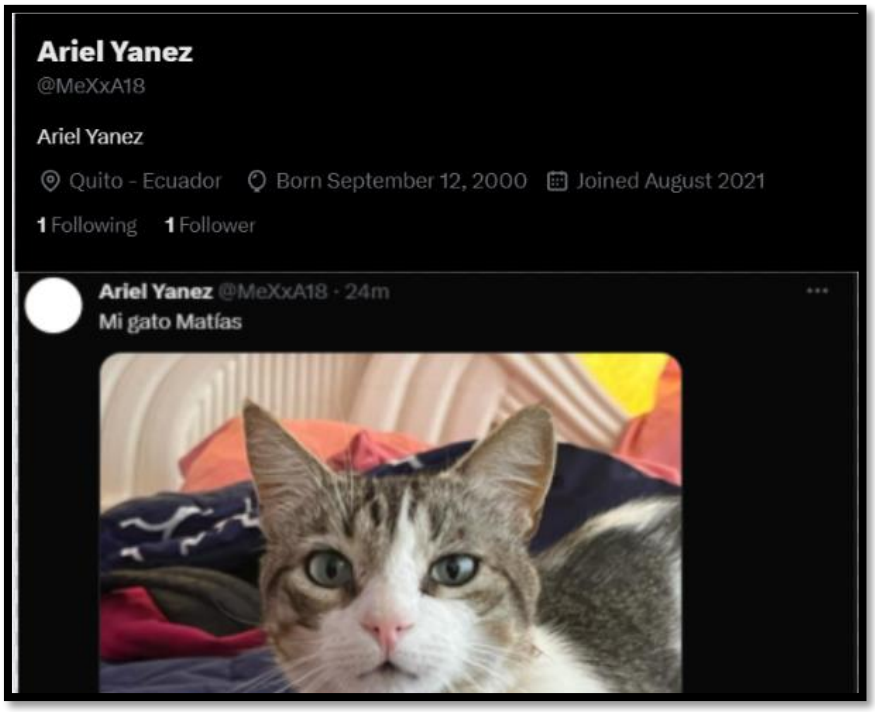
Información del usuario en la red social Facebook.



De la red social X(Twitter), se encontró la siguiente información, y se realizó dos capturas de pantalla, una del mural principal del usuario y la segunda de una publicación realizada que contiene un pie de página agregado por el dueño.

Figura 7

Información del usuario en la red social X (Twitter).



De la red social Instagram se encontró la siguiente información, y se realizaron dos capturas de pantalla de la descripción del perfil y del comentario realizado en una publicación del usuario.

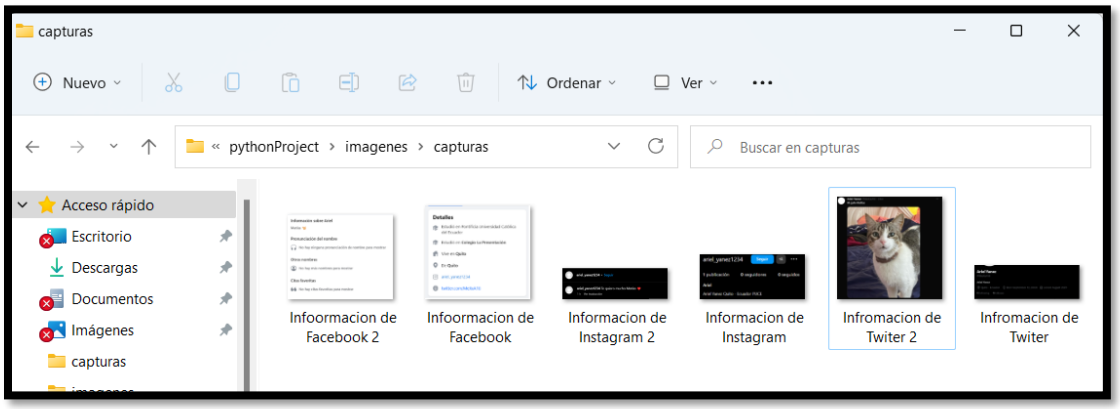
Figura 8

Información del usuario en la red social Instagram.

Una vez tomadas las capturas de pantalla se procedió a guardar las imágenes, en formato `.png` y asignándole un nombre a cada una, en un repositorio llamado “capturas” el cual será utilizado por el script realizado en Python para la transcripción a texto de las palabras encontradas.

Figura 9

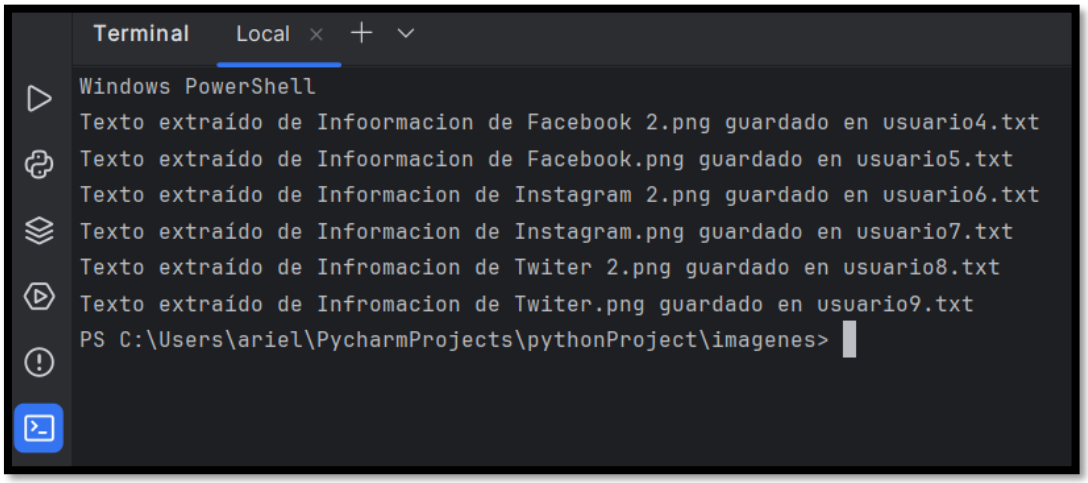
Capturas de pantalla tomadas de la rede sociales del usuario.



Ejecutando el script creado se crea un archivo en formato `.txt` por cada captura de pantalla que se encuentre en la carpeta en la que fueron guardadas, la Figura 9 muestra el texto extraído con éxito y su almacenamiento.

Figura 10

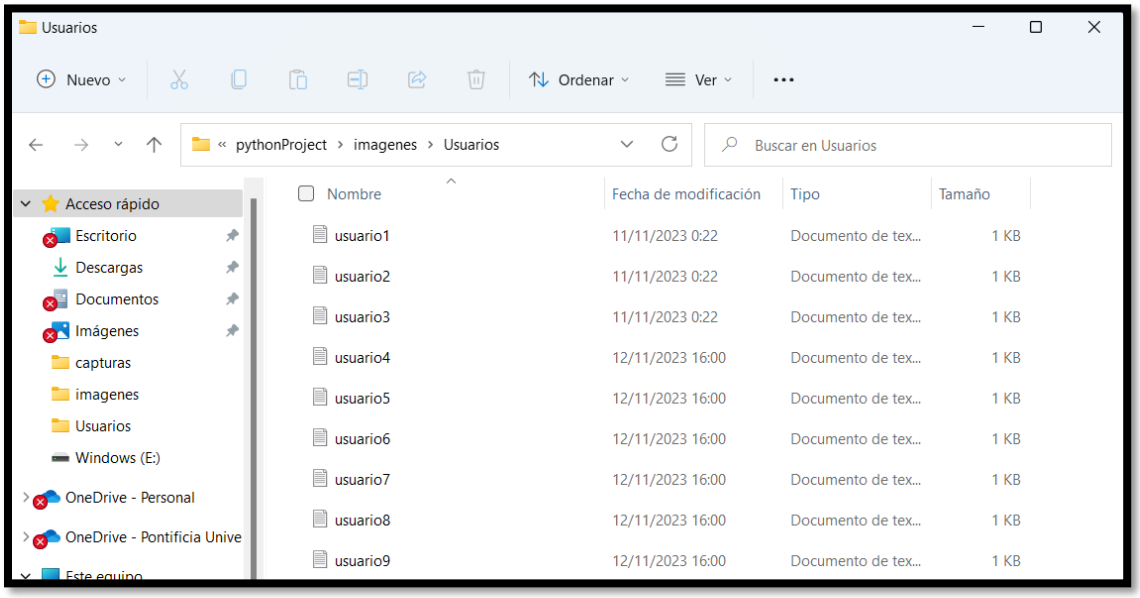
Salida del script ejecución correcta.



Creación de `.txt` con nombres En la Figura 10 se muestra el repositorio de nombre “Usuarios” en el cual se almacenan los archivos creados, que contienen la transición de imágenes a texto de las imágenes guardadas, cada archivo se crea con el nombre de “usuario” y se agrega un numero consecutivo al final para evitar la duplicidad de archivos creados.

Figura 11

Creación de documentos por cada foto tomada.



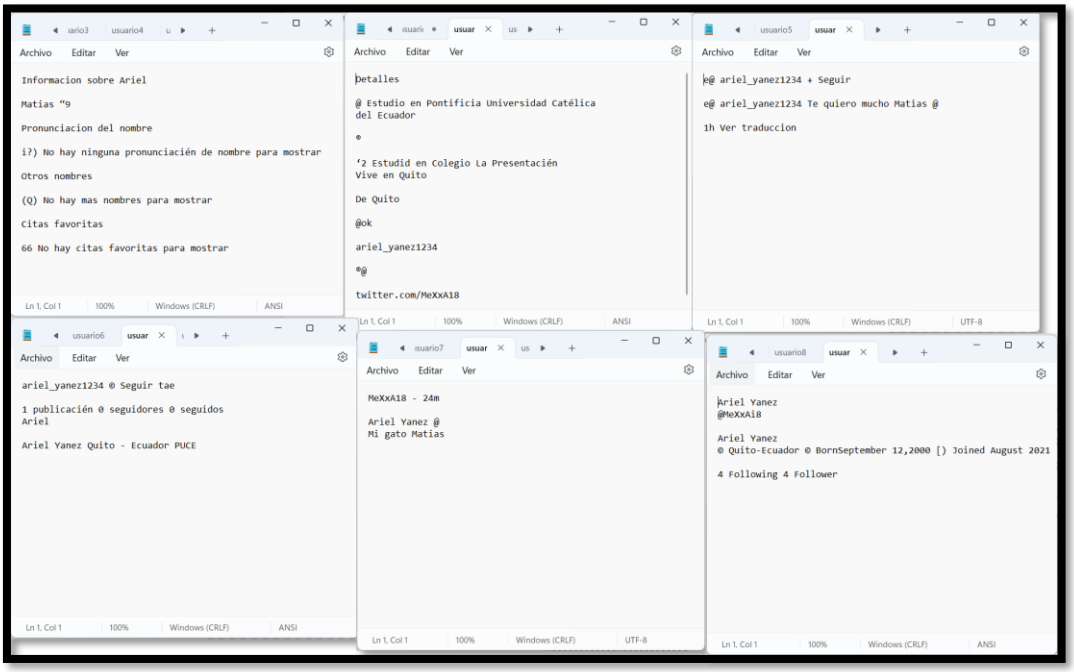
Hasta este punto culmina la fase de extracción de información de un usuario seleccionado, identificando las redes sociales de la persona escogida, realizando capturas de pantalla y utilizando el script creado en Python.

4.4.Depuración de la información extraída

En la Figura 11 se muestran los archivos `.txt` creados con todo el texto reconocido, transcrito y extraído de las capturas de pantalla realizadas anteriormente para su posterior depuración.

Figura 12

Textos extraídos de las imágenes.



Para depurar la información extraída se utilizó el siguiente script creado en Python el cual lee todos los caracteres de los archivos creados anteriormente e identifica cuales son las palabras más comunes y las menos comunes para después enlistarlas, logrando un mejor ordenamiento y entendimiento de la información que se ha obtenido.

Figura 13

Script en Python para depurar información.

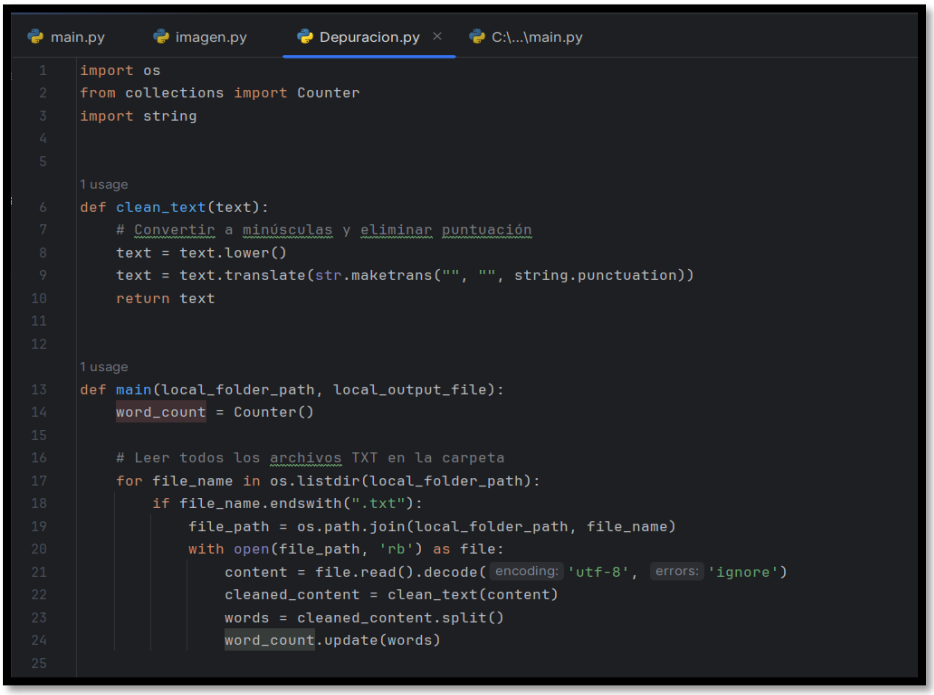
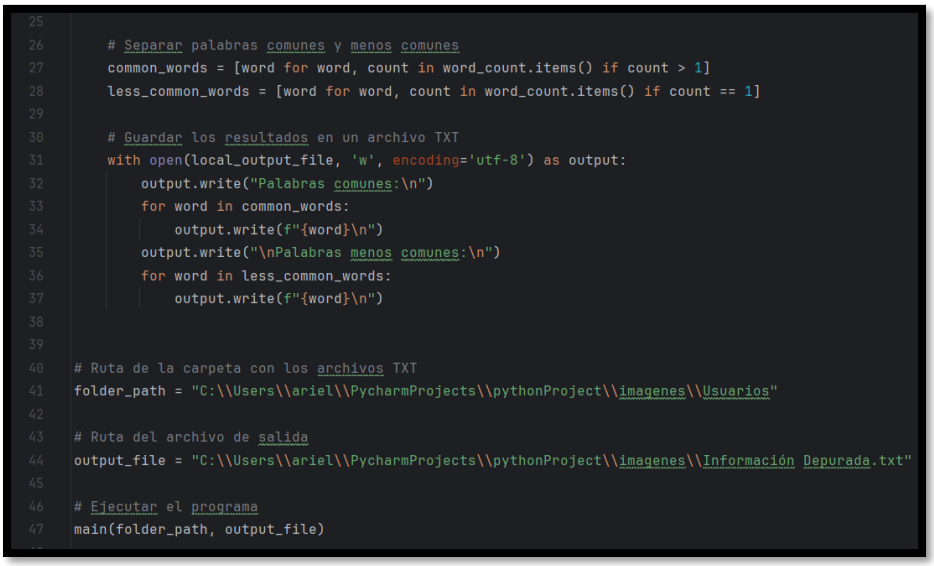


Figura 14

Script en Python para depurar información parte 2.



```
    # Separar palabras comunes y menos comunes
    common_words = [word for word, count in word_count.items() if count > 1]
    less_common_words = [word for word, count in word_count.items() if count == 1]

    # Guardar los resultados en un archivo TXT
    with open(local_output_file, 'w', encoding='utf-8') as output:
        output.write("Palabras comunes:\n")
        for word in common_words:
            output.write(f"{word}\n")
        output.write("\nPalabras menos comunes:\n")
        for word in less_common_words:
            output.write(f"{word}\n")


# Ruta de la carpeta con los archivos TXT
folder_path = "C:\\Users\\ariel\\PycharmProjects\\pythonProject\\imagenes\\Usuarios"

# Ruta del archivo de salida
output_file = "C:\\Users\\ariel\\PycharmProjects\\pythonProject\\imagenes\\Información Depurada.txt"

# Ejecutar el programa
main(folder_path, output_file)
```

Para un mejor entendimiento del script realizado se explica a continuación cada línea escrita y que realiza cada una de ellas enumerándolas, pero sin tomar en cuenta las líneas que ya están comentadas dentro del script.

1. `import os`
    - Importa el módulo **`os`**, que proporciona funciones para interactuar con el sistema operativo.
2. `from collections import Counter`
    - Importa la clase **`Counter`** del módulo **`collections`**, utilizada para contar objetos hashables.
3. `import string`
    - Importa el módulo **`string`**, que contiene operaciones comunes de strings y constantes como la cadena de puntuación.
4. `def clean_text(text):`

- Define una función llamada `clean_text` que acepta un argumento `text`.

5. `text = text.lower()`
   - Convierte todo el texto a minúsculas.
6. `text = text.translate(str.maketrans("", "", string.punctuation))`
   - Elimina toda la puntuación del texto.
7. `return text`
   - Retorna el texto limpio.
8. `def main(local_folder_path, local_output_file):`
   - Define la función principal `main` con dos argumentos: la ruta de la carpeta y el archivo de salida.
9. `word_count = Counter()`
   - Crea una instancia de `Counter` para contar las palabras.
10. `for file_name in os.listdir(local_folder_path):`
    - Inicia un bucle sobre cada archivo en la ruta de la carpeta especificada.
11. `if file_name.endswith(".txt"):`
    - Comprueba si el archivo termina en `.txt`.
12. `file_path = os.path.join(local_folder_path, file_name)`
    - Crea la ruta completa del archivo.
13. `with open(file_path, 'rb') as file:`
    - Abre el archivo en modo de lectura binaria.
14. `content = file.read().decode('utf-8', 'ignore')`
    - Lee el contenido del archivo, lo decodifica a UTF-8 e ignora errores.
15. `cleaned_content = clean_text(content)`
    - Limpia el contenido del archivo utilizando la función `clean_text`.

16. `words = cleaned_content.split()`
    - Separa el contenido limpio en palabras.
17. `word_count.update(words)`
    - Actualiza el contador de palabras con las palabras del archivo.
18. `common_words = [word for word, count in word_count.items() if count > 1]`
    - Crea una lista de palabras comunes (aquellas con más de una ocurrencia).
19. `less_common_words = [word for word, count in word_count.items() if count == 1]`
    - Crea una lista de palabras menos comunes (aquellas con una sola ocurrencia).
20. `with open(local_output_file, 'w', encoding='utf-8') as output:`
    - Abre el archivo de salida en modo escritura con codificación UTF-8.
21. `output.write("Palabras comunes:\n")`
    - Escribe en el archivo de salida un título para las palabras comunes.
22. `for word in common_words:`
    - Itera sobre cada palabra común.
23. `output.write(f"{word}\n")`
    - Escribe cada palabra común en el archivo de salida.
24. `output.write("\nPalabras menos comunes:\n")`
    - Escribe en el archivo de salida un título para las palabras menos comunes.
25. `for word in less_common_words:`
    - Itera sobre cada palabra menos común.
26. `output.write(f"{word}\n")`
    - Escribe cada palabra menos común en el archivo de salida.

27. `folder_path = "C:\\...\\Usuarios"`

- Establece la variable `folder_path` con la ruta de la carpeta que contiene los archivos de texto.

28. `output_file = "C:\\...\\Información Depurada.txt"`

- Define la variable `output_file` con la ruta del archivo de salida.

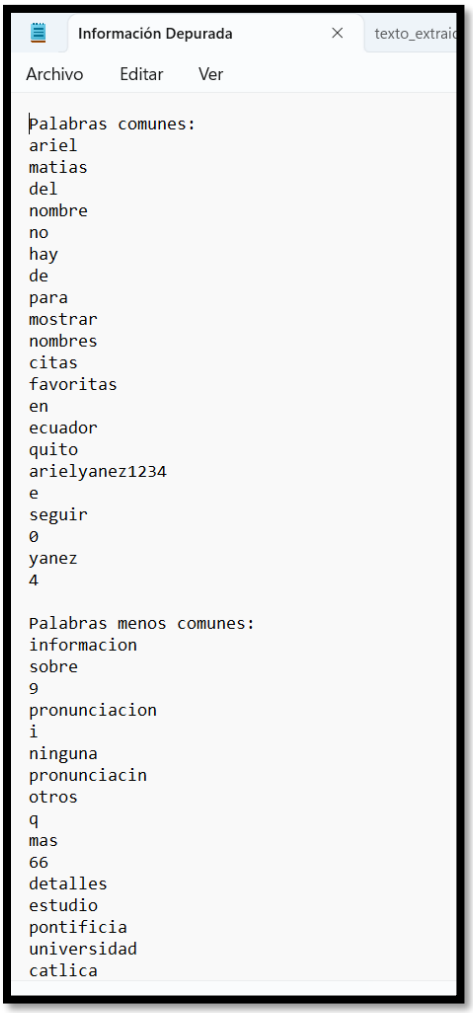
29. `main(folder_path, output_file)`

- Llama a la función `main` con las rutas de la carpeta y el archivo de salida como argumentos.

En la Figura 15 se muestra el archivo creado en el cual se agrupa toda la información extraída en los anteriores documentos, identificado cuales son las palabras más comunes y las menos comunes para enlistarlas y poder tener un mejor entendimiento de la información que se ha recopilado de las redes sociales del usuario.

Figura 15

Archivo .txt con información recopilada y depurada.



Aquí culmina la etapa para depurar la información obtenida, utilizando los archivos con las transcripciones de texto de las capturas de pantalla, un script en Python para ordenar las palabras y un archivo de texto para almacenar la nueva información depurada.

4.5.Elaboración de diccionario

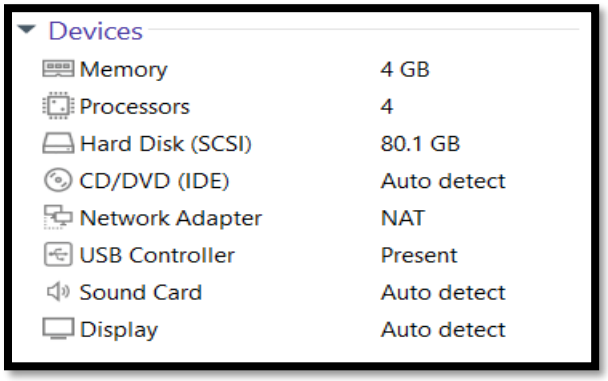
Para la creación del diccionario de datos se utilizarán dos herramientas, en el sistema operativo Kali Linux, las cuales son Cupp y Crunch las cuales permiten crear diccionarios de datos con la misma información, pero distinta técnica.

Para utilizar Kali Linux se virtualizo una computadora con la ayuda del hipervisor VMware, en la cual se instaló el sistema operativo y se creó con las siguientes características, considerando el hardware disponible de la computadora anfitriona.

En la Figura 16 se muestra las características con las cuales se creó la computadora virtualizada para poder instalar un sistema operativo diferente al de la computadora anfitriona, con el fin de utilizar Kali Linux y las herramientas que este sistema ofrece.

Figura 16

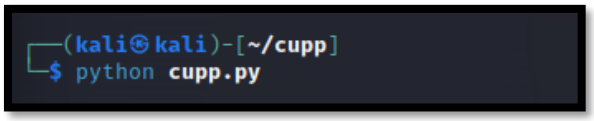
Características de computadora virtualizada.



Una vez ya instalado la aplicación se verifica la ubicación donde fue creada el repositorio y se lista el contenido de este usando el comando `$ls` comprobando que todos los elementos que se van a utilizar se encuentren dentro del repositorio creado, por último, para ingresar a la aplicación se usa el comando `$ Python cupp.py` como se muestra en la Figura 17.

Figura 17

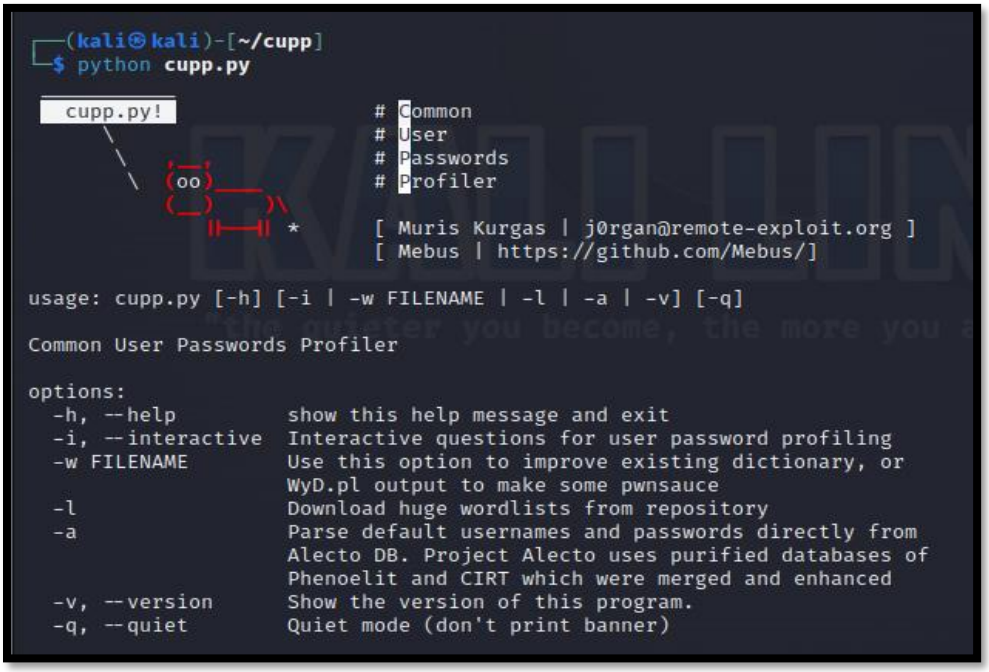
Ingresar a la herramienta Cupp.py.



La Figura 18 muestra la aplicación Cupp con las opciones y acciones que permite realizar la aplicación, también muestra las opciones que se pueden realizar para crear el diccionario de datos listadas con el nombre del comando que se debe utilizar para hacer el llamado a esa opción, el nombre de la opción y la descripción de lo que realiza.

Figura 18

Índice de la aplicación Cupp

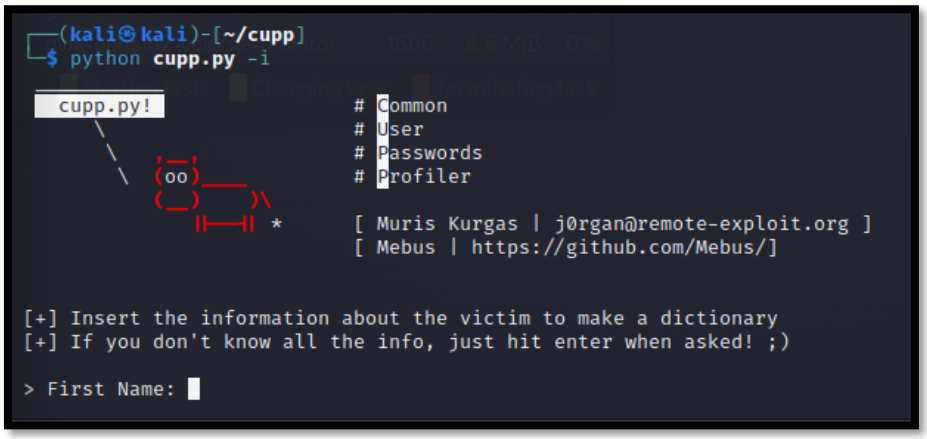


En la Figura 19 se muestra la selección de la opción i usando el comando `$ Python cupp.py -i`, ya que esta opción permite realizar un diccionario de datos de una manera

interactiva de forma en que la aplicación va a realizar preguntas ya establecidas con el fin de recopilar información personal de la persona sobre la que se está realizando el diccionario, el programa permite ingresar nombres usando mayúsculas y minúsculas, números y caracteres especiales.

Figura 19

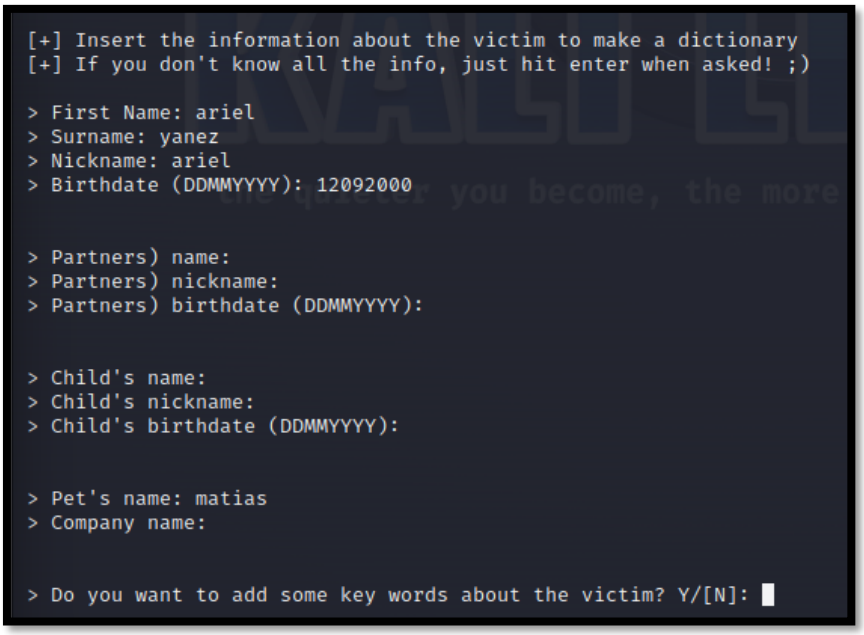
Selección de las opciones



La Figura 20 muestra las preguntas realizadas por el programa, estas preguntas son ya predeterminadas con el fin de recopilar información personal de la persona como son sus datos y los de sus familiares, mascotas, trabajo estos son datos comunes que las personas utilizan para crear sus contraseñas ya que son datos fáciles de recordad y que no van a variar.

Figura 20

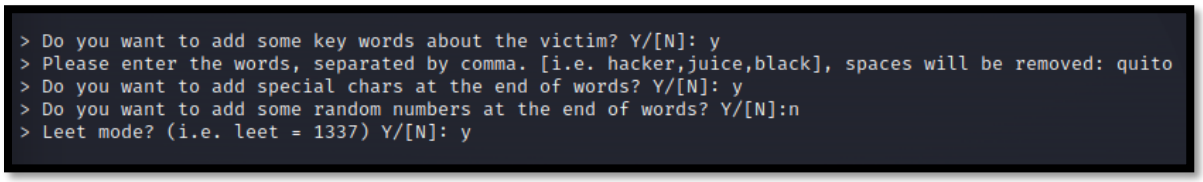
Preguntas realizadas por la aplicación para crear diccionario.



Una vez contestadas las preguntas, las cuales son realizadas por la aplicación y que son prestablecidas, se pregunta si se desea ingresar más información adicional, si se quiere agregar caracteres especiales y números aleatorios al final de las palabras con el finde tener una mayor posibilidad de crear una combinación de caracteres que sea idéntica a la contraseña.

Figura 21

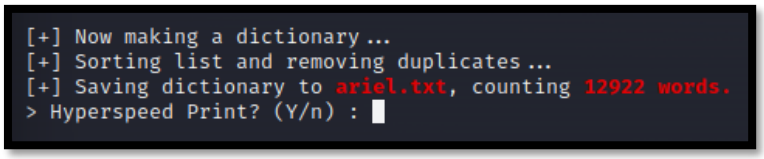
Preguntas para creación de diccionario.



Después de contestar las preguntas que realiza la aplicación y de confirmar que ya no se desea agregar más datos, se muestra el nombre del archivo de texto en el que se guardará el diccionario a crear y el número de palabras que contendrá dicho diccionario.

Figura 22

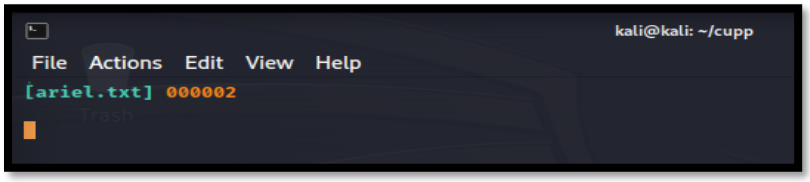
Especificaciones del diccionario a crear.



En la Figura 23 se muestra cómo se va creando las diferentes combinaciones de caracteres con las palabras ingresadas, este proceso puede tardar despendiendo de la cantidad de datos ingresos y el procesamiento de la computadora anfitriona del programa.

Figura 23

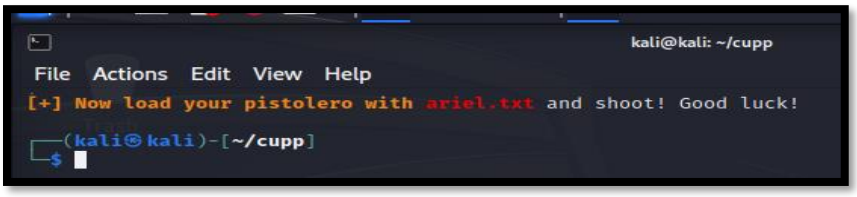
Creación del diccionario de datos.



Cuando se culmina de crear todas las posibles combinaciones con las palabras ingresadas en programa muestra el nombre del archivo que se definió posteriormente y se avisa que fue realizado con éxito como se observa en la Figura 24.

Figura 24

Fin creación de diccionario.



Para poder visualizar el diccionario de datos con todas las combinaciones creadas se puede ingresar el comando `$ cat ariel.txt` que abrirá el archivo de texto como se puede visualizar en la Figura 25.

Figura 25

Combinaciones creadas en la herramienta Cupp

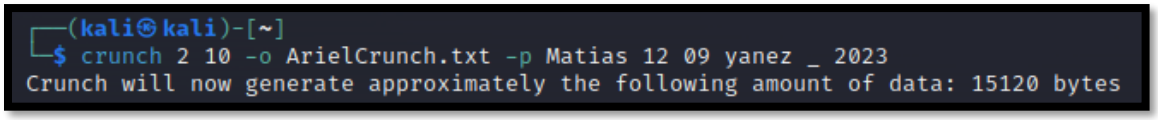
Otra herramienta que permite realizar diccionarios de datos es Crunch, y la principal diferencia es que en esta aplicación no se realizan preguntas específicas, es de total control que quien la utiliza, se recomienda utilizarla cuando se conoce palabras más específicas de la contraseña a descifrar.

En la Figura 26 se muestra cómo se debe especificar el número de palabras que se van a ingresar y el número de líneas que se desean, a continuación, se ingresan los caracteres que pueden ser palabras específicar, números o caracteres especiales, una vez ingresado todos los datos

necesarios el programa muestra el tamaño aproximado del documento en bytes o si es mucho más pesado en MB, GB, TB o PB.

Figura 26

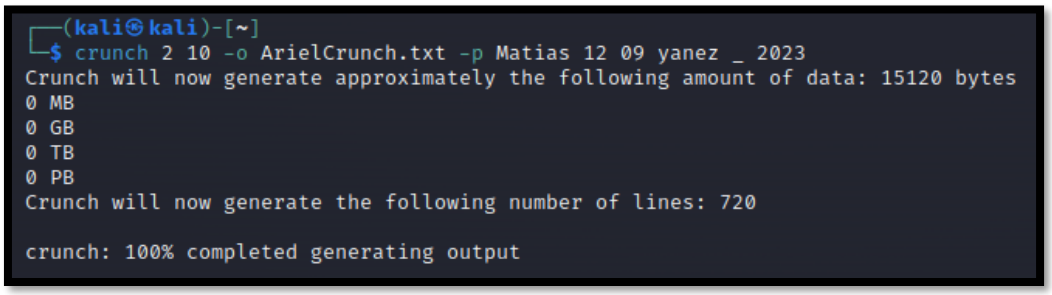
Aplicación Crunch



Una vez terminado de ingresar los datos y parámetros que tendrán el diccionario a construir, el programa Crunch muestra, que, con los 6 datos ingresados, el tamaño del archivo que se creara es de 15120 bytes y que contendrá una cantidad de 720 líneas o contraseñas creadas.

Figura 27

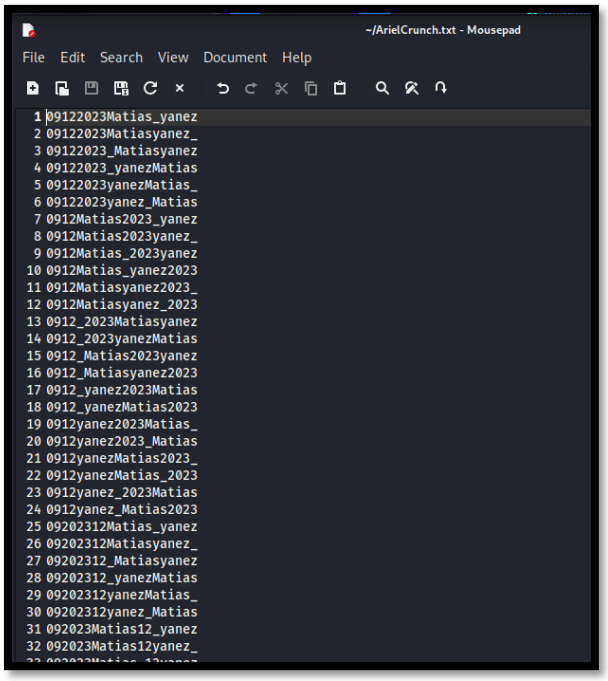
Creación de diccionario con la herramienta Crunch



En la Figura 28 se muestra todas las combinaciones con los datos que fueron ingresados y enumerados para conocer el número total de posibles contraseñas creadas, guardadas en un archivo de texto.

Figura 28

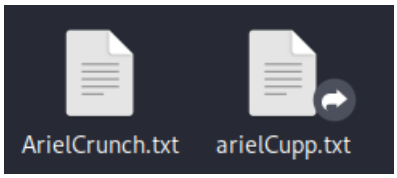
Lista de posibles contraseñas



Los dos diccionarios de contraseñas creados con las herramientas Crunch y Cupp se muestran en la Figura 29 cada uno en formato `.txt,` los cuales serán utilizados posteriormente para probar todas las combinaciones que se encuentran dentro de cada documento.

Figura 29

Diccionarios de contraseñas creados.



4.6.Comprobación de obtención de clave

Culminada la creación de los diccionarios de datos con las diferentes herramientas, se lleva a cabo la comprobación de las contraseñas generadas, para lo cual se utilizó el programa Hydra, el cual permite definir a que sitio web se enviara cada contraseña para su autentificación, pero por métodos de seguridad que tienen las aplicaciones solo se puede comprobar un número limitado de credenciales.

Para iniciar esta fase lo primero que se debe hacer es la instalación del programa a utilizar lo cual se muestra a continuación el proceso que se llevó acabo de instalación, configuración e inicio de la aplicación.

Figura 30

Instalación de paquetes Python, curl, wget, php.

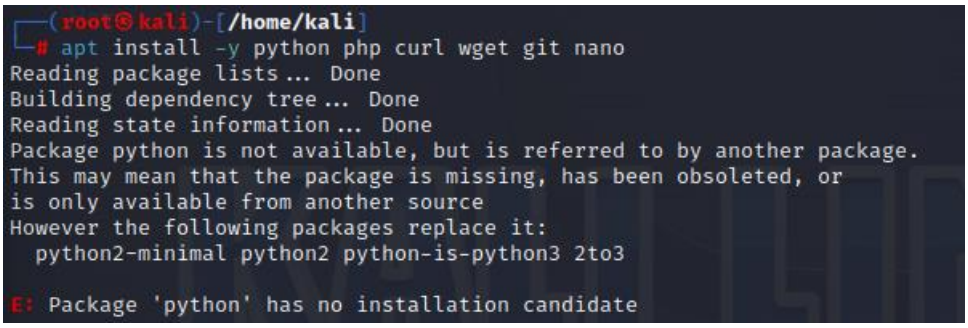


Figura 31

Acceso al directorio de inicio de Termux.

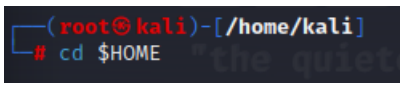


Figura 32

Clonación del enlace Hydra en termux/Kali.

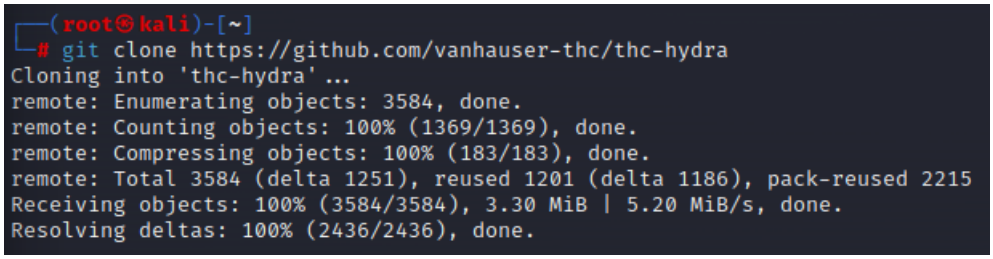


Figura 33

Acceso al repositorio Hydra clonado en el terminal.

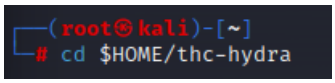


Figura 34

Instalación de paquetes usando script de configuración.

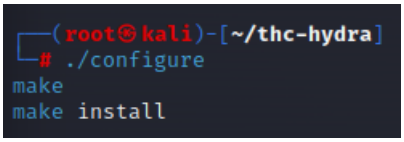
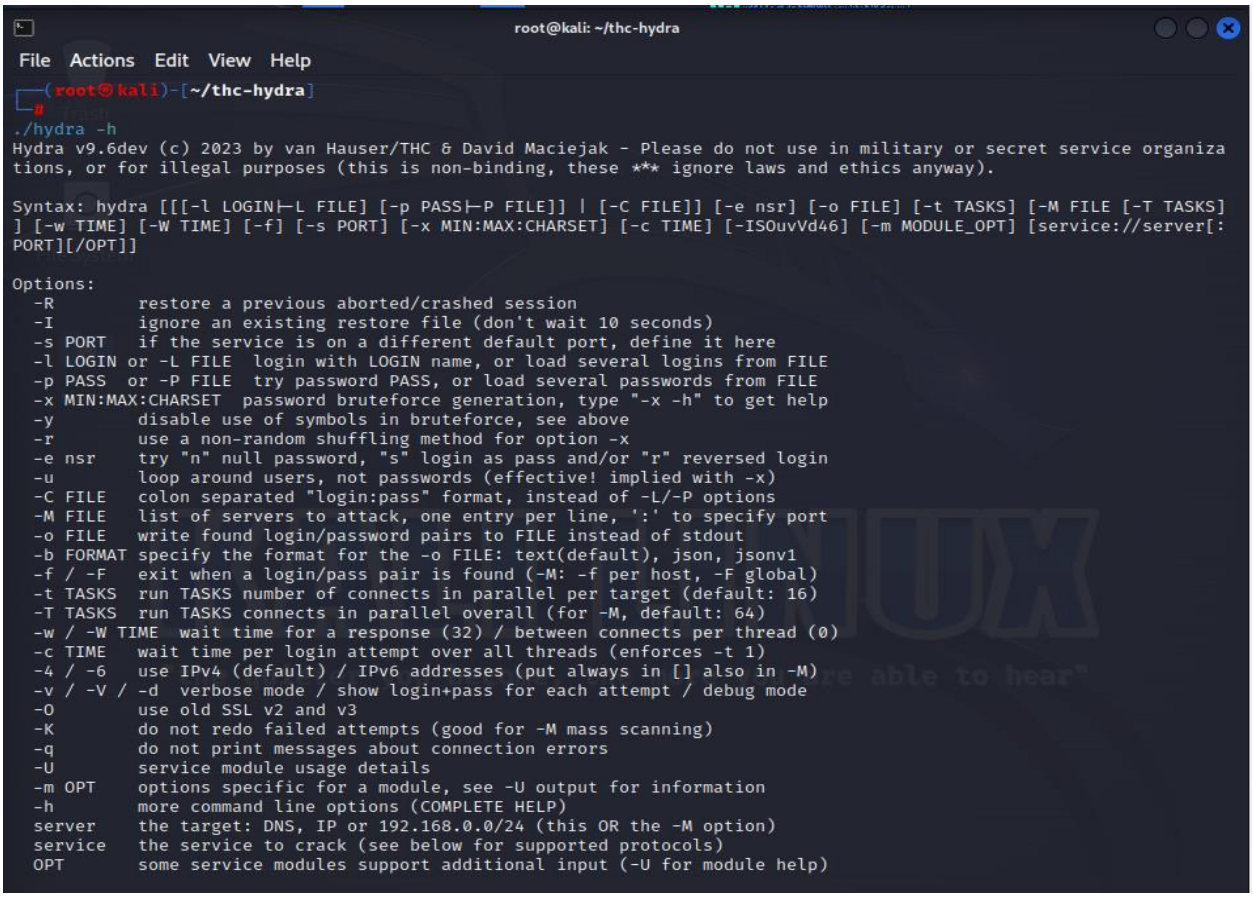


Figura 35

Instalación completa de Hydra.

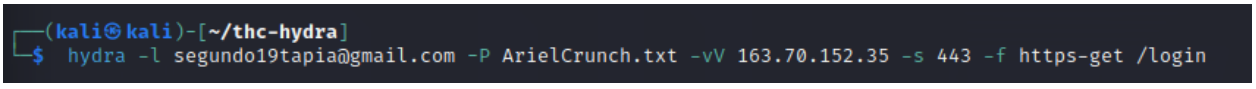


Una vez culminada la instalación del programa se puede comenzar a probar los diccionarios de contraseñas previamente creados.

Los parámetros que se usaron para poner en marcha el programa son `-l` para el nombre de usuario único, `-p` para indicar el fichero de lista de contraseñas, `-vV` para mostrar el intento por cada login más comprobación de contraseña realizado, después se indica la dirección IP de la red social, seguido por el comando `-s` que parametriza el puerto a utilizar y, por último `-f` para enlace del login como se observa en la figura 36.

Figura 36

Comando con parámetros para usar hydra



Se puede observar en la figura 37 que el programa ya se encuentra en ejecución y permite probar un máximo de 20 contraseñas por las seguridades que tiene implementada la red social Facebook, al culminar los intentos permitidos la herramienta muestra un mensaje de salida que finaliza el comando ingresado.

Figura 37

Comprobación de contraseñas

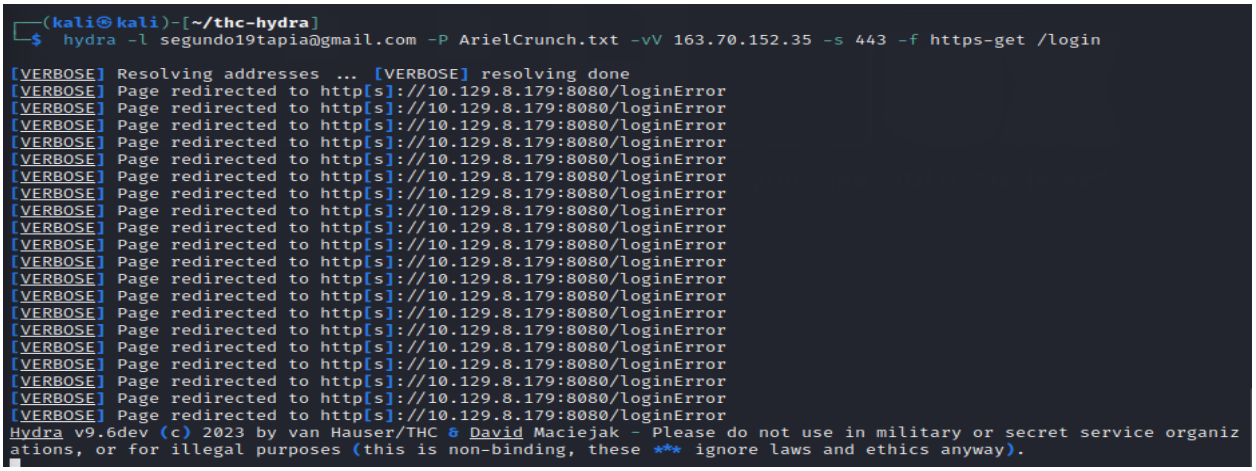
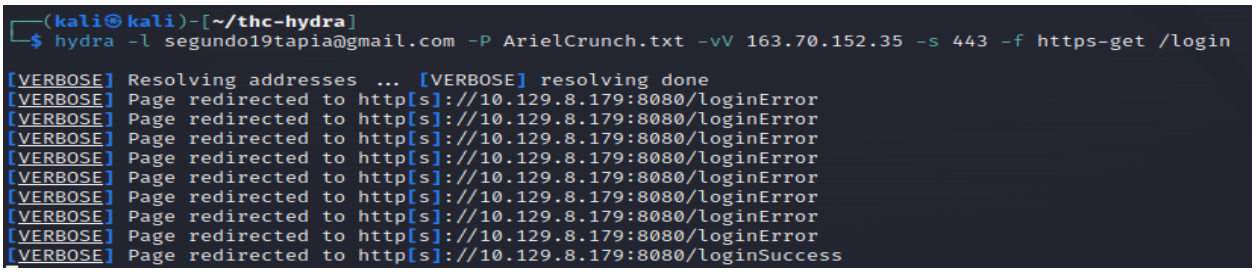


Figura 38

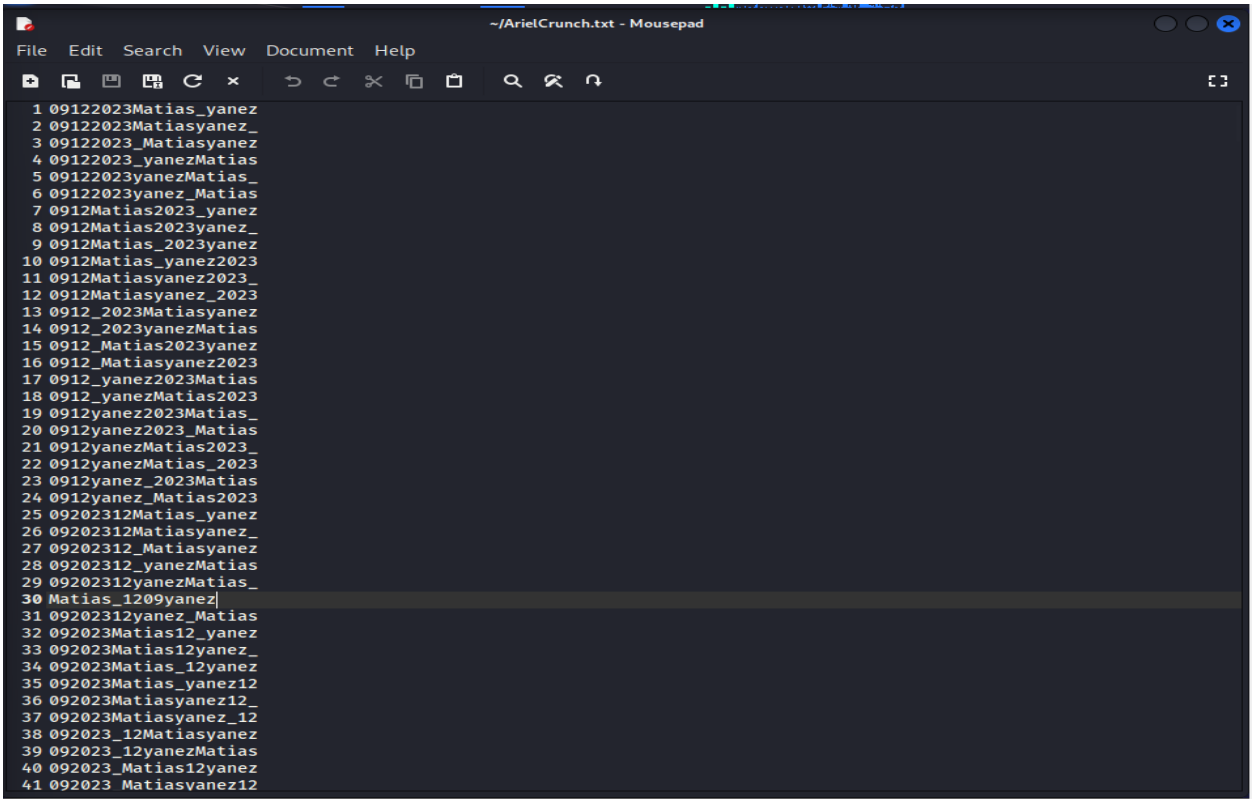
Segunda comprobación de contraseñas



Ya que solo se permite comprobar un bloque de 20 contraseñas se realizó el mismo comando por segunda vez, pero borrando las primeras credenciales ya probadas que se encuentran en el diccionario creado, en el intento número treinta se puede observar a la salida un loginSuccess lo que significa que se pudo acceder a la cuenta.

Figura 39

Identificación de contraseña en aplicación Crunch

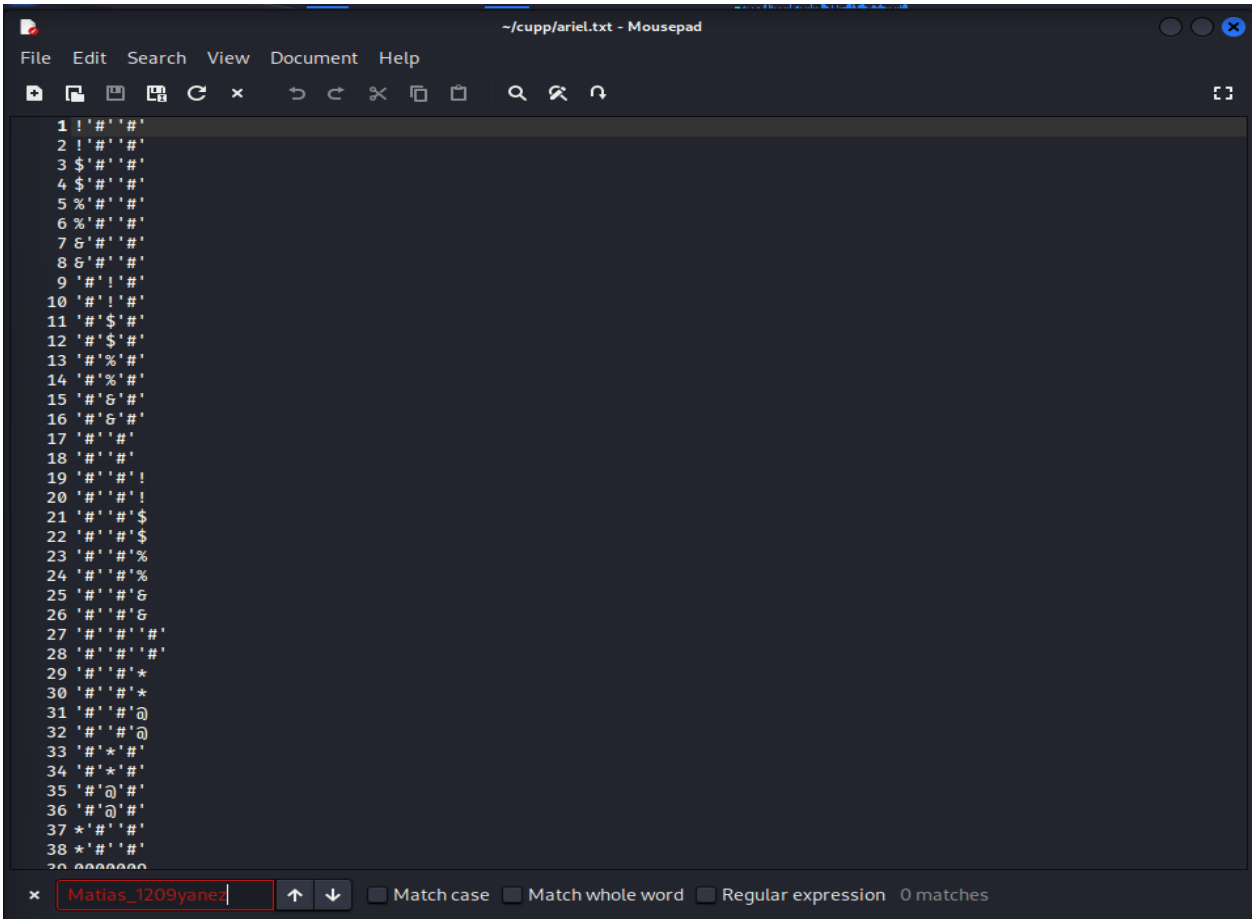


Como en la aplicación Hydra no muestra que numero de combinación es la que permite un login al usuario escogido se contó el número de intentos realizados y se verifico en el diccionario de datos cual es la contraseña correspondiente.

Una vez terminada el proceso de comprobación e identificación de la contraseña, en el diccionario creado con la herramienta Crunch se verifico su existe dentro del diccionario creado con la herramienta Cupp, como se visualiza en la figura 40 dentro de este repositorio no se logró generar esta contraseña.

Figura 40

Identificación de contraseña en aplicación Cupp



4.7.Implicaciones legales

Previo a poner en marcha un ataque de diccionario, fuerza bruta u otros ataques que atenten contra la integridad de los sistemas o accesos no autorizados, se debe considerar la tipificación de los delitos informáticos en el Ecuador.

En la Sección 3ra. “Delitos contra la seguridad de los activos de los sistemas de información y comunicación” del Código Orgánico Integral Penal (COIP) publicado en el Registro

Oficial de 10 de febrero de 2014 (Asamblea Nacional del Ecuador, 2014), se tipifican entre otros los siguientes delitos:

- Artículo 232.- Ataque a la integridad de sistemas informáticos, penas de tres a cinco años de privación de libertad, y si la infracción se comente a un servicio.
- Artículo 234.- Acceso no consentido a un sistema informático, telemático o de telecomunicaciones, con penas de tres a cinco años de privación de libertad.

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5. Conclusiones y recomendaciones

5.1. Conclusiones

El trabajo logra con éxito implementar un ataque de diccionario ya que, en uno de los archivos creados, con las combinaciones realizadas de palabras claves extraídas de las redes

sociales que fueron compartidas por el usuario, se observa que si se pudo obtener las credenciales creadas por el usuario.

Se planteo extraer información de las redes sociales, usando la técnica de web scraping pero al encontrase con las nuevas seguridades implementadas en las redes sociales, se cambió de estrategia para la obtención de información la cual consiste en tomar capturas de pantalla del perfil de la víctima donde se muestre información personal para por medio de un script transcribirlas a texto y poder manejar de mejor forma los datos extraídos.

Se pudo denotar que a partir de la información que es publica y compartida por el propio dueño de las diferentes redes sociales se puede obtener su contraseña, ya que por lo general esta debe ser fácil de recordada para poder ingresar los diferentes sitios web y su reciclaje o similitud para otras plataformas es una mala práctica que muchos de los usuarios de internet realizan muy a menudo.

5.2.Recomendaciones

Para poder generar un ataque más preciso reduciendo aún más el tiempo de prueba y las posibles contraseñas, se puede incorporar información más personal o sensible de la víctima y puesto a que en este trabajo se usa fases para convertir imagines a texto y después depurarlas agregar fotografías de la cedula de identidad, facturas, permisos de conducir o incluso de expedientes de la persona atacada ofrecería una mejor precisión para obtener una posible contraseña.

La metodología y el proceso que se platea y realiza para hacer un ataque de diccionario es únicamente con fines educativos y académicos, no se responsabiliza por el mal uso que se pueda llegar a darle a la información expuesta en este trabajo por parte de su autor.

Previo a poner en marcha un ataque de diccionario, fuerza bruta u otros ataques que atenten contra la integridad de los sistemas o accesos no autorizados, se debe considerar la tipificación de los delitos informáticos en el Ecuador.

Bibliografía

BID, & OEA. (2020). *Riesgos, avances y el camino a seguir en america latina y el caribe.* Obtenido de https://publications.iadb.org/publications/spanish/viewer/Reporte-

Ciberseguridad-2020-riesgos-avances-y-el-camino-a-seguir-en-America-Latina-y-el-Caribe.pdf

Castro, A. (15 de septiembre de 2023). *Internetizado*. Obtenido de Internetizado: https://www.internetizado.com/linux/kali

Dan, S. (05 de agosto de 2020). *CSOONLINE*. Obtenido de CSO: https://www.csoonline.com/article/569677/what-is-a-dictionary-attack-and-how-you-can-easily-stop-them.html

Dan, S. (05 de agosto de 2020). *CSOONLINE*. Obtenido de CSOONLINE: https://www.csoonline.com/article/569677/what-is-a-dictionary-attack-and-how-you-can-easily-stop-them.html

EHACKING. (9 de ENERO de 2019). *ehcgroup.io*. Obtenido de ehcgroup.io: https://blog.ehcgroup.io/2019/01/09/15/29/15/4518/como-utilizar-crunch-una-guia-completa/hacking/ehacking/

HACKWISE. (28 de SEPTIEMBRE de 2021). *HACKWISE.MX*. Obtenido de HACKWISE.MX: https://hackwise.mx/estas-son-las-tecnicas-mas-comunes-que-los-hackers-usan-para-descifrar-contrasenas/

Innovación, G. d. (2022). Vigilancia tecnológica en CIBERSEGURIDAD. *Tendencias tecnológicas.* Universidad Nacional Autónoma de México, Ciudad de México, Ciudad de Mexico. Recuperado el 12 de septiembre de 2022, de https://www.redinnovagro.in/pdfs/cyber.pdf

KALI.TOOLS. (30 de MARZO de 2021). *PENETRATION TESTING TOOLS*. Obtenido de PENETRATION TESTING TOOLS: https://en.kali.tools/?p=1305

Kemp, S. (2022). *Informe Global digital*. Obtenido de https://blog.hootsuite.com/es/informe-digital-estadisticas-de-redes-sociales/

KINSTA. (8 de octubre de 2020). *kinsta*. Obtenido de kintsa: https://kinsta.com/es/base-de-conocimiento/que-es-github/

LastPass. (2022). *Psychology of PASSWORDS.* LastPass. Obtenido de https://www.lastpass.com/-/media/3c627ed089e84bc39ca2bf6bf1d7cdec.pdf

Leighton, J. (2019). *Security Controls Evaluation, Testing, and Assessment Handbook.* Syngress.

LIVEJOURNAL. (18 de julio de 2019). *LIVEJOURNAL*. Obtenido de LIVEJOURNAL: https://deoxyt2.livejournal.com/118442.html

ORACLE. (31 de MAYO de 2022). *ORACLE*. Obtenido de ORACLE: https://developer.oracle.com/es/learn/technical-articles/what-is-python

Red Hat. (24 de enero de 2023). *Red Hat*. Obtenido de Red Hat: https://www.redhat.com/es/topics/open-source/what-is-open-source

Research Data Services. (7 de noviembre de 2019). Obtenido de Research Data Services: https://researchdata.wisc.edu/news/an-introduction-to-web-scraping-for-research/

Semymas. (20 de marzo de 2019). Riesgos de compartir información personal en redes sociales. *Riesgos de compartir información personal en redes sociales*. Obtenido de https://semymas.com/riesgos-informacion-personal/

Sicilia Piñero, M., Palazón Vidal, M., López López, I., & López Pérez, M. (2021). *Marketing en redes sociales.* Madrid: alphaeditorial.

Snider, K., Shandler, R., Zandani, S., & Canetti, D. (2021). Cyberattacks, cyber threats, and attitudes. *Journal of Cybersecurity*, 4-5.

Trepp, D. (26 de febrero de 2021). *ISACA*. Obtenido de ISACA: https://www.isaca.org/resources/isaca-journal/issues/2021/volume-2/the-gentle-art-of-password-management