PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE INFORMACIÓN

Trabajo de Titulación

GENERACIÓN DE UN DICCIONARIO DE DATOS RECOPILADOS A TRAVÉS DE LA APLICACIÓN DE HERRAMIENTAS DE RECONOCIMIENTO PASIVO DE LA INFORMACIÓN PÚBLICA, PARA PROPONER BUENAS PRÁCTICAS DE SEGURIDAD DE LA INFORMACIÓN

AUTOR:

Eduardo Andrés Padilla Sanipatín

Director: Ing. Beatriz Campos Villarroel, MBA.

QUITO DM, ENERO 2025

DEDICATORIA

Dedico esta tesis a mis padres, quienes me enseñaron el significado de esfuerzo y perseverancia, y me han apoyado en cada paso de este largo camino. A mis abuelitos, quienes siguen siendo mi fuente de inspiración para alcanzar mis metas cada día, a pesar de que ya no se encuentran conmigo. Y a Dios, por llenarme de bendiciones y darme la fortaleza para superar cada desafío.

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mis docentes, por su orientación académica, paciencia y conocimientos impartidos, que han sido fundamentales en mi formación. A mi tutora de tesis, por su apoyo constante y su guía a lo largo de este trabajo. A mi familia y novia, por su apoyo y amor incondicional en momentos difíciles. Todo esto ha sido fundamental para la conclusión de este proyecto. Estoy profundamente agradecido por tenerlos en mi vida.

RESUMEN

La presente investigación, tiene como objetivo principal generar un diccionario personalizado utilizando herramientas de reconocimiento pasivo y a partir de los resultados obtenidos, proponer estrategias de concientización y buenas prácticas para reducir la efectividad de los ataques de diccionario.

El estudio se centra exclusivamente en la recopilación de datos públicos disponibles en redes sociales y sitios web, sin acceder a información privada, asegurando el cumplimiento de principios éticos y legales. A lo largo del proyecto, se utilizaron herramientas de código abierto como Cupp, Crunch y Cewl, integradas en el sistema operativo Kali Linux, para generar diccionarios basados en la información pública recolectada.

La metodología aplicada combina un enfoque documental y experimental. La investigación documental permitió construir un sólido marco conceptual. Por su parte, la fase experimental incluyó la ejecución de pruebas prácticas en cuatro etapas principales:

- Extracción de información pública: Recolección de datos relevantes compartidos por los usuarios en redes sociales y sitios web.
- Creación de diccionarios: Generación de listas de contraseñas personalizadas utilizando las herramientas seleccionadas.
- Validación de contraseñas: Uso de un script en Python para probar las contraseñas generadas con un usuario ficticio, evaluando la efectividad de los diccionarios creados.
- Análisis y evaluación: Comparación de los resultados obtenidos por cada herramienta en términos de eficiencia y precisión en la generación de contraseñas.

Como resultado, se comprobó que las herramientas de reconocimiento pasivo pueden generar

diccionarios efectivos, siempre que cuenten con datos públicos suficientes. Sin embargo, se evidenció que la exposición de información personal en redes sociales incrementa significativamente el riesgo de ser víctima de ataques de este tipo.

Finalmente, se proponen buenas prácticas de seguridad de la información orientadas al usuario, como limitar la cantidad de datos personales compartidos públicamente, crear contraseñas robustas y utilizar medidas adicionales como la autenticación de dos factores.

Tabla de contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I – DENICIÓN DEL PROYECTO DE TITULACIÓN

1.1. JUSTIFICACIÓN

En la actualidad, la disponibilidad de información personal en internet ha incrementado la posibilidad de recopilar datos públicos para desarrollar ataques de diccionario. Los ataques de diccionario son un tipo de ataque de fuerza bruta en el que los hackers intentan adivinar la contraseña de un usuario a través de una lista de palabras, frases o combinaciones numéricas de uso frecuente (Kaspersky, ¿Qué es un ataque de diccionario?, 2023). Existen diversas técnicas para crear diccionarios para ataques de fuerza bruta. Una de las más comunes es la recopilación de información pública, a partir de la cual se generan listas de palabras basadas en términos frecuentes, como nombres y fechas importantes.

La facilidad para crear diccionarios se debe a la abundancia de información disponible en línea, la falta de conciencia sobre la privacidad y la existencia de herramientas open source que simplifican el proceso de los atacantes.

Entendiendo que el reconocimiento pasivo es una técnica utilizada para recopilar información de manera discreta, sin interactuar directamente con el usuario. Este reconocimiento se realiza aprovechando las fuentes públicas como los perfiles en redes sociales o cualquier información expuesta en internet que pueda ser recopilada sin alertar al usuario.

Esta investigación presentará propuestas de buenas prácticas de seguridad de la información, luego de analizar cómo se recopila y utiliza la información pública para crear diccionarios, resaltando la importancia de proteger la información publicada en línea.

1.2. PLANTEAMIENTO DEL PROBLEMA

En un entorno digital donde la información personal es compartida abiertamente a través de internet, los atacantes tienen la oportunidad de recolectar esta información pública para llevar a cabo ataques de diccionario basados en reconocimiento pasivo. Estos ataques aprovechan la información accesible para construir los diccionarios personalizados, poniendo en riesgo la seguridad de cuentas personales, profesionales y financieras de los usuarios.

El problema central radica en la falta de conocimiento y concientización sobre los riesgos de seguridad asociados con la exposición de información personal. Muchos usuarios continúan subestimando los peligros que representa la exposición de información en redes sociales, según estudios realizado por Kaspersky, los atacantes pueden utilizar datos como nombres, fechas de nacimiento, correos electrónicos y preferencias compartidas públicamente para generar patrones en los diccionarios, lo que aumenta las probabilidades de éxito en los ataques.

Al analizar esta temática, surgen las siguientes problemáticas: ¿Qué tipo de información es más susceptible a ser utilizada en los ataques de diccionario? ¿Qué estrategias de concientización pueden implementarse para mitigar estos riesgos? ¿Cuáles son las mejores prácticas de seguridad que se pueden implementar para minimizar estos ataques?

1.3. OBJETIVOS

1.3.1. GENERAL

Generar un diccionario de datos a través de la aplicación de herramientas de reconocimiento pasivo de la información pública, para proponer buenas prácticas de seguridad de la información.

1.3.2. ESPECÍFICOS

1. Investigar las funcionalidades, capacidades y limitaciones de las herramientas de reconocimiento pasivo.
2. Ejecutar las herramientas de generación de diccionarios open source, como Crunch, Cupp y Cewl.
3. Analizar los resultados obtenidos de las herramientas de generación de diccionarios.
4. Proponer estrategias de concientización y buenas prácticas para fortalecer la seguridad de las cuentas, minimizando el riesgo de accesos no autorizados.
5. Redactar conclusiones y recomendaciones.

1.4. ALCANCE

El presente estudio tiene como alcance la recopilación de información pública de usuarios en sitios web y redes sociales, utilizando técnicas de reconocimiento pasivo para crear diccionarios personalizados. Este proceso se llevará a cabo sin acceder a información privada, centrándose únicamente en datos públicos que el usuario ha compartido.

Se ejecutará las herramientas de generación de diccionarios como Crunch, Cupp y Cewl, para evaluar su eficiencia en la creación de contraseñas complejas. Se

considerarán aspectos como la longitud de las contraseñas, el uso de mayúsculas, minúsculas, números y caracteres especiales.

El estudio propondrá buenas prácticas de seguridad para minimizar la exposición de información pública, con el fin de reducir la efectividad de los ataques de diccionario.

CAPÍTULO II - MARCO TEÓRICO Y CONCEPTUAL

2.1. ANTECEDENTES Y MARCO REFERENCIAL

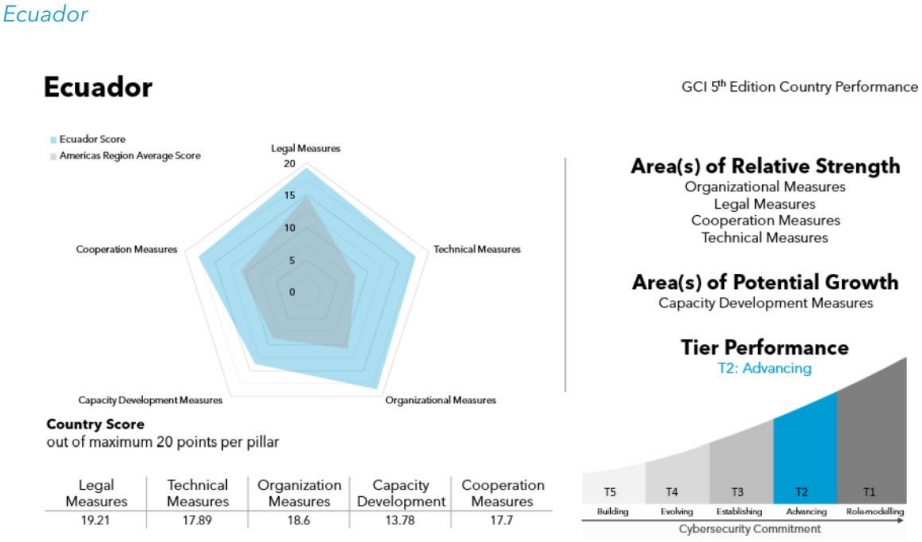
2.1.1. ANTECEDENTES EN ECUADOR

Según varios estudios de Movistar Empresas, Ecuador registró más de 12 millones de ciberataques en 2023. Aunque el número de ataques cibernéticos disminuyó en un 27% respecto a años anteriores, el panorama sigue siendo preocupante (TelefónicaEcuador, 2024).

No obstante, la quinta edición del Índice Global de Ciberseguridad (GCI), publicado en 2024 por la Unión Internacional de Telecomunicaciones (UIT), posicionó a Ecuador en el tercer lugar de ciberseguridad en América Latina, con una puntuación de 17.43/20. Esta calificación refleja mejoras en los pilares evaluados, como medidas legales, medidas de organización, medidas técnicas, medidas de cooperación y desarrollo de capacidades (Ambrissi, 2024).

Ilustración 1

Situación actual del Ecuador en 2024. Tomado de (ITU, 2024).



Notablemente, el marco legal del país se fortaleció, alcanzando un puntaje de

19.21 y consolidándose como el segundo mejor país de la región en términos de marcos legales. Además, Ecuador se destacó en las medidas organizacionales, con un puntaje de 18.6 que evidencia una clara coordinación entre los sectores públicos, privados y académicos para la implementación de estrategias de ciberseguridad.

A pesar de los progresos, todavía existen desafíos en el desarrollo de capacidades, donde Ecuador consiguió una puntuación de 13.78. Esta medida indica los esfuerzos en capacitación de ciberseguridad, esenciales para construir un ecosistema sólido en un entorno digital (Ambrissi, 2024).

El avance global de Ecuador en el GCI evidencia un compromiso mayor con la seguridad digital y una alineación con los estándares internacionales. Esto no solo fortalece la posición frente a ciberamenazas, sino que también mejora el atractivo para inversiones tecnológicas.

2.1.2. CASOS RELEVANTES

- Uno de los incidentes más recientes ocurrió en septiembre de 2019, se descubrió que un servidor administrado por Novaestrat, una empresa ecuatoriana de análisis de datos había dejado expuesto 18 GB de información personal de más de 20 millones de individuos, la mayoría de estos señalaban estar ubicados en Ecuador. Esta cifra es llamativa si se considera que en Ecuador la población es de aproximadamente 18 millones de habitantes. Sin embargo, esta diferencia se explica porque en bases de datos mal gestionadas, se encuentran datos duplicados, datos adicionales de personas fallecidas e información de personas extrajeras. Los atacantes accedieron a información sensible, como:
    - Nombres completos.
    - Números de identificación.

- Números de teléfono.
- Registros familiares.
- Historial educativo y laboral.
- Datos financieros.

El servidor en cuestión estaba ubicado en Miami y carecía de protocolos de seguridad básicos, lo que permitió que expertos de la empresa de seguridad informática vpnMentor descubrieran la vulnerabilidad en un mapeo web a gran escala. Los expertos identificaron una base de datos abierta al realizar un escaneo de puertos IP. Tras el descubrimiento, notificaron al equipo de seguridad informática de Ecuador, que inmediatamente restringió el acceso al servidor (BBC, 2019).

- El Banco del Austro fue víctima de un ciberataque en 2015, que resultó en la pérdida de 12 millones de dólares. Los atacantes lograron acceder a los sistemas del banco y realizar transferencias internacionales fraudulentas al comprometer el sistema SWIFT (Society for Worldwide Interbank Financial Telecommunication), una red internacional utilizada para realizar transferencias bancarias. Esto permitió a los atacantes acumular 12 millones, que se transfirieron a cuentas en Hong Kong, Dubái, Nueva York y Los Ángeles (TrendMicro, 2016).

2.2. SITUACIÓN ACTUAL EN EL MUNDO

La ciberseguridad se ha convertido en un elemento fundamental para garantizar la estabilidad y el desarrollo de las naciones en un mundo cada vez más interconectado. Según el índice de Ciberseguridad Global de la UIT, las amenazas más destacadas son los ataques de ransomware, dirigidos a servicios gubernamentales (Talayero, 2024). En general, la mayoría de los países se encuentran trabajando para fortalecer sus acciones en el área de ciberseguridad y en todas las regiones hay países con altos y pobres

resultados.

En términos de medidas legales, tiende a ser el área de ciberseguridad más desarrollada en los países, las leyes sobre privacidad están aumentando, más del 90% de los países tienen al menos una normativa sobre la protección de datos personales o protección de intimidad. Sin embargo, en el informe se presentó la necesidad de una mayor coordinación en los esfuerzos de ciberseguridad, en cuanto a la protección de la infancia, la aplicación de estrategias sigue siendo limitada a nivel global (Talayero, 2024).

Los esfuerzos de formación y concientización en materia de ciberseguridad varían de una región a otra, especialmente en el intento de fortalecer esta área. A pesar de los distintos avances, en muchos países faltan iniciativas educativas que impulsen el desarrollo de conocimientos en ciberseguridad desde etapas tempranas hasta niveles avanzados.

2.3. MARCO CONCEPTUAL

2.3.1. RECONOCIMIENTO PASIVO

El reconocimiento pasivo es una técnica en el ámbito de la ciberseguridad, utilizado para recopilar información de un usuario sin interactuar directamente con él. Esto se logra a través de fuentes públicas como redes sociales y sitios web. A diferencia del reconocimiento activo, que, de igual manera, es una técnica de recolección de información, pero se requiere interactuar directamente con los sistemas, redes o usuarios (Vectra, 2024).

El propósito principal de realizar un reconocimiento pasivo a los usuarios es identificar patrones de comportamiento y relaciones personales o profesionales. Esto permite a los atacantes identificar posibles riesgos con la exposición excesiva de

información personal en línea, que podría ser utilizada para realizar ataques de diccionario, ingeniería social o robo de identidad.

2.3.2. ATAQUES DE DICCIONARIO

Los ataques de diccionario son un tipo de ataque de fuerza bruta utilizada por los atacantes para descifrar contraseñas mediante el uso de listas de palabras, frases o combinaciones numéricas conocidas como “diccionarios”. Los diccionarios pueden ser generados manualmente o creados mediante herramientas que toman en cuenta información personal del usuario, estas mismas optimizan el proceso de creación de contraseñas.

Cuando un ataque de diccionario consigue descifrar una contraseña, los atacantes pueden utilizarla para acceder a cuentas bancarias, perfiles de redes sociales o archivos protegidos por contraseña (Kaspersky, ¿Qué es un ataque de diccionario?, 2023). Este tipo de ataque es muy efectivo cuando las contraseñas no cumplen con los requisitos básicos, como incluir caracteres especiales, números y letras mayúsculas. Por ese motivo, la prevención de estos ataques se basa en implementar contraseñas robustas que cumplan todos los estándares mínimos de seguridad.

2.3.3. DICCIONARIO DE CONTRASEÑAS

Un diccionario de contraseñas es una lista de palabras, frases o combinaciones de caracteres que los atacantes utilizan para realizar ataques de diccionario, con el objetivo de descifrar una contraseña. Estas listas de palabras suelen estar diseñadas para probar contraseñas comunes y pueden incluir:

- Contraseñas extraídas de bases de datos hackeadas.
- Palabras comunes utilizadas en la vida cotidiana.

- Información recopilada del usuario.
- Combinaciones comunes que los usuarios tienden a elegir, como “12345”, “password”, “admin”, entre otras.

2.3.4. OPEN SOURCE

La expresión open source, código abierto en español, es un código diseñado de manera que sea accesible al público. Esto permite que cualquier persona pueda ver, modificar y distribuir el código del software, ya sea para personalizarlo, mejorarlo o adaptarlo a las necesidades.

El software open source se desarrolla de manera descentralizada y colaborativa, así que depende de la revisión y producción de la comunidad. Además, suele ser una alternativa más económica que los softwares propietarios, ya que los encargados de su desarrollo son las comunidades y no solo un autor o una sola empresa (RedHat, ¿Qué es el open source?, 2023).

2.3.5. CRUNCH

Crunch es una herramienta open source incluida en el sistema operativo Kali Linux, diseñada para generar diccionarios personalizados utilizados en ataques de fuerza bruta. Crunch permite a los usuarios crear listas de palabras basadas en parámetros específicos, como longitud mínima y máxima, conjunto de caracteres y parámetros predefinidos, esto permite generar una lista extensa de palabras para poder utilizar (elcursodelhacker, 2022).

2.3.6. CUPP

Cupp, por sus siglas en inglés Common User Passwords Profiler, es una herramienta open source incluida en Kali Linux, diseñada para generar listas de

contraseñas personalizadas utilizada principalmente en auditorías de seguridad y pruebas de penetración. Esta herramienta crea diccionarios con información específica de un usuario, como nombres, apodos, fecha de nacimiento, hobbies, entre otros datos personales. Cupp se fundamenta en el principio de que las contraseñas suelen ser a menudo, combinaciones de cosas significativas del usuario (Esgeeks, 2018).

2.3.7. CEWL

Cewl, por sus siglas en inglés Custom Word List Generator, es una herramienta open source incluida en Kali Linux, diseñada para generar listas de palabras a partir del contenido de páginas web. Utiliza un enfoque simple pero efectivo para recopilar palabras clave que podrían ser útiles para realizar un ataque de diccionario, como nombres de usuario, direcciones de correo electrónico o términos relacionados con el objetivo.

2.3.8. PYTHON

Python es un lenguaje de programación de alto nivel, ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Python se caracteriza por su sintaxis clara, lo que permite a sus desarrolladores escribir código de forma rápida y eficiente, y es fácil de aprender (AWS, ¿Qué es Python?, 2022).

Una de las principales ventajas de Python es su extensa biblioteca, que incluye diversos módulos para interactuar con el sistema operativo, trabajar con bases de datos, manejar datos en formatos como JSON y XML, y realizar operaciones matemáticas. En esta investigación, Python es elegido por su capacidad para adaptarse a una amplia gama de aplicaciones, por la comunidad extensa que proporciona recursos, bibliotecas y

frameworks de calidad, y su capacidad de integración con otros lenguajes de programación y tecnologías.

2.3.9. MÉTODOS DE CONCIENTIZACIÓN

La concientización es un método, que se plantea desde el campo educativo, con la idea de promover una conciencia crítica y un análisis, que permita comprender problemáticas y guiar acciones (Tomazella, 2020). Enfocándonos en el ámbito tecnológico, los métodos de concientización se enfocan en educar a los usuarios y organizaciones sobre prácticas seguras. La concientización sobre los riesgos asociados a la exposición de información pública es crucial para mitigar los ataques de diccionario.

Adoptar estrategias de concientización marca la diferencia en la protección de los usuarios frente a este tipo de amenazas. Los métodos efectivos de concientización no solo buscan informar, sino que también promueven la práctica de hábitos seguros y el desarrollo de una mentalidad preventiva.

2.3.10. KALI LINUX

Kali Linux es una distribución de Linux basada en Debian, es una herramienta utilizada por profesionales de seguridad informática para llevar a cabo pruebas de penetración, análisis de seguridad, auditorías de redes y análisis forense digital (Imaginaformacion, 2024).

Kali Linux es conocido por su flexibilidad y facilidad de uso, ofreciendo aproximadamente 600 herramientas especializadas en seguridad informática. Esta amplia variedad de herramientas lo convierte en una opción ideal para auditores, expertos de ciberseguridad y hackers éticos (Imagina, 2024). Además, es ampliamente conocido por su capacidad para realizar actividades como el descifrado de contraseñas,

gracias a las diversas herramientas incorporadas.

2.3.11. GITHUB

GitHub es una plataforma de desarrollo colaborativo muy popular entre los desarrolladores de software, ya que permite almacenar, compartir y trabajar en proyectos compartidos. GitHub aloja un sistema de control de versiones (VCS) llamado Git. Éste permite comparar el código de un archivo para ver las diferencias entre versiones, restaurar versiones antiguas y funcionar los cambios de distintas versiones (Hostinger, 2023).

La plataforma de GitHub es una de las más utilizadas en el mundo por los desarrolladores de software por su función pull requests, que permiten a los equipos trabajar juntos de manera más efectiva. Aloja más de 100 millones de repositorios y la mayoría son proyectos de código abierto.

2.3.12. SISTEMA OPERATIVO

El sistema operativo es un software que actúa como intermediario entre el hardware de una computadora y las aplicaciones. Su principal función es administrar y controlar todas las aplicaciones que utiliza el usuario en una computadora. Actualmente, los sistemas operativos más utilizados son Windows, Linux y macOS (Concepto, 2020). Además, existen sistemas operativos especializados para dispositivos móviles, como Android y iOS y sistemas operativos para servidores, como Ubuntu Server o Windows Server, que están diseñados para satisfacer necesidades específicas.

Los sistemas operativos proporcionan interfaces gráficas que permite a los usuarios interactuar con el equipo, ya sea a través de comandos en una consola o de un entorno gráfico. Entre las principales tareas de un sistema operativo se incluyen la

administración de procesos, la gestión de archivos, el control de dispositivos y la seguridad del sistema. También son responsables de la gestión de recursos como la memoria RAM y la CPU del equipo, permitiendo que múltiples programas se ejecuten de manera simultánea.

2.3.13. CIBERSEGURIDAD

La ciberseguridad se refiere a todas las tecnologías, prácticas y políticas para prevenir los ciberataques. La ciberseguridad tiene como objetivo proteger los sistemas informáticos, las aplicaciones, los dispositivos, los datos, los activos financieros y las personas contra el ransomware y otros malware, las estafas de phishing, el robo de datos y otras ciberamenazas (IBM, 2024).

Además de centrarse en la protección contra amenazas, la ciberseguridad abarca áreas fundamentales como la seguridad de redes, la seguridad en la nube, la protección de datos personal y empresariales, y la seguridad de aplicaciones. En la actualidad, la ciberseguridad no solo es crucial para las empresas y los gobiernos, sino también para los usuarios individuales, quienes deben ser conscientes de los riesgos digitales. La educación y concientización juegan un papel fundamental para reducir vulnerabilidades.

2.3.14. ATAQUE DE FUERZA BRUTA

Un ataque de fuerza bruta es una técnica que utiliza pruebas y errores para descifrar contraseñas, credenciales de inicio de sesión y claves de cifrado. Este tipo de ataque se basa en la capacidad del atacante para probar múltiples combinaciones de forma rápida, aprovechando herramientas especificas diseñadas para generar combinaciones (Fortinet, 2017).

Este método puede ser efectivo, cuando las contraseñas de los usuarios son

débiles o predecibles, donde el atacante intenta con varios nombres de usuario y contraseñas, hasta lograr descifrar la información correcta de inicio de sesión.

2.3.15. ÍNDICE GLOBAL DE CIBERSEGURIDAD

GCI, por sus siglas en inglés Índice Global de Ciberseguridad, es una iniciativa desarrollada por la Unión Internacional de Telecomunicaciones (UIT) que mide el compromiso con la ciberseguridad de 194 países a nivel mundial, según cinco pilares: medidas legales, medidas técnicas, medidas organizativas, desarrollo de capacidades y cooperación (AgenciaGUB, 2024). El objetivo principal de la GCI es proporcionar una visión del panorama global de la ciberseguridad, identificar áreas de mejora y fomentar un enfoque colaborativo entre las naciones.

El GCI no solo es una herramienta para comparar el desempeño de los países, sino que también promueve la implementación de buenas prácticas y estrategias para abordar las amenazas cibernéticas.

2.3.16. INGENIERÍA SOCIAL

La ingeniería social es una técnica de manipulación que aprovecha el error humano para obtener información privada, acceso a sistemas u objetos de valor. Esta técnica hace que los usuarios desprevenidos expongan datos, propaguen infecciones de malware o den acceso a sistemas restringidos. Los ataques pueden ocurrir en línea o en persona (Kaspersky, ¿Qué es la ingeniería social?, 2017).

Algunos métodos comunes de la ingeniería social incluyen el phishing, que consiste en correos electrónicos fraudulentos, el pretexting, que implica la creación de historias falsas para obtener información y el baiting, que se basa en el uso de recompensas falsas para atraer a las víctimas. La prevención de este tipo de ataque

requiere que los usuarios se encuentren bien informados y capacitados para identificar señales de manipulación. Esto incluye reconocer correos electrónicos sospechosos, evitar proporcionar información personal o empresarial sin verificar la legitimidad del solicitante y ser precavido ante ofertas o recompensas inesperadas.

2.3.17. RANSOMWARE

El ransomware es una clase de malware que representa un riesgo para los dispositivos tecnológicos. El término comienza con la palabra “ransom”, que significa “rescate”. El ransomware es un software extorsivo, su finalidad es impedir que el usuario use su dispositivo hasta que haya pagado un rescate (Kaspersky, El ransomware: qué es, cómo se lo evita, cómo se elimina, 2018). Este ataque puede afectar tanto a personas como a empresas e incluso a infraestructuras y suele iniciarse a través de correos electrónicos maliciosos o descarga de archivos engañosos.

Los ataques de ransomware no solo comprometen la disponibilidad de los datos, sino que también suelen amenazar con divulgar información sensible si no se paga el rescate. Para prevenir este tipo de ataque, es importante tomar medidas como realizar copias de seguridad frecuentes, mantener el sistema operativo siempre actualizado, usar herramientas que detecten amenazas y capacitar a los usuarios para que puedan reconocer correos fraudulentos o engaños de ingeniería social.

2.3.18. HIPERVISOR

Un hipervisor es un software que se utiliza para ejecutar máquinas virtuales. Todas las máquinas virtuales disponen de un sistema operativo y de aplicaciones propias. El hipervisor asigna los recursos físicos, como el procesamiento, la memoria y el almacenamiento, a las máquinas virtuales según sea necesario (AWS, ¿Qué es un

hipervisor?, 2023).

Existen dos tipos de hipervisores: Hipervisor de tipo1 y tipo 2.

El hipervisor de tipo 1, también conocido como servidor dedicado, se instala directamente sobre el hardware físico del servidor. No necesita un sistema operativo subyacente para funcionar. Algunos hipervisores populares son VMware ESXI, Microsoft Hyper-V y XEN. Este tipo de hipervisor suele usarse más en los centros de datos empresariales o en entornos que se basan en servidores y son conocidos por su eficiencia y seguridad, ya que tiene acceso directo a los recursos del hardware. El hipervisor de tipo 2, también conocido como hipervisor alojado, se ejecuta sobre un sistema operativo anfitrión, como si fuera una aplicación más. Algunos de los hipervisores más conocidos son VMware Workstation y Oracle VirtualBox (RedHat, ¿Qué es un hipervisor?, 2023).

CAPÍTULO III – ATAQUES DE DICCIONARIO

3.1. ATAQUE DE DICCIONARIO

Como se explicó en el marco conceptual, un ataque de diccionario es un método que trata de adivinar contraseñas usando palabras comunes hasta dar con la contraseña correcta. A diferencia de los ataques de fuerza bruta, donde un programa introduce automáticamente combinaciones de letras, símbolos y números de forma totalmente aleatoria, los ataques de diccionario utilizan listas de palabras predefinidas. Estas listas suelen generarse a partir de bases de datos de contraseñas filtradas, palabras frecuentemente utilizadas o términos personalizados con un objetivo específico (Grigas, 2022).

Los ataques de diccionario son en esencia ataques de fuerza bruta, pero con un enfoque más dirigido. Al usar listas predefinidas, el proceso se vuelve más eficaz, ya que no necesita tantos intentos y combinaciones. Por otro lado, si la contraseña de los usuarios cumple con los requisitos mínimos de seguridad, el ataque de diccionario no surtirá efecto. Pues, los ataques de diccionario rara vez tienen éxito contra sistemas protegidos que aplican medidas como la autenticación multifactor o bloqueos tras varios intentos fallidos.

3.2. METODOLOGÍA APLICADA

Para llevar a cabo el desarrollo del trabajo, se utilizó una combinación de técnicas de investigación documental y experimental. Estas técnicas se encargan de recopilar, analizar y seleccionar información a través de diversos documentos, como libros, revistas, grabaciones, sitios web, periódicos, bibliografías, entre otros (Ortega, 2019). En la primera fase, la técnica de investigación documental permitió recopilar y analizar información relacionada con la seguridad de la información, ataques de diccionario y

herramientas de reconocimiento pasivo. Este proceso ayudó a construir un marco conceptual sólido y a identificar métodos y herramientas relevantes para el desarrollo del diccionario.

La etapa experimental se centró en la aplicación de herramientas de reconocimiento pasivo, como Cewl, Crunch y Cupp, todas estas integradas en Kali Linux. Estas herramientas fueron utilizadas para crear diccionarios personalizados, a partir de recolectar información pública disponible en perfiles de redes sociales y otras fuentes, como nombres, fechas, palabras clave y patrones que pueden ser utilizados para la construcción del diccionario.

Finalmente, se empleó un enfoque analítico para procesar los datos recopilados. Este proceso consistió en evaluar los diccionarios generados por cada herramienta, analizando su eficiencia en cuanto a la longitud de contraseñas, el uso de mayúsculas, minúsculas, números y caracteres especiales. A partir de los resultados obtenidos, generar propuestas de buenas prácticas destinadas a mejorar la protección de la información pública y prevenir posibles ataques de diccionario.

3.3. PROCESO DE ATAQUE DE DICCIONARIO

El ataque de diccionario es un método sistemático que sigue una serie de pasos para intentar descifrar contraseñas utilizando listas de palabras o combinaciones comunes. Este proceso inicia con la recopilación de información del usuario, como nombres, fechas importantes o datos disponibles en redes sociales. Seguidamente, se genera un diccionario con la información recopilada que incluye palabras comunes o combinaciones frecuentes.

Con un diccionario listo, el atacante utiliza herramientas automatizadas que se encargan de intentar el inicio de sesión con cada contraseña del diccionario. Este proceso

automatizado permite a los atacantes probar miles de contraseñas en poco tiempo, aumentando las probabilidades de éxito. Si la contraseña llega a coincidir, los atacantes lograrán tener acceso no autorizado, de lo contrario, el ataque fracasa y da lugar a explorar otros métodos adicionales.

Para aumentar la efectividad, los atacantes suelen incluir en los diccionarios caracteres especiales, combinaciones de palabras, uso de números y palabras relacionadas con el usuario. Esto hace que los ataques de diccionario sean una amenaza constante, aunque fácilmente mitigable mediante el uso de contraseñas robustas y medidas de seguridad adicionales.

3.4. MEDIDAS DE PROTECCIÓN

Protegerse contra ataques de diccionario requiere adoptar buenas prácticas que fortalezcan la seguridad de la información. Una de las medidas esenciales es elegir contraseñas robustas que combinen letras mayúsculas y minúsculas, números y caracteres especiales, evitando cualquier relación con datos personales. Además, es importante cambiar las contraseñas con regularidad, tanto en dispositivos electrónicos como en cuentas de redes sociales, para reducir la exposición a posibles ataques (Oliveira, 2023).

También es importante no reutilizar contraseñas en diversas cuentas, ya que hacerlo aumenta el riesgo si una de ellas es comprometida. Limitar la cantidad de información que se comparte en redes sociales es otra medida clave, ya que los atacantes suelen usar datos públicos para generar diccionarios personalizados.

3.5. TIPOS DE ATAQUES DE DICCIONARIO

3.5.1. ATAQUE DE DICCIONARIO SIMPLE

El ataque de diccionario simple es un método de descifrado de contraseñas que se basa en probar una lista de contraseñas previamente conocidas. En lugar de intentar todas las combinaciones posibles como en el ataque de fuerza bruta, este método utiliza diccionarios predefinidos que contienen combinaciones sencillas y contraseñas filtradas con anterioridad. Esto hace que sea más rápido y eficiente que otros métodos, especialmente contra contraseñas débiles o predecibles (TokioSchool, Ataque de diccionario simple, 2023).

Sin embargo, la efectividad del ataque de diccionario simple disminuye frente a contraseñas complejas que incluyen caracteres especiales o no están basadas en palabras comunes. Medidas como el uso de contraseñas aleatorias y no relacionadas con información personal pueden proteger eficazmente contra este tipo de ataques. Este tipo de ataque sencillo sigue siendo común debido a su rapidez y facilidad de implementación.

3.5.2. ATAQUE DE DICCIONARIO PERSONALIZADO

El ataque de diccionario personalizado es una técnica que aprovecha información específica del usuario para aumentar las probabilidades de éxito en descifrar la contraseña. A diferencia de otros tipos de ataque de diccionario, este se centra en recopilar datos personales, como nombres de familiares, mascotas, fechas de nacimiento, lugares favoritos o incluso referencias públicas obtenidas en redes sociales. Con esta información, los atacantes generan un diccionario personalizado con combinaciones que

reflejan detalles particulares de la vida del usuario (TokioSchool, Ataque de diccionario personalizado, 2023).

Este método es efectivo cuando los usuarios basan sus contraseñas en datos personales que son fáciles de recordar. Los atacantes, al trabajar con un conjunto pequeño de contraseñas incrementan significativamente las posibilidades de éxito. Para evitar este tipo de ataque, es crucial evitar el uso de información personal en las contraseñas y optar por combinaciones aleatorias y robustas que no puedan ser descifradas a partir de los datos que se encuentran públicamente.

3.5.3. ATAQUE DE DICCIONARIO HÍBRIDO

El ataque de diccionario híbrido combina las técnicas de los ataques de diccionario simple con elementos adicionales como números, caracteres especiales u otros elementos. Este método permite a los atacantes realizar modificaciones en las contraseñas del diccionario, como sustituir letras por números similares, por ejemplo, “h3ll0” en lugar de “hello” o añadir combinaciones como “123” o “!”. Estas modificaciones amplían la cantidad de posibles contraseñas correctas, aumentando la probabilidad de éxito (TokioSchool, Ataque de diccionario híbrido, 2023).

Una de las ventajas de este ataque es su capacidad para adaptarse a contraseñas comunes de los usuarios, quienes suelen mezclar contraseñas simples con números o caracteres especiales para hacerlas más seguras. Sin embargo, las contraseñas complejas que combinan longitud, aleatoriedad y variedad de caracteres, siguen siendo un desafío para los atacantes. A pesar de sus limitaciones, los ataques híbridos son efectivos contra

contraseñas débiles o moderadamente complejas y son ampliamente utilizados por los atacantes debido a su velocidad y éxito.

3.5.4. ATAQUE DE DICCIONARIO AVANZADO

El ataque de diccionario avanzado utiliza algoritmos y estrategias sofisticadas para generar combinaciones de contraseñas, lo que lo hace más eficiente que los métodos simples o híbridos. Los atacantes implementan técnicas como reglas gramaticales, sustituciones inteligentes de letras, por ejemplo, cambiar "o" por "0" o "a" por "@" y patrones predefinidos basados en combinaciones y estructuras habituales utilizadas por los usuarios al crear contraseñas. Estos métodos permiten a los atacantes descifrar combinaciones más complejas sin probar todas las posibilidades como en un ataque de fuerza bruta (TokioSchool, Ataque de diccionario avanzado, 2023).

Este método es efectivo porque puede descifrar contraseñas que parecen seguras, pero en realidad siguen patrones que son fáciles de identificar. Por ejemplo, contraseñas basadas en secuencias de teclado o en palabras predecibles. Al emplear diccionarios más extensos y algoritmos avanzados, este tipo de ataque incrementa significativamente su precisión, representando una amenaza crítica para la seguridad de contraseñas que no sean completamente aleatorias y únicas.

3.6. COMPARACIÓN DE ATAQUES DE DICCIONARIO

Tabla 1

Comparativa entre ataques de diccionario.

Tipo de ataque	Características clave	Estrategias utilizadas	Ventajas	Limitaciones	Eficacia

Diccionario Simple	-Utiliza palabras comunes. -Utiliza contraseñas filtradas con anterioridad.	-Uso de listas de palabras predefinidas, como diccionarios básicos.	-Rápido para contraseñas simples. -Requiere pocos recursos computacionales.	-Ineficaz contra contraseñas complejas o aquellas que no contienen palabras comunes. -Depende de un diccionario limitado.	Baja: solo funciona con contraseñas débiles que contienen palabras comunes.
Diccionario Personalizado	-Ocupa información personal.	-Creación de un diccionario adaptado a la víctima, basado en información pública y privada.	-Altamente efectivo si la víctima utiliza información personal en sus contraseñas. -Menor cantidad de intentos necesarios.	-Requiere una etapa previa de recolección de información, lo que consume tiempo. -Ineficaz contra contraseñas no relacionadas con la víctima.	Alta: muy efectivo si la víctima usa información personal en sus contraseñas.
Diccionario Híbrido	-Sustitución de letras por números o caracteres especiales. -Agregar	-Combinación de palabras del diccionario con variaciones.	-Mayor probabilidad de éxito con contraseñas que usan modificacione	-No tan eficaz con contraseñas únicas o con combinaciones muy	Moderada: puede romper contraseñas con modificacio

	sufijos/prefijos comunes.		s predecibles.	aleatorias de caracteres.	nes predecibles, pero no con combinaciones únicas.
Diccionario Avanzado	-Emplea secuencias de teclado. -Emplea patrones de gramática.	-Uso de algoritmos complejos para generar combinaciones avanzadas. -Aplicación de reglas dinámicas y patrones comunes.	-Capaz de superar contraseñas complejas con patrones predecibles. -Más eficiente que otros métodos contra contraseñas robustas.	-Consume más recursos computacionales. -Puede ser ineficaz con contraseñas verdaderamente aleatorias y únicas.	Alta: Sobresale con contraseñas complejas que poseen patrones reconocibles o tendencias comunes.

El análisis comparativo de los distintos ataques de diccionario demuestra que cada método tiene sus fortalezas y debilidades y su eficiencia depende, en gran medida, de los patrones que los usuarios ocupan en sus contraseñas. Para el desarrollo de este trabajo, se empleará un diccionario personalizado basado en la recopilación de información pública, aprovechando datos específicos de la víctima para aumentar las probabilidades de éxito. Para lograrlo, se utilizarán herramientas de reconocimiento pasivo como CUPP, CRUNCH y CEWL, las cuales permitirán generar contraseñas con información recolectada.

CAPÍTULO IV – EJECUCIÓN DE HERRAMIENTAS

4.1. INTRODUCCIÓN

La ejecución de las herramientas se llevará a cabo en varias fases, comenzando con la recopilación de información pública de los usuarios. Luego, se procederá a la ejecución de las herramientas, comenzando con CUPP, para crear un diccionario inicial. Posteriormente, se utilizará CRUNCH para generar combinaciones adicionales y finalmente, se usará CEWL para extraer palabras específicas de un sitio web. Cada herramienta será analizada y evaluada por su capacidad para generar contraseñas complejas.

4.1.1. CUPP

Cupp es una herramienta de código abierto escrita en el lenguaje de Python, diseñada para crear diccionarios personalizados a partir de información pública disponible. Su principal función es tomar los datos proporcionados por el atacante, como el nombre de la persona, nombres de mascotas, fechas de nacimiento, números de teléfono, entre otros detalles personales, para generar combinaciones de contraseñas que podrían ser utilizadas por la víctima. Esto aumenta considerablemente la probabilidad de éxito en un ataque de diccionario.

Al ser una herramienta altamente personalizable, Cupp permite especificar el tipo de información que se desea recopilar. Muchas personas tienden a colocar algunos patrones cuando se trata de elegir una contraseña. Por lo general, eligen contraseñas que son fáciles de recordar e incluyen cosas personales en sus contraseñas. Por ejemplo, para recordar fácilmente una contraseña, puede contener el nombre o el cumpleaños de alguien. Si el nombre es Andrés, cuya fecha de nacimiento es el 07/09/2000, es posible que el usuario tenga una contraseña similar a esta “Andres07092000”. Estas contraseñas predecibles es lo que

Cupp explota para construir diccionarios efectivos (Cybrary, 2020).

Sin embargo, la efectividad de Cupp depende en gran medida de la precisión de los datos recopilados. Si la información sobre la víctima es escasa, incorrecta o difícil de obtener, el diccionario podría no ser tan eficaz. Esto resalta la importancia de un enfoque bien planificado, al recopilar datos para garantizar un mayor éxito en la generación de contraseñas.

4.1.2. CRUNCH

Crunch es una herramienta de código abierto, diseñada para sistemas Unix y Linux, ampliamente utilizada en pruebas de penetración y auditorías de seguridad. Su principal función es generar listas de palabras personalizadas a partir de parámetros específicos, como la longitud mínima y máxima de las contraseñas y los caracteres a incluir, como letras, números, símbolos, entre otras. Esta herramienta es muy útil cuando se necesita generar una gran cantidad de combinaciones en un tiempo limitado. La ventaja de Crunch es su capacidad para crear diccionarios muy diversos, lo que lo convierte en una opción muy potente y eficaz para realizar un ataque de diccionario (elcursodelhacker, 2022).

Una de las mayores fortalezas de Crunch es su capacidad para crear diccionarios extremadamente variados y adaptados a las necesidades del usuario. Esto la convierte en una herramienta indispensable para profesionales de la seguridad que buscan explorar una amplia gama de posibles contraseñas. Sin embargo, Crunch también presenta algunas limitaciones. A diferencia de herramientas como Cupp, no tiene la capacidad de incorporar información personal de las víctimas, lo que puede reducir su eficacia en escenarios donde esta información es clave para aumentar las probabilidades de éxito. Además, la generación masiva de combinaciones puede ser intensiva en recursos, requiriendo tanto espacio de almacenamiento como tiempo de procesamiento.

4.1.3. CEWL

Cewl es una herramienta open source integrada en Kali Linux, diseñada para la recolección de palabras clave, números, correos electrónicos y otros datos relevantes a partir del contenido de un sitio web, lo que permite crear listas de posibles contraseñas basadas en temas que podrían ser de interés para una víctima. Esto es útil especialmente cuando se tiene acceso a las redes sociales o sitios web de una persona, ya que permite crear diccionarios basados en interés específicos (Reydes, 2014).

Sin embargo, una de sus limitaciones es que se basa únicamente en contenido accesible en línea, por lo que, si la víctima no tiene una presencia significativa en la web o en las redes sociales, el diccionario generado será menos efectivo. Aunque puede ser muy útil para obtener palabras relacionadas con la víctima, el proceso de recolección puede ser más lento en comparación con otras herramientas más directas, como Crunch.

4.2. SISTEMA OPERATIVO KALI LINUX

Para la práctica y ejecución de herramientas, se ha seleccionado el sistema operativo de Kali Linux, ejecutado en una máquina virtual. Como se mencionó en el marco conceptual, Kali Linux es una distribución de código abierto basada en Debian, especialmente diseñada para pruebas de penetración y auditorías de seguridad. Su potencial radica en la amplia gama de herramientas preinstaladas que ofrece, esta distribución es ampliamente reconocida en el ámbito de la ciberseguridad por su versatilidad y capacidad de adaptación a diversos entornos de prueba.

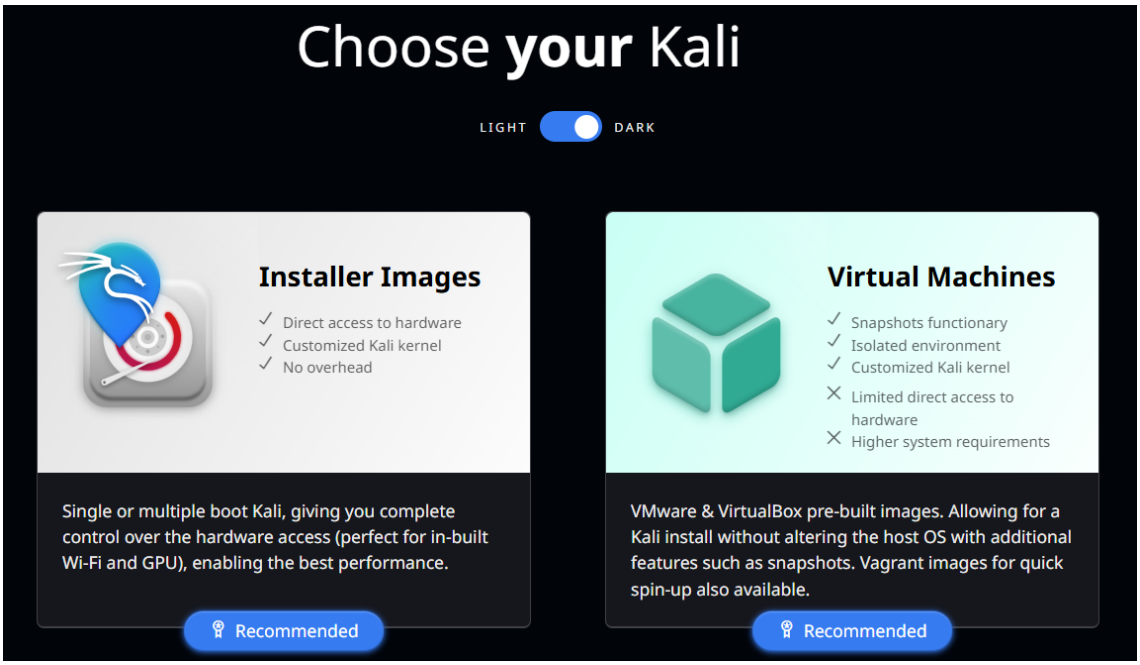
En este proyecto, Kali Linux se convierte en una elección ideal, ya que proporciona un entorno controlado y seguro para la ejecución de herramientas como Cupp, Crunch y Cewl. Su capacidad para instalar y gestionar estas herramientas de manera eficiente, le convierte en un sistema operativo super útil para el desarrollo de esta práctica. Además, al

ejecutarse en un entorno virtual, se garantiza el aislamiento de las actividades realizadas, evitando cualquier impacto no deseado en el sistema operativo principal.

En la elección del Kali Linux, se optó por la opción de “Virtual Machines” debido a las ventajas que ofrece. Estas permiten una configuración más rápida, ya que vienen preinstaladas y preconfiguradas. Esto elimina la necesidad de llevar a cabo un proceso de instalación completa desde cero.

Ilustración 2

Selección del sistema operativo.



Al optar por la opción de Virtual Machines, se seleccionó una imagen preconfigurada en formato OVA, para el hipervisor VirtualBox. Como la imagen se encuentra preconfigurada, el sistema operativo vino por defecto con una memoria base de 2048 MB, 2 CPU y un tamaño de disco virtual de 80 GB. Como se observa en las ilustraciones 3, 4 y 5, estos parámetros no requieren ajustes adicionales, ya que han sido establecidas para brindar un equilibrio entre el rendimiento y la compatibilidad con el sistema anfitrión.

Ilustración 3

Memoria base del sistema.

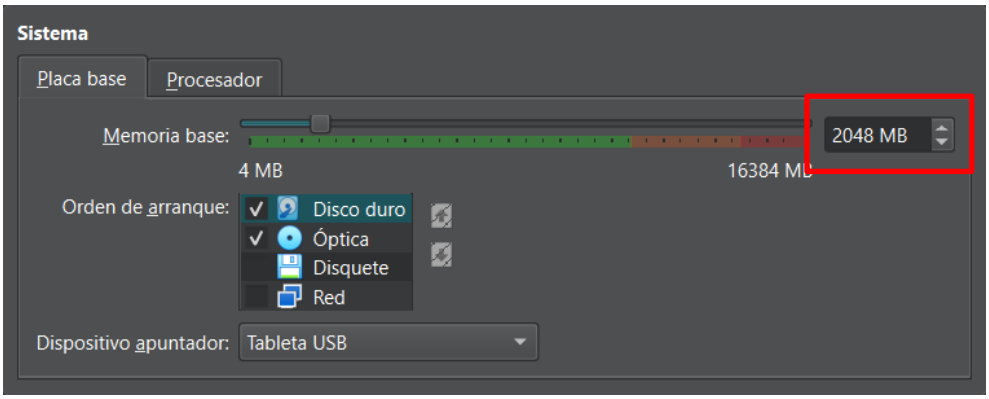


Ilustración 4

Procesadores del sistema.

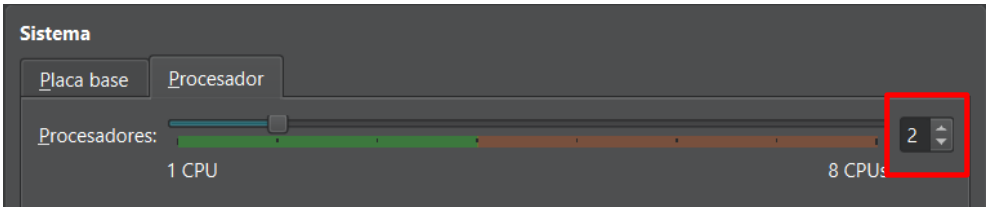
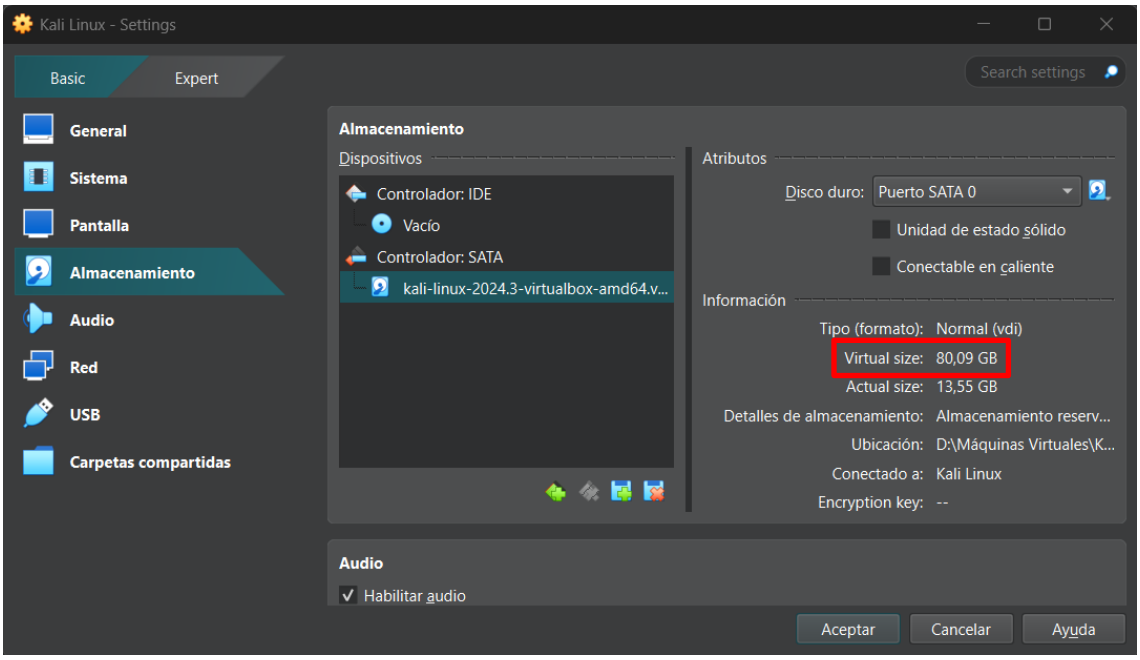


Ilustración 5

Almacenamiento del sistema.



4.3. LEVANTAMIENTO DE INFORMACIÓN

Usuario: Daniel Noboa Azín.

Daniel Noboa Azín es un político, empresario y actual presidente del Ecuador desde 2023. Hijo de Álvaro Noboa, reconocido magnate bananero y político ecuatoriano, Daniel tiene una trayectoria destacada en el ámbito empresarial y político. Antes de asumir la presidencia, trabajó como asambleísta y lideró importantes iniciativas legislativas relacionadas con el desarrollo económico, comercio exterior y tecnología. Nació el 30 de noviembre de 1987 en Guayaquil, Ecuador.

Para la recopilación de información sobre Daniel Noboa, se utilizó el sitio web llamado Wikipedia, el cual proporciona una bibliografía detallada sobre el actual presidente del Ecuador. Este sitio web incluye datos relevantes como sus nombres completos, los nombres de sus padres, su cónyuge y sus hijos. Además, se encuentran bien especificadas las fechas y los lugares de nacimiento tanto de Daniel como de sus familiares, así como otros

aspectos importantes sobre su trayectoria profesional. Como se muestra en las ilustraciones.

Ilustración 6

Información del usuario 1.

Usuario: Kale Anders.

Kale Anders es un empresario y creador de contenido originario de Suecia, promueve cursos para aprender inglés de manera efectiva a través de distintas plataformas como Youtube, Instagram y cursos en línea, evitando métodos convencionales como hablar, leer y escribir. Además, utiliza estrategias de marketing digital para promover sus cursos en línea, lo que lo ha posicionado como un referente conocido en el sector del aprendizaje de idiomas.

Sin embargo, a pesar de que Anders ofrece servicios a través de redes sociales y es conocido como una figura pública, la información sobre su vida personal es limitada. Según el sitio web Famous Birthdays, una plataforma que recopila datos sobre figuras públicas. Se

logró encontrar un poco de información personal, pero el sitio web no dispone de detalles relacionados con su entorno familiar o relaciones. Esto evidencia el contraste en la disponibilidad de información pública entre figuras como políticos y personalidades emergentes del ámbito digital.

Ilustración 7

Información del usuario 2.

Para el último usuario de prueba, se consideró como posible víctima a un perfil creado específicamente para este propósito. Con el fin de evidenciar lo complicado que se resulta la recolección de información cuando los usuarios no son públicos. Como se muestra en la ilustración, se procedió con la creación de la cuenta de Instagram, donde se visualiza la contraseña que se utilizó para el perfil.

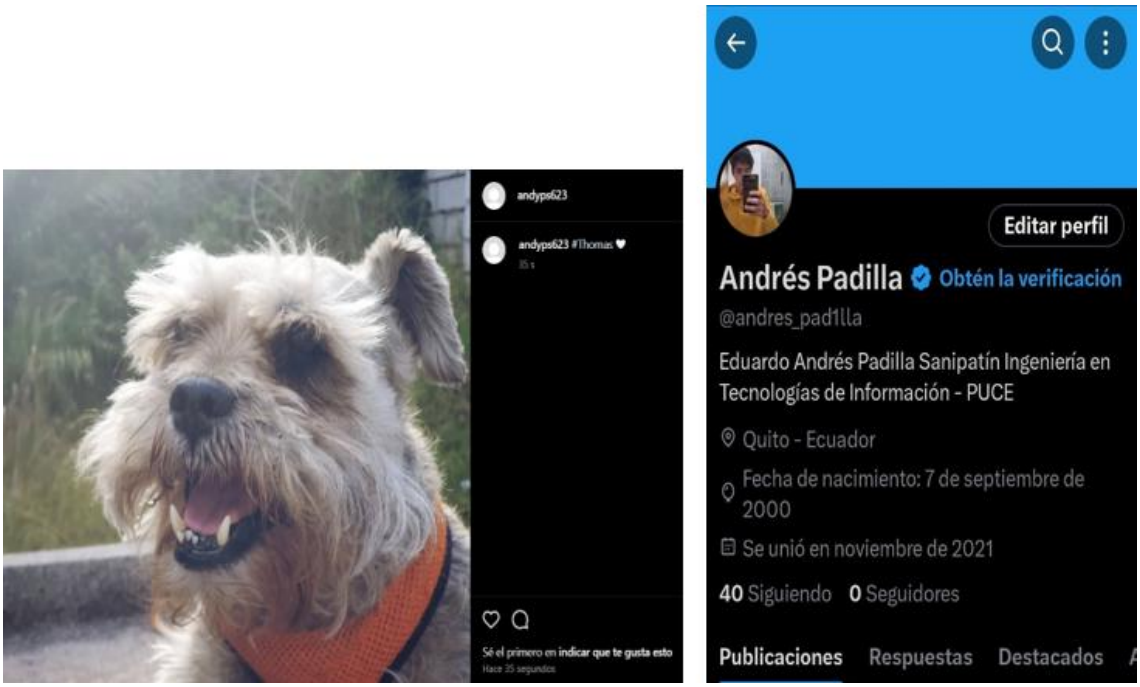
Ilustración 8

Creación del perfil.

Adicionalmente, se recopiló información a partir de una publicación realizada en la cuenta de Instagram creada previamente y mediante un perfil de la red social X.

Ilustración 9

Información del usuario 3.

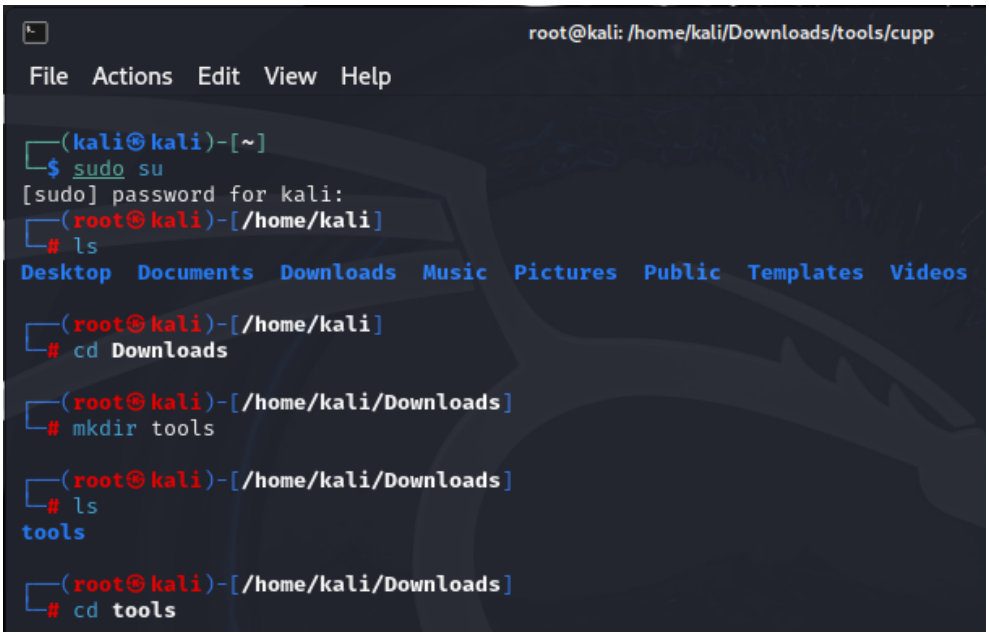


4.4. INSTALACIÓN DE HERRAMIENTA CUPP

Es importante señalar que, a diferencia de las otras herramientas utilizadas en este proyecto, Cupp es la única que requiere instalación, ya que las demás vienen preinstaladas en Kali Linux. Como se muestra en la ilustración 10, el procedimiento se inicia trabajando en modo superusuario mediante el comando **sudo su**, el cual otorga privilegios de root. Estos son los permisos más altos que puede tener un usuario en un sistema operativo. Posteriormente, se utilizó el comando **cd** para ingresar a la carpeta de descargas (*Downloads*) y se empleó el comando **mkdir** para crear una nueva carpeta nombrada *tools.*

Ilustración 10

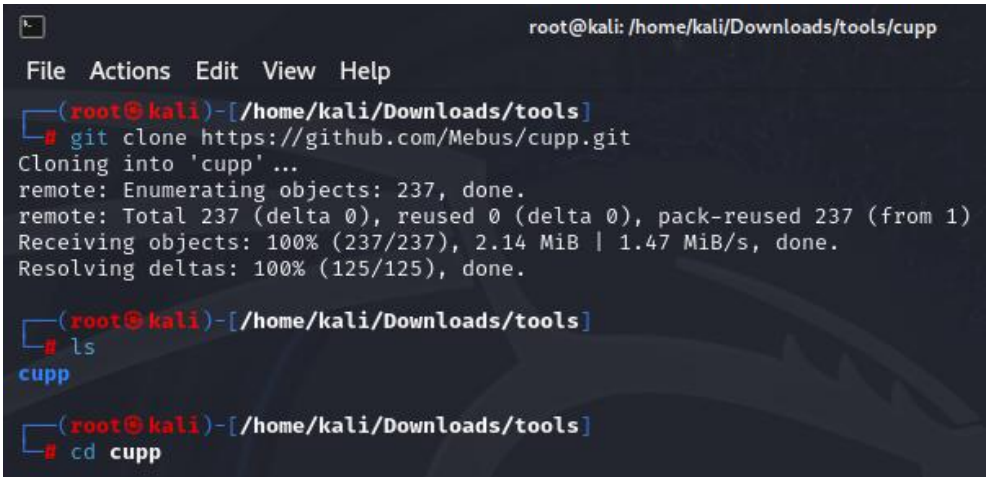
Instalación inicial de Cupp.



Se creó la carpeta *tools* con el objetivo de descargar la herramienta CUPP en esta carpeta. Una vez dentro de la carpeta, se ejecutó el comando **git clone**, el cual permite descargar el repositorio de CUPP desde GitHub. Finalmente, se verificó la correcta descarga utilizando el comando **ls**, que muestra la lista de archivos contenidos en el directorio actual.

Ilustración 11

Instalación de la herramienta Cupp.



Al ejecutar el comando **python cupp.py**, se inicia la ejecución del script *cupp.py*. La

herramienta CUPP presenta diversas opciones de uso, entre las cuales tenemos:

Tabla 2

Opciones de la herramienta Cupp.

-h, --help	Muestra un mensaje de ayuda y la descripción de todas las opciones disponibles.
-i, --interactive	Activa el modo interactivo, donde la herramienta hace preguntas para recopilar información de la víctima, con el objetivo de generar el diccionario personalizado.
-w FILENAME	Utiliza un archivo existente llamado FILENAME, para mejorar el diccionario de contraseñas.
-l	Descarga grandes listas de palabras directamente desde un repositorio.
-a	Extrae nombres de usuario y contraseñas predeterminadas de una base de datos específica.
-v, --version	Muestra la versión actual de la herramienta.
-q, --quiet	Ejecuta la herramienta en modo silencioso, es decir, no muestra el banner inicial. Es útil para mantener la consola limpia.

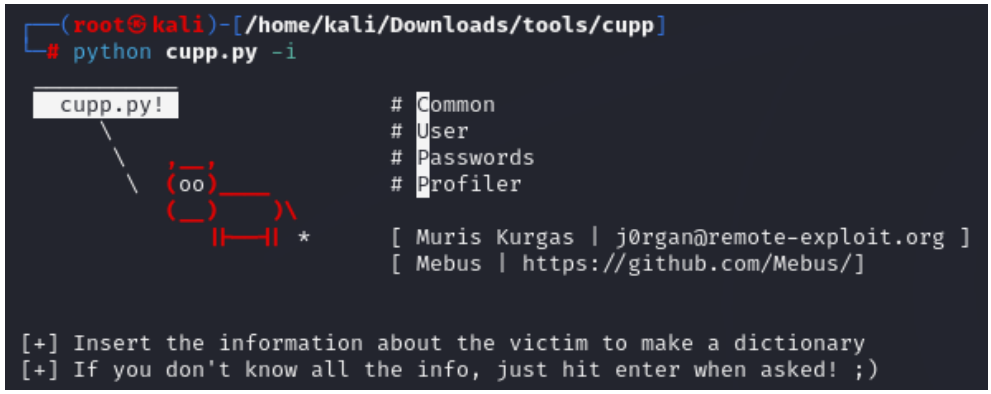
4.5. EJECUCIÓN DE CUPP

Como se evidencia en la ilustración 12, se utilizó la opción *-i* con el comando **Python cupp.py -i**, que, como se detalló en la tabla 2, permite generar un diccionario personalizado a partir de información específica. Esta opción solicita una serie de datos personales, como nombres, fechas de nacimiento, apodos, entre otros detalles que podrían ser utilizados como

posibles contraseñas.

Ilustración 12

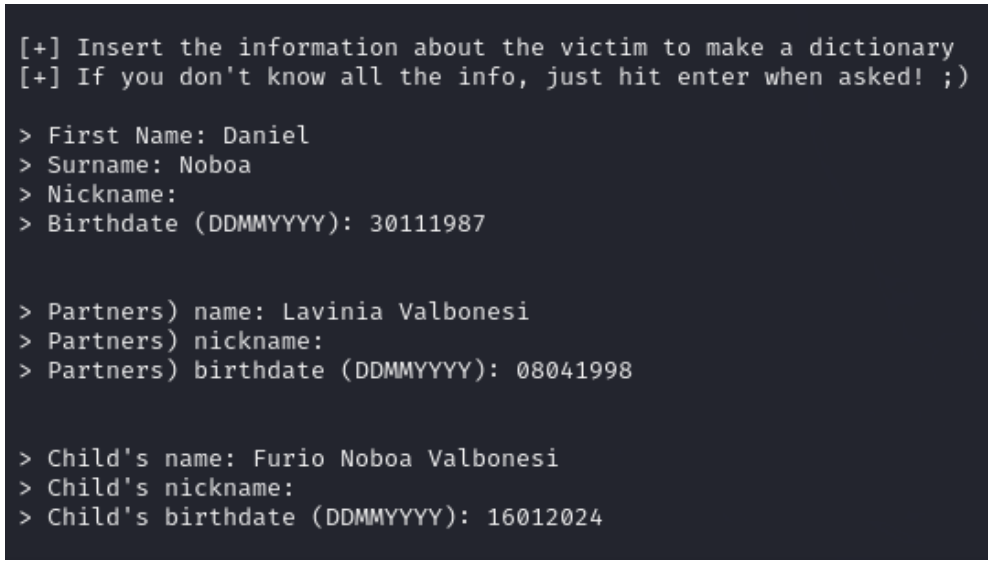
Modo interactivo de Cupp.



Para comenzar, se iniciaron las pruebas con el usuario número 1, Daniel Noboa. Como mencionamos con anterioridad, a través del sitio web Wikipedia, se obtuvo información como los nombres completos y las fechas de nacimiento del usuario, su esposa y su hijo, además del nombre de la compañía con la que está relacionado.

Ilustración 13

Ingreso de datos personales del usuario 1.

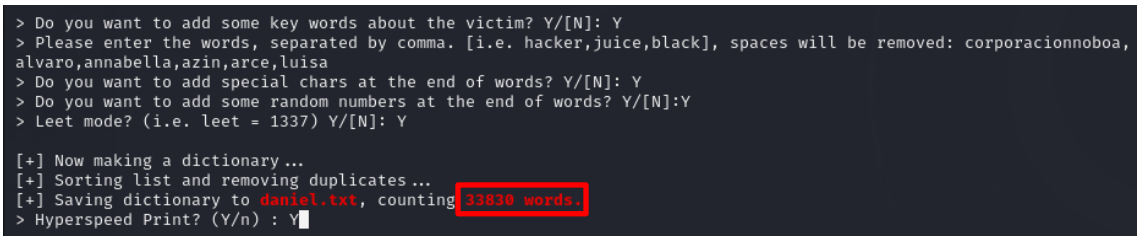


Tras completar las preguntas preestablecidas por la herramienta, esta solicita al

atacante si se desea ingresar palabras clave adicionales relacionadas con la víctima. Además, ofrece la opción de incluir caracteres especiales y números aleatorios al final de las palabras, con el propósito de aumentar las probabilidades de éxito.

Ilustración 14

Opciones de personalización para el usuario 1.



Como se muestra en la ilustración 14, la herramienta Cupp indica que las contraseñas generadas se guardarán en un archivo de texto llamado daniel.txt, con un total de 33830 contraseñas generadas, Una vez finalizada la creación del diccionario, se utilizó el comando **wc -l daniel.txt**, el cual se encarga de contar el número de líneas en el archivo, lo que permite verificar la cantidad total de contraseñas generadas.

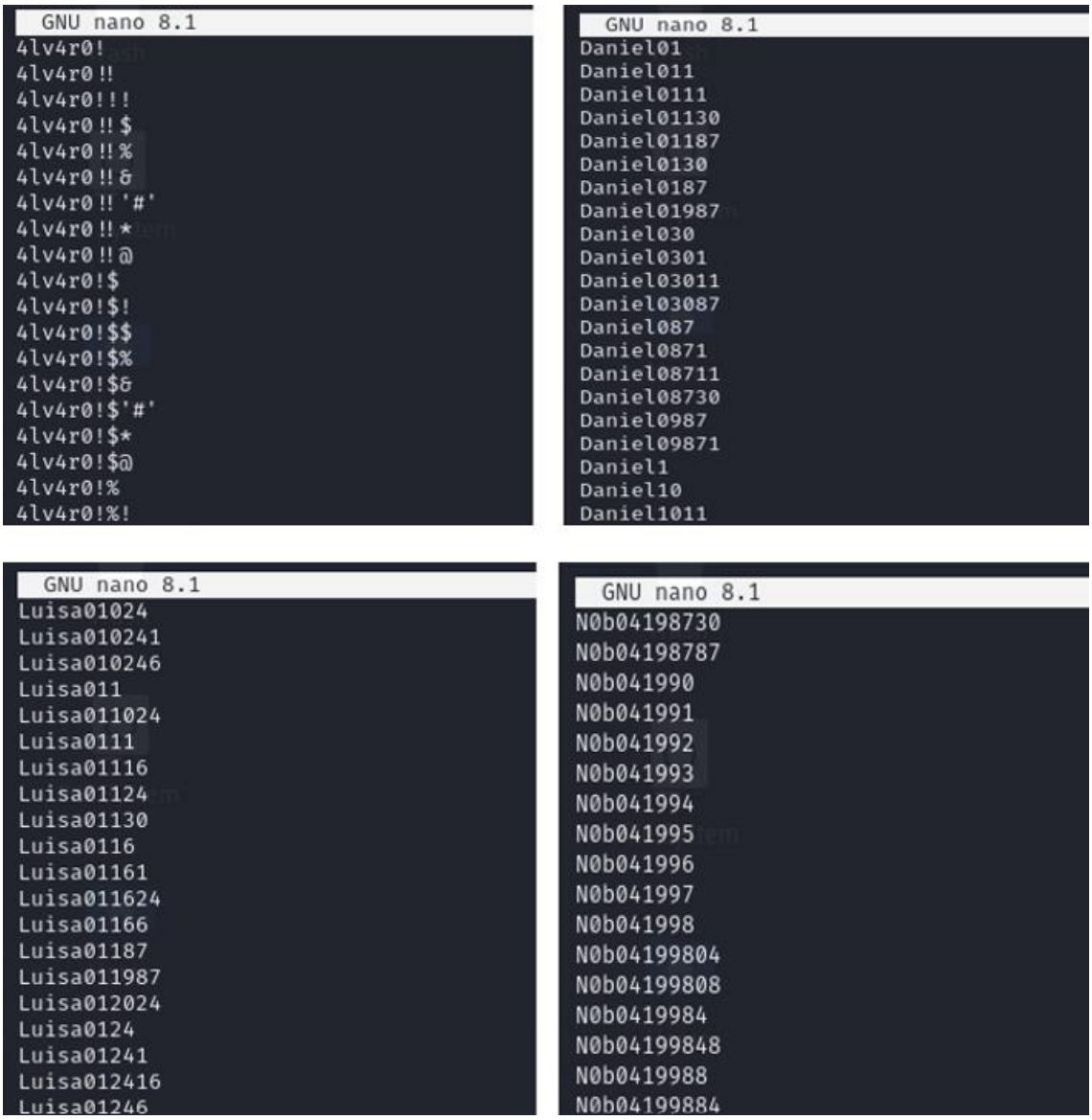
Ilustración 15

Cantidad total de contraseñas generadas para el usuario 1.

Una vez generado el diccionario y guardado en el archivo daniel.txt, se utilizó el comando **nano daniel.txt**, para acceder al archivo y visualizar las contraseñas generadas, tal como se muestra en la ilustración 16.

Ilustración 16

Contraseñas generadas para el usuario 1.



De manera similar al proceso realizado con el primer usuario, se continuó con las pruebas para el usuario 2, Kale Anders. En este caso, se logró recolectar información a través del sitio web Famous Birthdays, el cual proporcionó datos personales del usuario.

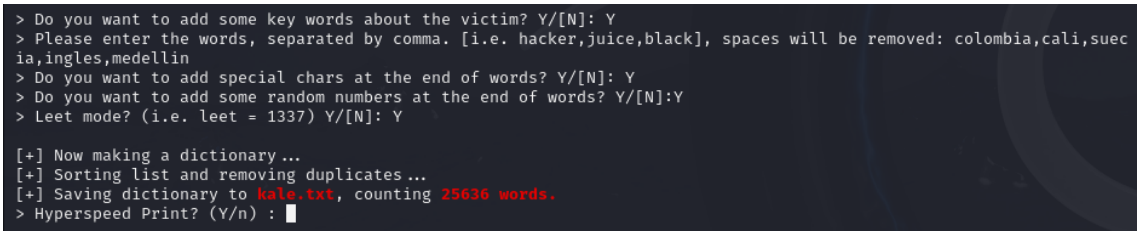
Siguiendo el mismo flujo de trabajo, se ingresaron los datos solicitados por la herramienta para realizar el diccionario personalizado. Además, como se mencionó anteriormente, se respondieron las preguntas adicionales que la herramienta presenta.

Ilustración 17

Ingreso de datos personales del usuario 2.

Ilustración 18

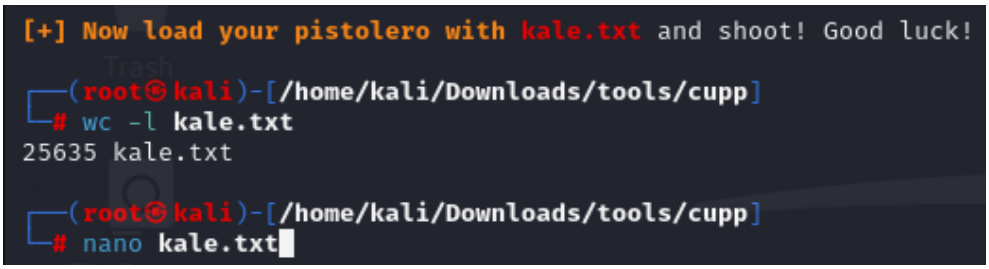
Opciones de personalización para el usuario 2.



Como parte del proceso de validación, se volvió a ejecutar el comando **wc -l kale.txt** para garantizar el número total de contraseñas generadas en el archivo correspondiente al usuario Kale Anders.

Ilustración 19

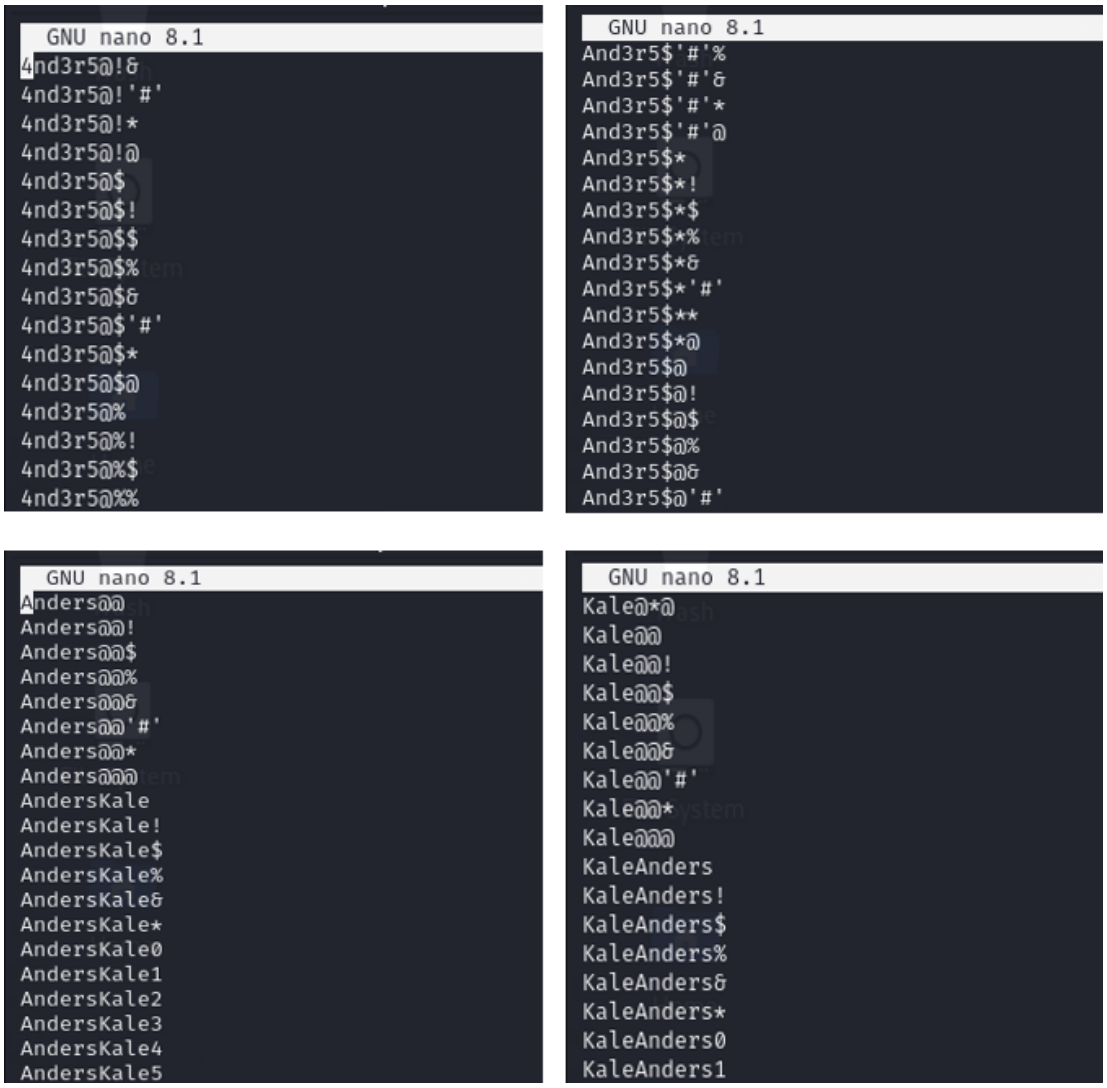
Cantidad total de contraseñas generadas para el usuario 2.



Se utilizó el comando **nano kale.txt** para acceder al archivo que contiene las contraseñas generadas para el usuario Kale Anders.

Ilustración 20

Contraseñas generadas para el usuario 2.



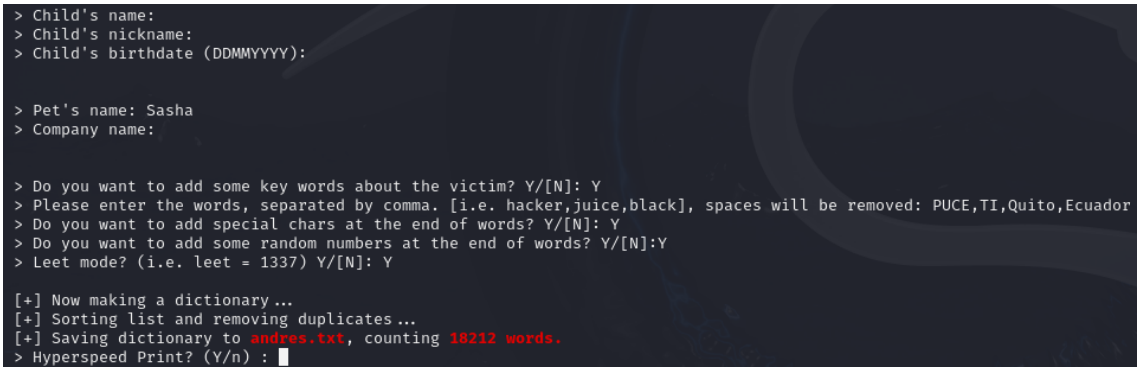
De manera reiterativa, se llevó a cabo el mismo procedimiento para el tercer usuario, el cual fue creado con el propósito de validar la efectividad de los diccionarios generados mediante un código en Python, que será ejecutado en etapas posteriores del proyecto.

Ilustración 21

Ingreso de datos personales del usuario 3.

Ilustración 22

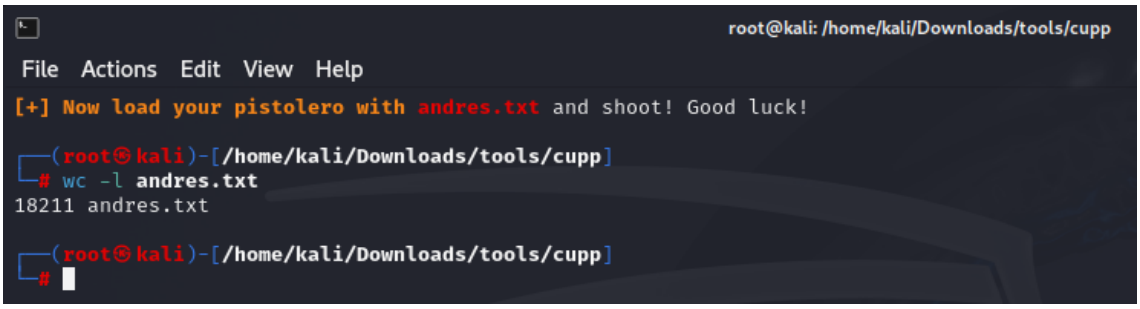
Opciones de personalización para el usuario 3.



Validación de la cantidad total de líneas del archivo *andres.txt.* utilizando el comando **wc -l andres.txt**.

Ilustración 23

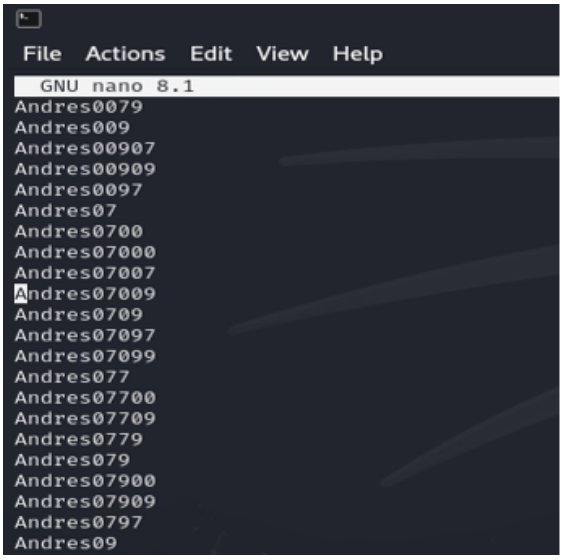
Cantidad total de contraseñas generadas para el usuario 3.

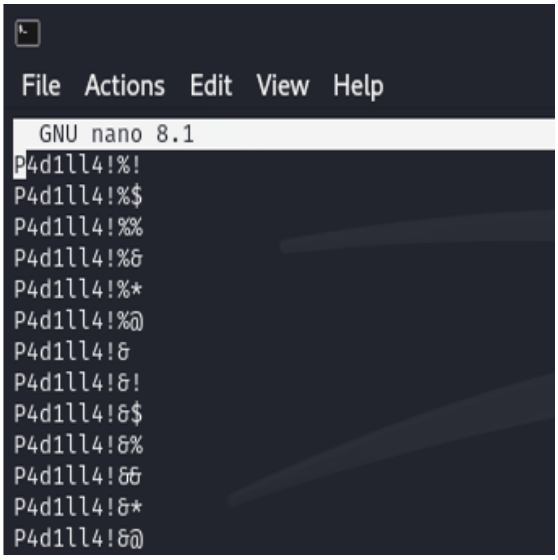


Para revisar el contenido del diccionario generado para el tercer usuario, se empleó nuevamente el comando **nano** para acceder al archivo.

Ilustración 24

Contraseñas generadas para el usuario 3.



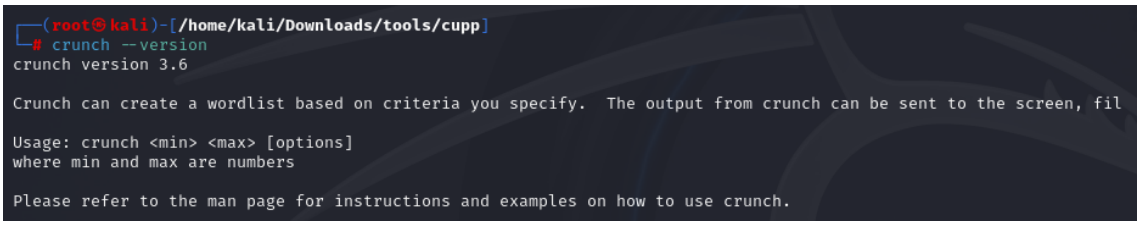


4.6. EJECUCIÓN DE CRUNCH

La herramienta Crunch no requirió instalación, ya que se encuentra preinstalada en el sistema operativo. Para confirmar su disponibilidad, se utilizó el comando **crunch –version**, el cual permitió verificar la versión instalada.

Ilustración 25

Versión de la herramienta Crunch.



Al ejecutar el comando **man crunch**, se obtiene la documentación de la herramienta Crunch, donde se detallan las diversas opciones de uso. Entre las más relevantes tenemos:

Tabla 3

Opciones de la herramienta Crunch.

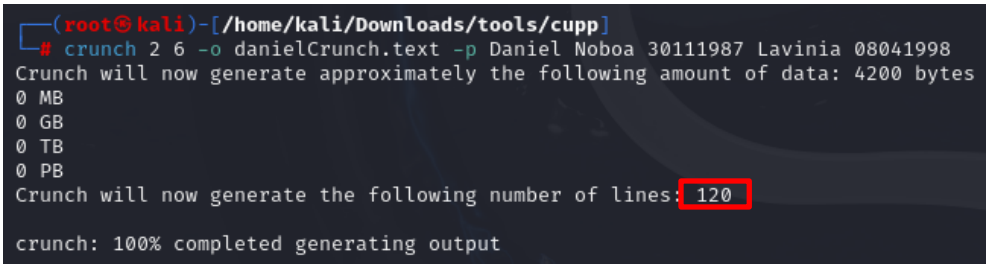
-h, --help	Muestra un mensaje de ayuda con la descripción de todas las opciones disponibles en Crunch.
-b	Establece el tamaño máximo del archivo de salida. Es útil cuando se desea limitar el tamaño del archivo generado para no sobrecargar los recursos del sistema.
-c	Limita la cantidad de combinaciones a generar según la especificación del atacante.
-t	Define un patrón de caracteres a seguir para la generación de contraseñas.
-o	Permite especificar el nombre del archivo donde se guardará el diccionario generado.
-p	Define un patrón específico para las contraseñas generadas.

Para la generación del diccionario, se utilizó el comando que se muestra en la ilustración 26, donde **crunch 2 6**, indica que las contraseñas generadas tendrán una longitud mínima de 2 caracteres y una longitud máxima de 6 caracteres. La opción **-o** especifica que el archivo de salida se llamará *danielCrunch.text*, en el cual se guardarán las contraseñas

generadas. Finalmente, el comando **-p** define un patrón específico para las contraseñas, basado en los datos proporcionados.

Ilustración 26

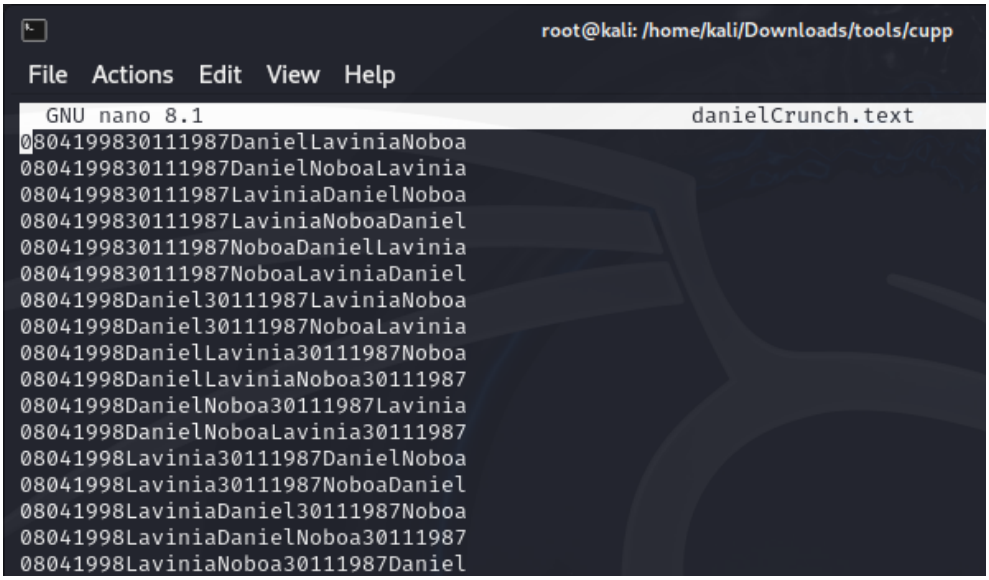
Ejecución de la herramienta Crunch para el usuario 1.



Una vez concluida la ejecución de la herramienta, podremos observar el número de combinaciones generadas, como se muestra en la ilustración 26. Para acceder al archivo y visualizar las contraseñas generadas se utilizó el comando **nano**.

Ilustración 27

Contraseñas generadas por la herramienta Crunch para el usuario 1.



De igual manera, se utilizarán los mismos comandos para los usuarios 2 y 3, tanto en la ejecución como en la visualización, como se muestran en las ilustraciones 28, 29, 30 y 31.

Ilustración 28

Ejecución de la herramienta Crunch para el usuario 2.

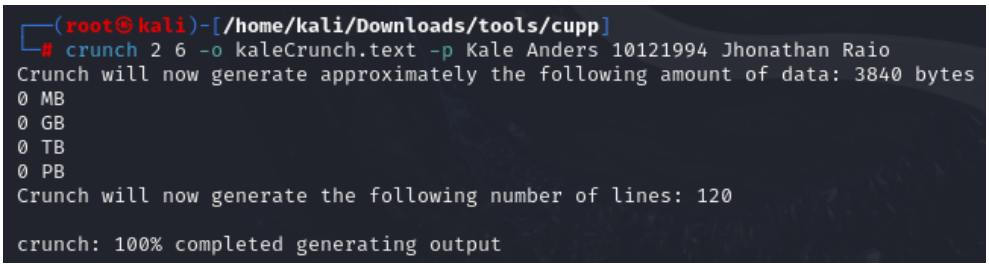


Ilustración 29

Contraseñas generadas por la herramienta Crunch para el usuario 2.

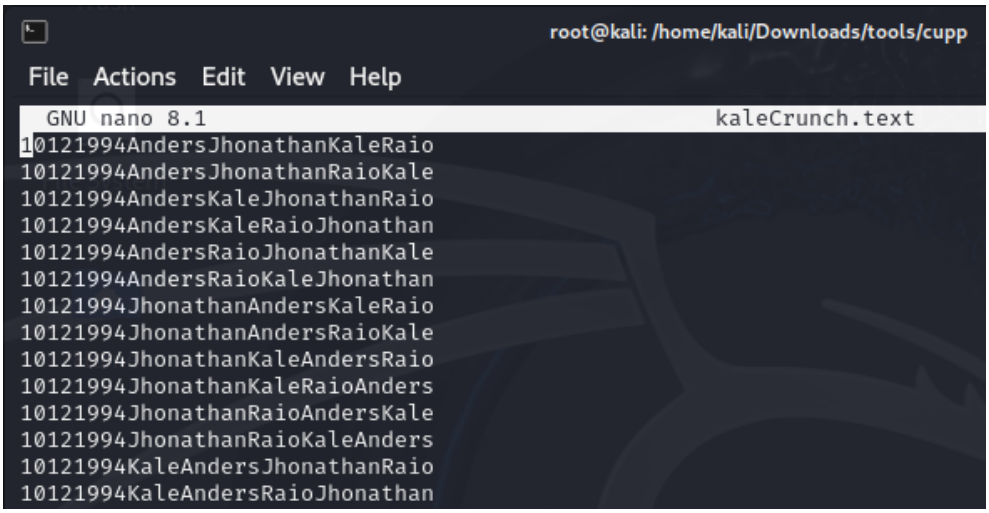


Ilustración 30

Ejecución de la herramienta Crunch para el usuario 3.

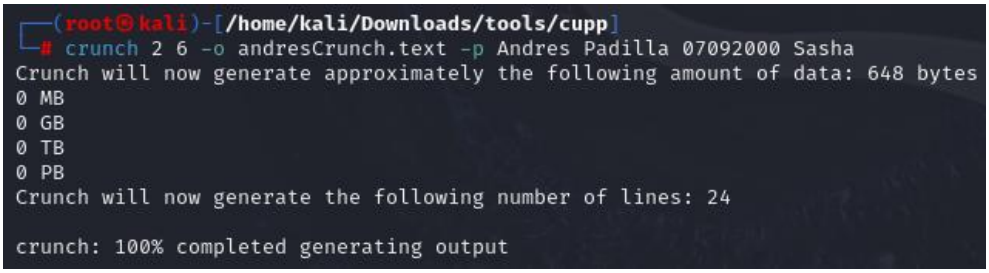
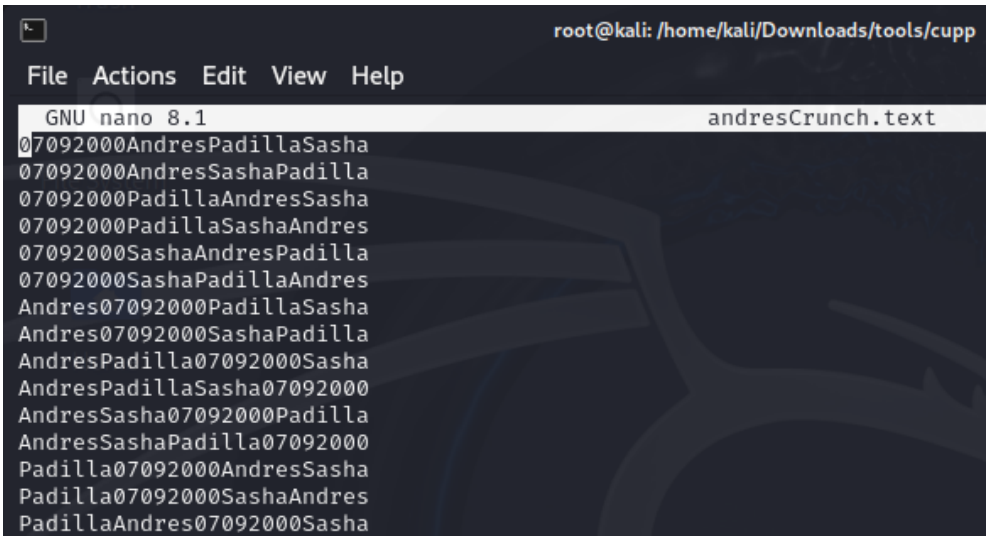


Ilustración 31

Contraseñas generadas por la herramienta Crunch para el usuario 3.



4.7. EJECUCIÓN DE CEWL

Para el primer caso se recurrió al sitio web de Daniel Noboa, donde se encuentra información relevante sobre él, como su educación, profesión y experiencia laboral.

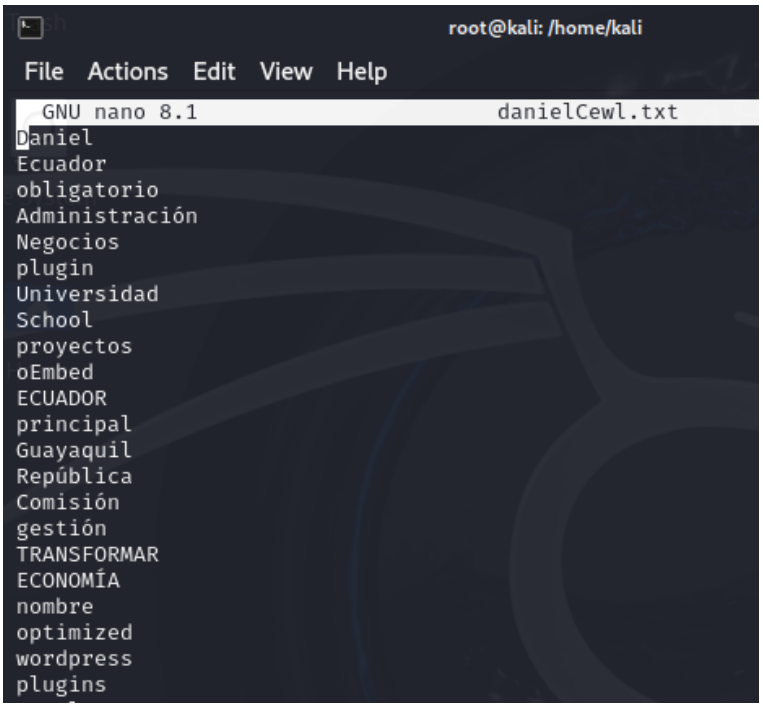
Para la ejecución de la herramienta se utilizó el comando que se muestra en la ilustración 32, donde el parámetro **-d 2** indica que Cewl debe recorrer la página web a una profundidad de 2 enlaces, es decir, seguirá los enlaces dentro del sitio web hasta el segundo nivel. El parámetro **-m 6** establece que solo se capturen palabras con una longitud mínima de 6 caracteres. Finalmente, el parámetro **-w** especifica que las palabras extraídas se guardarán en un archivo llamado *danielCewl.txt*.

Ilustración 32

Ejecución de la herramienta Cewl para el usuario 1.

Ilustración 33

Recopilación de palabras generadas por la herramienta Cewl para el usuario 1.



De manera similar, se utilizó el mismo comando para los usuarios 2 y 3, realizando adaptaciones en el nombre del archivo destinado al almacenamiento de la lista de palabras y en la dirección del sitio web correspondiente a cada caso.

Ilustración 34

Ejecución de la herramienta Cewl para el usuario 2.

Ilustración 35

Recopilación de palabras generadas por la herramienta Cewl para el usuario 2.

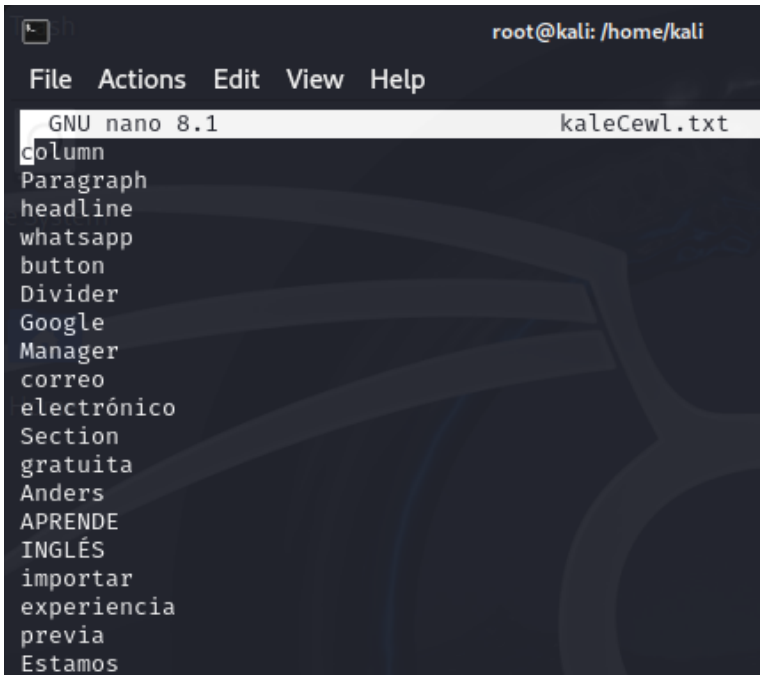
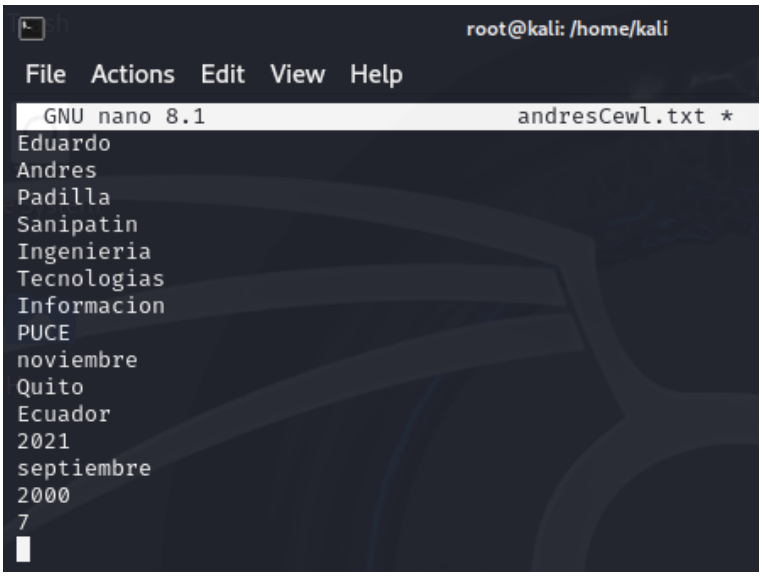


Ilustración 36

Ejecución de la herramienta Cewl para el usuario 3.

Ilustración 37

Recopilación de palabras generadas por la herramienta Cewl para el usuario 3.



Se conoce que la principal fortaleza de la herramienta Cewl radica en su capacidad para extraer y generar listas de palabras directamente del contenido textual de sitios web, lo que la convierte en una herramienta altamente efectiva para la creación de diccionarios personalizados. Con el objetivo de ampliar las probabilidades de éxito, se desarrolló un script en Python para complementar los resultados generados por Cewl. La principal función de este script es agregar algunos caracteres especiales y combinar las palabras extraídas por la herramienta. El proceso inició con la creación del archivo *.py*, en el cual se desarrolló el script.

Ilustración 38

Archivo Python.

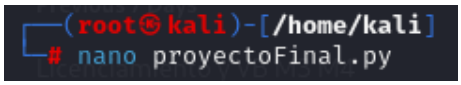


Ilustración 39

Script en Python para complemento de resultados.

```
  GNU nano 8.1                    proyectoFinal.py
import random
import itertools

with open("andresCewl.txt", "r") as file:
    words = [line.strip() for line in file]

special_chars = ["!", "@", "#", "$", "%", "&", "*", "_", "-"]

combinations = []
for r in range(2, len(words) + 1):
    combinations += list(itertools.combinations(words, r))
    if len(combinations) > 1000:
        break

final_combinations = []
for combo in combinations:
    char = random.choice(special_chars)
    combined = char.join(combo)
    final_combinations.append(combined)

with open("combinaciones_generadas.txt", "w") as output_file:
    for combination in final_combinations:
        output_file.write(combination + "\n")
print(f"Se generaron {len(final_combinations)} combinaciones y se guardaron en 'combinaciones_generadas.txt'.")
```

Para facilitar la comprensión del script desarrollado, se proporciona una explicación detallada de cada una de sus líneas.

1. import random
    - Se importa el módulo random para trabajar con selección aleatoria de elementos.
2. import itertools
    - Se importa el módulo intertools, que incluye herramientas para crear combinaciones.
3. with open("andresCewl.txt", "r") as file:
    - Abre el archivo andresCewl.txt en modo lectura r.
4. words = [line.strip() for line in file]
    - Lee cada línea del archivo, elimina los saltos de línea al inicio y final (strip()) y guarda cada palabra en una lista llamada words.
5. special_chars = ["!", "@", "#", "$", "%", "&", "*", "_", "-"]
    - Lista de caracteres especiales que se utilizarán para unir las

combinaciones de palabras.

6. combinations = []
    - Se inicializa una lista vacía para almacenar las combinaciones de palabras.
7. for r in range(2, len(words) + 1):

   combinations += list(itertools.combinations(words, r))
    - El bucle recorre diferentes tamaños r desde 2 hasta el total de palabras en la lista words y genera todas las combinaciones posibles de ese tamaño usando itertools.combinations. Luego, convierte estas combinaciones en una lista y las añade a la variable combinations.

   if len(combinations) > 1000:
    - Verifica si la lista combinations contiene más de 1000 elementos.

   break
    - Si se alcanzan más de 1000 combinaciones, el bucle se detiene para evitar generar demasiados datos.
8. final_combinations = []
    - Se inicializa una lista vacía para almacenar las combinaciones finales, donde las palabras estarán unidas por un carácter especial.
9. for combo in combinations:
    - Itera sobre cada combinación de palabras.

   char = random.choice(special_chars)
    - Selecciona un carácter especial aleatorio de la lista special_chars.

   combined = char.join(combo)
    - Une las palabras en la combinación actual (combo) usando el carácter

especial seleccionado.

final_combinations.append(combined)

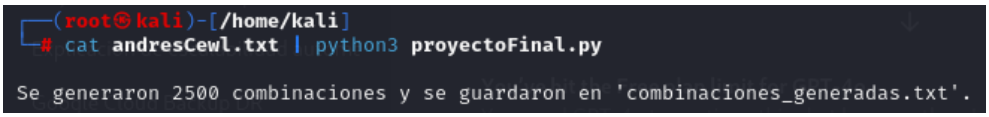
- Añade la combinación final como una cadena a la lista final_combinations.

10. with open("combinaciones_generadas.txt", "w") as output_file:
    - Crea un archivo llamado combinaciones_generadas.txt en modo escritura w.
11. for combination in final_combinations:
    - Itera sobre cada combinación en final_combinations.
12. output_file.write(combination + "\n")
    - Escribe sobre cada combinación en una línea del archivo.
13. print(f"Se generaron {len(final_combinations)} combinaciones y se guardaron en 'combinaciones_generadas.txt'.")
    - len(final_combinations): Calcula el número total de combinaciones generadas.
    - print(): Muestra un mensaje indicando cuantas combinaciones se generaron y donde se guardaron.

Se ejecutó el script de Python con el comando que se presenta en la ilustración 40, el cual pasó el contenido del archivo *andresCewl.txt* como entrada al script *proyectoFinal.py* para que fuera procesado por el programa.

Ilustración 40

Ejecución del script.



Una vez concluida la ejecución del script, podremos observar el número de combinaciones generadas, como se muestra en la ilustración 40. Para acceder al archivo y visualizar las contraseñas generadas se utilizó el comando **nano**.

Ilustración 41

Diccionario regenerado a partir del código Python.

4.8. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

El análisis de los resultados obtenidos mediante las herramientas de reconocimiento pasivo reveló diferencias significativas en la forma en que cada una generó los diccionarios de contraseñas, mostrando tanto ventajas como limitaciones en el proceso.

En cuanto a Cupp, la experiencia personal fue bastante positiva, ya que fue la única herramienta que requirió instalación, pero el proceso fue muy rápido y sencillo. La herramienta proporciona una experiencia interactiva al solicitar información relevante sobre la víctima, con preguntas que podrían ser fácilmente accesibles. Además, las opciones adicionales para agregar palabras clave, caracteres especiales y números al final de las

palabras, así como la combinación de las distintas respuestas proporcionadas, resultaron ser muy útiles para crear el diccionario. Aunque el proceso de creación del diccionario fue lento debido a la gran cantidad de contraseñas generadas (33.830 para el usuario 1, 25.636 para el usuario 2 y 18.212 para el usuario 3), esto se justifica por el alto volumen de combinaciones que la herramienta produce. Es importante destacar que, si no logra recopilar mucha información sobre la víctima, no se puede sacar el máximo provecho a la herramienta.

En contraste, Crunch ofrece una amplia variedad de opciones, pero su documentación no especifica de manera clara el funcionamiento de cada parámetro. Esto puede resultar complicado para quienes se inician en el campo del hacking ético. En cuanto a la generación de diccionarios, esta herramienta se destacó por su rapidez, ya que es capaz de generar una gran cantidad de combinaciones en un corto período de tiempo. Personalmente, me pareció una herramienta bastante útil como generador de contraseñas. Sin embargo, un aspecto negativo de esta herramienta es que, al querer generar un diccionario personalizado, es recomendable utilizar el comando -p para especificar patrones. Al ingresar los distintos patrones relacionada con la víctima, Crunch los toma a todos y los junta. Por ejemplo, si el atacante ingresa 10 patrones que podrían estar relacionados con la víctima, la herramienta unirá aleatoriamente todos los patrones, lo que da como resultado contraseñas excesivamente largas. Por lo que, siendo realistas, para un usuario común este tipo de contraseñas no serían fáciles de recordar ni de usar.

Por último, en cuanto a Cewl, su ejecución resultó sencilla y no requirió instalación, lo cual representa una ventaja. Sin embargo, para aprovechar al máximo esta herramienta, es fundamental emplear más opciones de manera estratégica. Los “diccionarios” generados por Cewl no se pueden considerar diccionarios tradicionales, sino más bien listas de palabras claves extraídas del contenido de sitios web. Para mejorar los resultados y aumentar la

probabilidad de éxito, se desarrolló un script en Python que permitió combinar las palabras extraídas y añadir caracteres especiales, lo que dio como resultado un diccionario más completo y efectivo.

4.9. VALIDACIÓN DE LA OBTENCIÓN DE LA CONTRASEÑA

Aunque el objetivo principal de este estudio no era la validación de las herramientas, se decidió realizar una prueba adicional utilizando al usuario 3, creado específicamente para verificar la efectividad de las contraseñas generadas por las herramientas. En esta prueba, se implementó un código en Python que comparó la contraseña del perfil del usuario ficticio con las contraseñas generadas por la herramienta Cupp.

El código, que se muestra en la ilustración 42, abre el archivo generado por Cupp, que contiene un listado de contraseñas y lo compara con la contraseña predefinida del usuario 3 que esta almacenada en una variable. Si alguna de las contraseñas en el diccionario coincide con la contraseña real, el script lo reporta como un hallazgo exitoso. Esto simula un ataque de diccionario, en el que se intenta acceder a una cuenta utilizando contraseñas comunes o previsibles.

Ilustración 42

Script para validar la obtención de la contraseña.

```
  GNU nano 8.1
real_password = "Andres07"

diccionario_path = "andres.txt"

try:
    with open(diccionario_path, "r") as diccionario:
        for linea in diccionario:
            contrasena_prueba = linea.strip()

            if contrasena_prueba == real_password:
                print(f"¡Contraseña hackeada! La contraseña es: {contrasena_prueba}")
                break
        else:
            print("La contraseña no se encuentra en el diccionario.")
except FileNotFoundError:
    print(f"No se encontró el archivo: {diccionario_path}")
except Exception as e:
    print(f"Ocurrió un error: {str(e)}")
```

Ilustración 43

Ejecución del script.

```
┌──(root㉿kali)-[/home/kali/Downloads/tools/cupp]
└─# python3 pruebaHack.py
¡Contraseña hackeada! La contraseña es: Andres07
```

En este ejemplo, el archivo *andres.txt* contiene el diccionario de contraseñas generado por la herramienta Cupp. El script valida si la contraseña real del usuario 3 "Andres07" se encuentra dentro del archivo generado. Al ejecutarse, el código imprime el mensaje "¡Contraseña hackeada! La contraseña es: Andres07", demostrando que el ataque fue exitoso.

Es importante señalar que, debido a que muchas plataformas y redes sociales modernas implementan mecanismos para limitar los intentos fallidos de inicio de sesión, la validación de esta prueba se realizó en un entorno controlado.

4.10. ESTRATEGIAS DE CONCIENTIZACIÓN Y BUENAS PRÁCTICAS

La implementación de estrategias para prevenir ataques de diccionario es fundamental para mejorar la seguridad de las cuentas y minimizar el riesgo de accesos no autorizados. En este apartado, se detallan cuatro buenas prácticas para fortalecer la seguridad

de las cuentas y promover una cultura de protección ante amenazas cibernéticas:

1. Fomentar el uso de autenticación multifactor (MFA):

   Es fundamental concientizar a los usuarios sobre la importancia de habilitar la autenticación de dos o más factores en todas sus cuentas. Este método combina algo que el usuario sabe (contraseña) con algo que posee (un dispositivo móvil o una aplicación de autenticación). De esta manera, incluso si una contraseña es comprometida, el acceso será imposible sin los factores adicionales.

2. Promover contraseñas seguras y únicas:

   Se debe educar a los usuarios para que creen contraseñas aleatorias y complejas que no incluyan información personal, como nombres, fechas o datos fácilmente accesibles. Además, es recomendable el uso de gestores de contraseñas para generar y almacenar contraseñas únicas de manera segura, evitando la repetición entre cuentas.

3. Fomentar el uso de frases de contraseñas:

   En lugar de simples palabras o combinaciones de caracteres, se debe alentar a los usuarios a utilizar frases completas como contraseñas. Estas frases pueden incluir caracteres especiales, números y letras mayúsculas, convirtiéndolas en opciones más seguras y fáciles de recordar. Por ejemplo, una frase como "¡Ing3ni3r1A T3cn0LoG1c@!" es más difícil de descifrar que una contraseña corta.

4. Actualizar contraseñas de forma periódica:

   Cambiar las contraseñas regularmente es una medida preventiva clave para minimizar el riesgo de que las credenciales comprometidas sean utilizadas por atacantes. Es recomendable que los usuarios actualicen sus contraseñas al menos cada 3 o 6 meses, especialmente en cuentas críticas como correos electrónicos, plataformas bancarias o

sistemas empresariales.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

1. Luego de la ejecución de las herramientas seleccionadas y del análisis de los resultados obtenidos, se comprobó que, a pesar de que Cupp no es una herramienta preinstalada en Kali Linux, es una de las más completas, debido a su capacidad para personalizar diccionarios de manera detallada, basándose en información que puede ser fácilmente accesible.
2. El enfoque de la herramienta Cupp en la creación de diccionarios precisos, resulta altamente efectivo para generar contraseñas complejas y variadas, lo que la convierte en una herramienta valiosa en el proceso de generación de diccionarios para pruebas de seguridad, aunque requiera un tiempo de procesamiento considerable.
3. Las herramientas Crunch y Cewl ofrecen utilidades complementarias, ya que poseen enfoques distintos en la generación de diccionarios. Crunch es ideal para crear grandes listas de combinaciones con parámetros específicos, como la longitud y el uso de caracteres especiales, mientras que Cewl permite personalizar el diccionario extrayendo palabras directamente de sitios web, lo que es valioso para ataques personalizados.
4. De los hallazgos encontrados en el análisis de resultados de la ejecución de las herramientas, resaltan la importancia de ser cuidadosos con lo que se comparte en internet y de usar contraseñas fuertes y únicas.
5. La implementación de un código en Python para probar las contraseñas generadas permitió validar la efectividad de los diccionarios producidos. Este proceso reveló que en situaciones donde no existen limitaciones de intentos de inicio de sesión, las

herramientas utilizadas pueden comprometer contraseñas débiles de manera eficiente.

6. Para minimizar el riesgo de ser víctima de un ataque de diccionario, los usuarios deben adoptar prácticas adicionales como la utilización de contraseñas únicas para cada cuenta y habilitar métodos de autenticación adicionales, como la autenticación de dos factores. Estas medidas no solo mejoran la seguridad, sino que también ayudan a reducir la eficacia de los ataques automatizados y protegen las cuentas contra accesos no autorizados.
7. La investigación evidencia que la educación y prevención sobre la creación de contraseñas seguras son fundamentales para reducir el riesgo de ataques de diccionario. Los usuarios deben comprender la importancia de utilizar contraseñas complejas y únicas, combinando letras, números y caracteres especiales y evitar el uso de información personal fácilmente accesible.

5.1. RECOMENDACIONES

1. Se recomienda diseñar scripts personalizados en Python o en otros lenguajes para validar la eficacia de los diccionarios generados por las herramientas. Estos scripts deben implementarse en entornos controlados y seguros, para evitar cualquier riesgo.
2. Es importante documentar los resultados obtenidos, con el fin de realizar análisis posteriores que sirvan para mitigar los riesgos.
3. Se recomienda a los administradores de res implementar medidas de seguridad robustas para proteger las cuentas de los usuarios. Esto incluye la integración de mecanismo como la limitación de intentos de inicio de sesión, autenticación multifactor y monitoreo de accesos.
4. Aunque no todos los usuarios pueden controlar directamente el número de intentos de inicio de sesión, la recomendación es que las plataformas ofrezcan, y los usuarios

habiliten, autenticación multifactor como medida adicional para proteger las cuentas frente a ataques de diccionario y otros intentos de acceso no autorizado.

5. Los usuarios deben ser más cautelosos con la información que comparten en redes sociales y sitios web públicos. Limitar la exposición de detalles personales, como nombres, fechas de nacimiento y ubicaciones, puede reducir significativamente la eficacia de los ataques de diccionario.

BIBLIOGRAFÍA

AgenciaGUB. (17 de Septiembre de 2024). *Índice global de ciberseguridad (GCI)*. Obtenido de https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/datos-y-estadisticas/estadisticas/indice-global-ciberseguridad-gci#:~:text=El%20%C3%8Dndice%20Global%20de%20Ciberseguridad,Desarrollo%20de%20capacidades%3B%20y%20Cooperaci%C

Ambrissi, R. (15 de Octubre de 2024). *Ecuador está tercero en el ranking de ciberseguridad en América Latina en 2024*.

AWS. (2022). *¿Qué es Python?* Obtenido de https://aws.amazon.com/es/what-is/python/

AWS. (2023). *¿Qué es un hipervisor?* Obtenido de https://aws.amazon.com/es/what-is/hypervisor/#:~:text=Un%20hipervisor%20es%20un%20software,en%20una%20%C3%BAnica%20m%C3%A1quina%20f%C3%ADsica.

BBC. (16 de Septiembre de 2019). *Filtración de Datos en Ecuador.* Obtenido de https://www.bbc.com/mundo/noticias-america-latina-49721456

Concepto. (21 de Junio de 2020). *Sistema operativo*. Obtenido de https://concepto.de/sistema-operativo/

Cybrary. (21 de Enero de 2020). *Using the CUPP tool to generate powerful password lists*.

Obtenido de https://www.cybrary.it/blog/using-cupp-tool-generate-powerful-password-lists?

elcursodelhacker. (9 de Abril de 2022). *Como crear un diccionario de contraseñas con Crunch*. Obtenido de https://www.elcursodelhacker.com/como-crear-un-diccionario-de-contrasenas-con-crunch/

Esgeeks. (2018). *Cómo utilizar Cupp: una guía completa*. Obtenido de https://esgeeks.com/como-utilizar-cupp/

Fernández, Y. (30 de Octubre de 2019). *Qué es Github y qué es lo que le ofrece a los desarrolladores*. Obtenido de https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores

Fortinet. (2017). *¿Qué es un ataque de fuerza bruta?* Obtenido de https://www.fortinet.com/lat/resources/cyberglossary/brute-force-attack

Grigas, L. (1 de Abril de 2022). *Ataque de diccionario*. Obtenido de https://nordpass.com/es/blog/what-is-a-dictionary-attack/

Hostinger. (10 de Junio de 2023). *Qué es GitHub y cómo empezar a usarlo*. Obtenido de https://www.hostinger.es/tutoriales/que-es-github

IBM. (12 de Agosto de 2024). *¿Qué es la ciberseguridad?* Obtenido de https://www.ibm.com/es-es/topics/cybersecurity

Imagina. (14 de Noviembre de 2024). *¿Qué es Kali Linux y para qué se utiliza?* Obtenido de https://imaginaformacion.com/tutoriales/que-es-kali-linux

Imaginaformacion. (2024). *¿Qué es Kali Linux y para qué se utiliza?* Obtenido de https://imaginaformacion.com/tutoriales/que-es-kali-linux

ITU. (2024). *Global cybersecurity index 2024 5th edition*. Obtenido de https://www.itu.int/dms_pub/itu-d/opb/hdb/d-hdb-gci.01-2024-pdf-e.pdf

Kali. (11 de Marzo de 2024). *Crunch | Kali Linux*. Obtenido de https://www.kali.org/tools/crunch/

Kaspersky. (6 de Diciembre de 2017). *¿Qué es la ingeniería social?* Obtenido de https://latam.kaspersky.com/resource-center/definitions/what-is-social-engineering

Kaspersky. (11 de Abril de 2018). *El ransomware: qué es, cómo se lo evita, cómo se elimina*. Obtenido de https://latam.kaspersky.com/resource-center/threats/ransomware

Kaspersky. (8 de Diciembre de 2023). *¿Qué es un ataque de diccionario?* Obtenido de https://latam.kaspersky.com/resource-center/definitions/what-is-a-dictionary-attack

Oliveira, L. (1 de Junio de 2023). *¿Qué es un ataque de diccionario? Aprende a evitarlo*. Obtenido de https://nordvpn.com/es/blog/ataque-de-diccionario/

Ortega, C. (20 de Febrero de 2019). *¿Qué es la investigación documental?* Obtenido de https://www.questionpro.com/blog/es/investigacion-documental/#:~:text=La%20investigaci%C3%B3n%20documental%20es%20una,%2C%20peri%C3%B3dicos%2C%20bibliograf%C3%ADas%2C%20etc.

RedHat. (24 de Enero de 2023). *¿Qué es el open source?*

RedHat. (23 de Marzo de 2023). *¿Qué es un hipervisor?* Obtenido de https://www.redhat.com/es/topics/virtualization/what-is-a-hypervisor

Reydes. (16 de Abril de 2014). *Generar listas personalizadas de palabras con Cewl*. Obtenido de https://www.reydes.com/d/?q=Generar_Listas_Personalizadas_de_Palabras_con_CeWL

Talayero, N. (8 de Octubre de 2024). *Ciberseguridad: ¿Cuál es el panorama actual?*

TelefónicaEcuador. (9 de Febrero de 2024). *Security Forum, de Movistar Empresas, analizó el escenario y tendencias de ciberseguridad en el país*.

TokioSchool. (15 de Septiembre de 2023). *Ataque de diccionario avanzado*. Obtenido de https://www.tokioschool.com/noticias/ataque-diccionario/

TokioSchool. (15 de Septiembre de 2023). *Ataque de diccionario híbrido*. Obtenido de https://www.tokioschool.com/noticias/ataque-diccionario/

TokioSchool. (15 de Septiembre de 2023). *Ataque de diccionario personalizado*. Obtenido de https://www.tokioschool.com/noticias/ataque-diccionario/

TokioSchool. (15 de Septiembre de 2023). *Ataque de diccionario simple*. Obtenido de https://www.tokioschool.com/noticias/ataque-diccionario/

Tomazella, M. (6 de Agosto de 2020). *Paulo Freire: el método de la concientización en la educación, para analizar y comprender el contexto actual de la globalización*.

TrendMicro. (20 de Mayo de 2016). *Banco ecuatoriano pierde 12 millones de dólares vía SWIFT*. Obtenido de https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/ecuadorean-bank-loses-12m-via-swift?utm_source=chatgpt.com

Trevino, A. (23 de Diciembre de 2022). *Como afecta a su privacidad compartir demasiado en las redes sociales*.

Vectra. (6 de Agosto de 2024). *Reconocimiento*. Obtenido de https://es.vectra.ai/topics/reconnaissance#:~:text=%C2%BFCu%C3%A1l%20es%20la%20diferencia%20entre,bas%C3%A1ndose%20en%20datos%20disponibles%20p%C3%BAblicamente.