SEGURIDAD DE LA INFORMACIÓN

Este documento tiene el fin de brindar un pequeño manual a los estudiantes, con un poco de teoría, prácticas y ejemplos de protección de la información. He recopilado información de múltiples documentos reuniendo lo que he considerado las bases y los fundamentos tanto iniciales como básicos para realizar lo que se conoce como “hacking ético”. Mucha de esta información es presentada en muchos documentos, sin embargo la idea es recopilar y mantener un manual de operaciones en seguridad de la información.

Dr. Apolinar González Potes

Ingeniería en Computación Inteligente
UNIVERSIDAD DE COLIMA

1. INTRODUCCIÓN

La ciberseguridad es uno de los campos de más rápido crecimiento dentro de las tecnologías de la información e industria. Cada día, los profesionales de seguridad descubren amenazas nuevas y emergentes a un ritmo rápido y los activos de las organizaciones se ven comprometidos por los actores de amenazas. Debido a estas amenazas en el mundo digital, se están creando nuevas profesiones dentro de muchas organizaciones para personas que pueden ayudar a proteger y salvaguardar sus activos. Este curso está diseñado con la intención de brindarle el conocimiento, la sabiduría y las habilidades que necesita un estudiante para probar estrategias de intrusión dentro de la industria de la seguridad cibernética. Un responsable de seguridad informática es un profesional de la ciberseguridad que tiene las habilidades de un hacker; son contratados por una organización para realizar simulaciones de ciberataques del mundo real en la infraestructura de red de la organización con el objetivo de descubrir y explotar vulnerabilidades de seguridad.

A lo largo de este curso, aprenderá cómo usar una de las distribuciones de Linux más populares dentro de la industria de la seguridad cibernética para simular ataques cibernéticos del mundo real con ejercicios de pruebas de intrusión para descubrir y explotar las debilidades de seguridad en sistemas y redes. El sistema operativo Kali Linux tiene toneladas de paquetes/aplicaciones de Linux preinstalados que se usan ampliamente en la industria de la ciberseguridad, por lo tanto, es un arsenal lleno de todo lo que necesitará. Usaremos un enfoque centrado en el estudiante, lleno de muchos ejercicios prácticos desde el nivel principiante hasta el intermedio.

Inicialmente se obtendrá una comprensión profunda de las diversas características de varios actores de amenazas, sus intenciones y los motivos detrás de sus ataques cibernéticos contra sus objetivos. A continuación, aprenderá sobre los factores clave que son importantes para los actores de amenazas, que determinan el nivel de complejidad para comprometer un sistema en comparación con los profesionales de la seguridad cibernética, como los piratas informáticos éticos y los evaluadores de intrusión que son contratados para descubrir y explotar las debilidades de seguridad ocultas dentro de una organización. Además, también descubrirá la necesidad de pruebas de intrusión, sus fases y enfoques utilizados por profesionales experimentados dentro de la industria.

1.1. TEMÁTICAS DE LA SEGURIDAD DE LA INFORMACIÓN

En todo el mundo, existe una gran demanda de profesionales de ciberseguridad, ya que muchas organizaciones están comenzando a comprender la necesidad de contar con profesionales calificados que los ayuden a asegurar y salvaguardar sus activos. Uno de los activos más valiosos para cualquier organización son los datos. Los actores de amenazas, como los piratas informáticos, están mejorando su plan de juego y la piratería se ha convertido en un negocio en la web oscura. Los actores de amenazas utilizan ataques y amenazas avanzados y sofisticados para comprometer los sistemas y las redes de las empresas, robar sus datos utilizando diversas técnicas de infiltración para eludir la detección de amenazas y vender los datos robados.

1.2. IDENTIFICACIÓN DE LOS ACTORES DE AMENAZA Y SUS INTENCIONES

Hace años, los piratas informáticos realizaban manualmente estas tareas; sin embargo, actualmente se han creado avanzadas amenazas como “ransomware”, que es un crypto-malware diseñado para comprometer sistemas vulnerables. Una vez que un sistema está infectado con “ransomware”, cifra todos los datos dentro de las unidades locales, excepto el sistema operativo. Además, el “ransomware” tiene la capacidad de comprometer también cualquier almacenamiento en la nube que esté vinculado al sistema infectado. Por ejemplo, imagine que el sistema de un usuario tiene Google Drive, Microsoft OneDrive o incluso Dropbox y los datos se sincronizan constantemente. Si el sistema está infectado, la infección también podría afectar los datos dentro del almacenamiento en la nube. Sin embargo, algunos proveedores de nube tienen protección integrada contra este tipo de amenazas.

El “ransomware” cifra los datos y los retiene mientras presenta una ventana de pago en el escritorio de la víctima solicitando el pago para recuperar los datos. Durante este tiempo, el actor de amenazas responsable también extrae sus datos y los vende en los mercados de web obscuros.

1.3. COMPRENDER LO QUE ES IMPORTANTE PARA LOS ACTORES DE AMENAZAS

El concepto de hackear otro sistema o red siempre les parecerá muy fascinante a muchos, mientras que para otros es bastante preocupante saber que el nivel de seguridad no es aceptable si un sistema puede verse comprometido por un actor de amenazas. Los actores de amenazas, los piratas informáticos éticos o incluso los evaluadores de penetración deben planificar y evaluar el tiempo, los recursos, la complejidad y el valor del pirateo antes de realizar un ataque cibernético en los sistemas o redes de un objetivo.

1. Tiempo. Comprender cuánto tiempo pasará desde que se comienza a recopilar información sobre el objetivo hasta que se cumple los objetivos del ataque es importante. A veces, un ciberataque puede llevar a un actor de amenazas desde días hasta algunos meses de planificación cuidadosa para garantizar que cada fase tenga éxito cuando se ejecuta en el orden correcto. Los actores de amenazas también deben tener en cuenta la posibilidad de que un ataque o explotación no funcione en el objetivo y esto crea un bache durante el proceso, lo que aumenta el tiempo necesario para cumplir los objetivos del hackeo. Este concepto se puede aplicar a los evaluadores de penetración, ya que necesitan determinar cuánto tiempo llevará completar una prueba de penetración para un cliente y presentar el informe con los hallazgos y las recomendaciones de seguridad.
2. Recursos. Sin recursos, será un desafío completar una tarea. Los actores de amenazas deben tener el conjunto adecuado de recursos, que pueden ser herramientas basadas en software y hardware. Si bien los piratas informáticos expertos y experimentados pueden descubrir y explotar manualmente las debilidades de seguridad en un sistema, puede ser un proceso que requiere mucho tiempo. Sin embargo, usar el conjunto adecuado de herramientas puede ayudar a automatizar estas tareas y mejorar el tiempo necesario para encontrar fallas de seguridad y explotarlas. Además, sin el conjunto adecuado de habilidades, un actor de amenazas puede enfrentar algunos desafíos para tener éxito en la realización del ciberataque. Esto puede conducir a obtener el apoyo de personas adicionales con las habilidades necesarias para ayudar y contribuir a lograr los objetivos del ciberataque. Una vez más, este concepto se puede aplicar a los profesionales de la seguridad, como los “tester” de intrusión dentro

de la industria. No todos tienen las mismas habilidades y es posible que se necesite un equipo para realizar una prueba de intrusión para un cliente.

3. Factores financieros. Otro recurso importante son los factores financieros. A veces, un actor de amenazas no necesita más recursos y puede realizar un exitoso ataque cibernético y comprometer sus objetivos. Sin embargo, puede haber momentos en los que se necesite una herramienta adicional basada en software o hardware para garantizar que el ataque tenga éxito. Tener un presupuesto permite a los actores de amenazas comprar los recursos adicionales necesarios. Del mismo modo, los evaluadores de intrusión están bien financiados por sus empleadores para garantizar que tengan acceso a las mejores herramientas dentro de la industria para sobresalir en sus trabajos.
4. “Hack value”. Por último, el valor de pirateo es simplemente la motivación o el motivo para realizar un ciberataque contra los sistemas y la red de una empresa. Para un actor de amenazas, es el valor de lograr los objetivos y metas de comprometer el sistema. Es posible que los actores de amenazas no apunten a una organización si creen que no vale la pena el tiempo, el esfuerzo o los recursos para comprometer sus sistemas. Otros actores de amenazas pueden apuntar a la misma organización con otro motivo.

1.4. DESCUBRIENDO TERMINOLOGÍAS DE CIBERSEGURIDAD

La siguiente es una lista de las terminologías más comunes dentro de la industria de la ciberseguridad:

Tarea Bonificadora:Hacer una lista y una descripción de la terminología más común en ciberseguridad

1.5. EXPLORANDO LA NECESIDAD DE PRUEBAS DE INTRUSIÓN Y SUS FASES

Las organizaciones o empresas se están dando cuenta de la necesidad de contratar a hackers, como evaluadores de intrusión cibernética con habilidades para simular ataques cibernéticos del mundo real en los sistemas y redes de la organización con la intención de descubrir y explotar vulnerabilidades ocultas. Estas técnicas permiten que el evaluador de intrusiones realice los mismos tipos de ataques que un hacker real; la

diferencia es que el hacker es contratado por la organización y se le ha otorgado permiso legal para realizar tales pruebas de seguridad intrusivas.

Los hackers generalmente tienen una sólida comprensión de las computadoras, los sistemas operativos, las redes y la programación, así como también cómo funcionan juntos. Lo más importante es que necesitas creatividad. El pensamiento creativo le permite a una persona pensar fuera de su entorno e ir más allá de los usos previstos de las tecnologías y encontrar nuevas formas de implementarlas.

Al final de la prueba de hackeo, se presenta un informe a las partes interesadas de la organización, detallando todos los hallazgos, como las vulnerabilidades y cómo se puede explotar cada debilidad. El informe también contiene recomendaciones sobre cómo mitigar y prevenir un posible ataque cibernético en cada vulnerabilidad encontrada. Esto permite a la organización comprender qué descubrirá un pirata informático si es un objetivo y cómo implementar contramedidas para reducir el riesgo de un ataque cibernético. Algunas organizaciones incluso realizan una segunda prueba de hackeo después de implementar las recomendaciones descritas en el informe de la prueba inicial para determinar si se han solucionado todas las vulnerabilidades y se ha reducido el riesgo.

1.6. COMPRENDER LOS ENFOQUES DE LAS PRUEBAS DE INTRUSIÓN

Una evaluación de “white box” típica de aplicaciones web pero puede extenderse a cualquier forma de prueba de hackeo. La diferencia clave entre las pruebas de “white box”, “black box” y “gray box” es la cantidad de información proporcionada a los hackers antes del compromiso. En una evaluación de ”white box”, el hacker recibirá información completa sobre la aplicación y sus tecnologías, y generalmente se le otorgarán credenciales con diversos grados de acceso para identificar de manera rápida y completa las vulnerabilidades en las aplicaciones, sistemas o redes. No todas las pruebas de seguridad se realizan utilizando el enfoque de “white box”; a veces, solo se proporciona el nombre de la empresa objetivo al hacker.

Las evaluaciones de “black box” son la forma más común de evaluación de la intrusión de la red y son los más típicos entre pruebas de intrusión de redes externas y pruebas de intrusión de ingeniería social. En una evaluación de “black box”, los hackers reciben muy poca o ninguna información sobre las redes o sistemas de destino que están

probando. Esta forma particular de prueba es eficiente cuando se trata de determinar qué descubrirá un pirata informático real y sus estrategias para obtener acceso no autorizado a la red de la organización y comprometer sus sistemas.

Las evaluaciones de “gray box” son un híbrido de pruebas de white box” y “black box” y generalmente se usan para proporcionar un escenario de prueba realista al tiempo que brinda a los hackers suficiente información para reducir el tiempo necesario para realizar el reconocimiento y otras actividades de prueba de “black box”. Además, es importante en cualquier evaluación asegurarse de que está probando todos los sistemas incluidos en el alcance. En una verdadera “black box”, es posible que se pierdan sistemas y, como resultado, queden fuera de la evaluación.

2. METODOLOGÍAS DE HACKING ÉTICO

Las metodologías nos facilitan la realización de un conjunto de actividades en un orden determinado y estableciendo una prioridad adecuada para intentar garantizar el éxito y alcanzar un objetivo final.

2.1 METODOLOGÍAS PRINCIPALES

- OSSTMM (Open-Source Security Testing Methodology Manual) https://www.isecom.org/OSSTMM.3.pdf
- The Penetration Testing Execution Standard: http://www.pentest-standard.org/index.php/Main_Page
- ISSAF (Information Systems Security Assessment Framework)
- OTP (OWASP Testitng Project)

2.2 METODOLOGÍA DE ESTE CURSO

- ***Definición del alcance del test de intrusión***

Esto es algo que realizamos con el cliente o con la organización a la que le vayamos a hacer ese Hacking ético

- ***Recopilación de información***

La fase de recopilación de información la dividiremos en tres fases: recopilación pasiva de información, recopilación semi-pasiva de información y recopilación activa de información.

- ***Identificación y análisis de vulnerabilidades***

Que es una fase de mucha intuición de acuerdo a la información que se va recopilando.

- ***Explotación de las vulnerabilidades***

Después identificación y análisis de vulnerabilidades, explotación de las vulnerabilidades que también lo dividiremos entre sus fases de explotación de vulnerabilidades en host, de explotación de vulnerabilidades en aplicaciones web y explotación de vulnerabilidades en redes.

- ***Post-explotación***

Medidas a tomar de acuerdo a los análisis realizados.

- ***Elaboración de un documento de reporte***

Ejemplos de informes de Hacking Ético y auditoría de seguridad
Una de las preguntas más frecuentes que pueden surgir después es la siguiente:

¿Cómo es un informe real de Hacking Ético o auditoría de seguridad?
Para resolver esta duda y con ejemplos reales de cara a poder organizar de la mejor forma posible los informes algunos recursos valiosos a revisar son:

Antes que nada, se debe tener en cuenta que los informes de Hacking ético y auditoría de seguridad dependen mucho de la organización que los realiza y del tipo de auditoría

que se ha llevado a cabo. No todas las auditorías son completas y siguen todas las fases que se enseñan o comentan en este curso, en algunas ocasiones se centran en fases o entornos específicos dentro de la infraestructura tecnológica de una organización. Todo esto se debe concretar en la fase de definición del alcance que se mencionaba en la sección anterior.

El primer recurso a compartir es un repositorio donde se pueden encontrar cientos de reportes de auditorías reales de diferentes empresas del sector del Hacking Ético y de la Ciberseguridad que se han ido recopilando a lo largo del tiempo. En este repositorio se tienen informes de todo tipo y que recogen una diversidad muy grande de auditorías:

https://github.com/juliocesarfort/public-pentesting-reports

El segundo recurso corresponde a diferentes plantillas que se pueden utilizar para comenzar a elaborar un propio informe de Hacking Ético o auditoría de seguridad e ir modificándolo para que se adapte a las necesidades:
https://pentestreports.com/templates/
https://github.com/hmaverickadams/TCM-Security-Sample-Pentest-Report

2.3 DEFINICIÓN DEL ALCANCE DEL HACKING ÉTICO

- Antes de realizar ninguna acción, discutir con el cliente las tareas que llevará a cabo el analista durante el Hacking Ético, así como los roles y responsabilidades de ambos

- Asegurar mediante contrato firmado que las acciones que se llevan a cabo son en representación del cliente

- Análisis de las políticas de la organización que definen el uso que los usuarios hacen de los sistemas y de la infraestructura

- Procedimiento el caso de que se localice una intrusión por un tercero

3. FASE DE RECOPILACIÓN PASIVA DE LA INFORMACIÓN

Concretamente, esta fase es una de las más relevantes porque es la que nos va a permitir obtener los datos iniciales con los que después vamos a continuar en las siguientes fases del proceso de “hacking ético”.

Esta fase de recopilación de información nosotros vamos a dividirla en tres fases: la recopilación pasiva de información, la recopilación semipasiva de información y la recopilación activa de información.

Bien, el objetivo de estas tres fases, como su propio nombre indica, va a consistir en tratar de obtener todos los datos, toda la información posible sobre nuestro objetivo. Si, por ejemplo, estamos realizando nuestro ejercicio de hacking ético sobre una organización determinada, entonces trataremos de obtener toda la información que podamos de la infraestructura tecnológica de esa organización, de manera que facilite las siguientes fases del proceso de hacking ético.

En este curso todas las fases las vamos a ir viendo de manera incremental o sea se van a ir construyendo sobre los hallazgos de la fase anterior, la fase de recopilación de información, después análisis de vulnerabilidades sobre esos activos tecnológicos que hemos encontrado, después, explotación de las vulnerabilidades que hemos encontrado y así sucesivamente.

La recopilación pasiva de información, como su propio nombre indica. Va a consistir en tratar de obtener todos los datos posibles sobre un objetivo, interactuando lo mínimo posible con él, a través de información almacenada en lugares públicos.

3.1. HACKING ÉTICO CON BUSCADORES: GOOGLE HACKING

El concepto “hacking de Google”, a veces denominado “Google dorking”, no es el proceso de piratear la infraestructura o los sistemas de la red de Google, sino el uso de parámetros de búsqueda avanzada dentro del motor de búsqueda de Google para filtrar resultados específicos. Muchas organizaciones no siempre prestan mucha atención a los sistemas y recursos que exponen en Internet. Google es un motor de búsqueda muy poderoso que rastrea/indexa todo en Internet y filtra la mayoría de los sitios web maliciosos. Dado que Google indexa todo, el motor de búsqueda puede descubrir automáticamente los directorios, recursos y portales de inicio de sesión en línea ocultos de muchas organizaciones.

El uso de técnicas de piratería de Google no es ilegal, pero hay una línea muy fina que no se debe cruzar; de lo contrario, estará en problemas legales. Podemos usar las técnicas de piratería de Google para descubrir ubicaciones ocultas y confidenciales en Internet, pero si usa esa información contra una organización, entonces se convierte en un problema.

3.2 PRÁCTICA DE GOOGLE HACKING

Usualmente google nos brinda ciertos comandos de esta manera:
comando:consulta

ejemplo:
site:ucol.mx archivos pdf
site:ucol.mx filetype:pdf

los principales comandos de búsqueda avanzada de Google:

” ” (comillas): buscar frase exacta
and or not: operadores lógicos “y” o “no”
+ y -: incluír y excluír. Ej: jaguar -coches: busca la palabra “jaguar”, pero omite las webs con la
palabra “coches”
* (asterisco): comodín, cualquier palabra, pero una sóla palabra
. (punto): comodín, cualquier palabra, una o muchas

- intitle o allintitle: la expresión buscada está en el título
- inurl o allinurl: la expresión buscada está en la url
- site: sólo busca resultados dentro de la web que va detrás de “site:”
- filetype: sólo busca archivos de un tipo (doc, xls, txt…)
- link: sólo busca en páginas que tienen un link a una determinada web
- inanchor: sólo busca en páginas que tienen en el texto de enlace la expresión buscada
- cache: muestra el resultado en la cache de Google de una página web
- related: busca webs relacionadas con una determinada

1. Para realizar una recopilación de información pasiva en una organización objetivo. Una técnica muy común es simplemente realizar una búsqueda en Google sobre la organización. Sin embargo, si desea ver resultados específicos que contengan solo el dominio de destino, use site:domain.com:

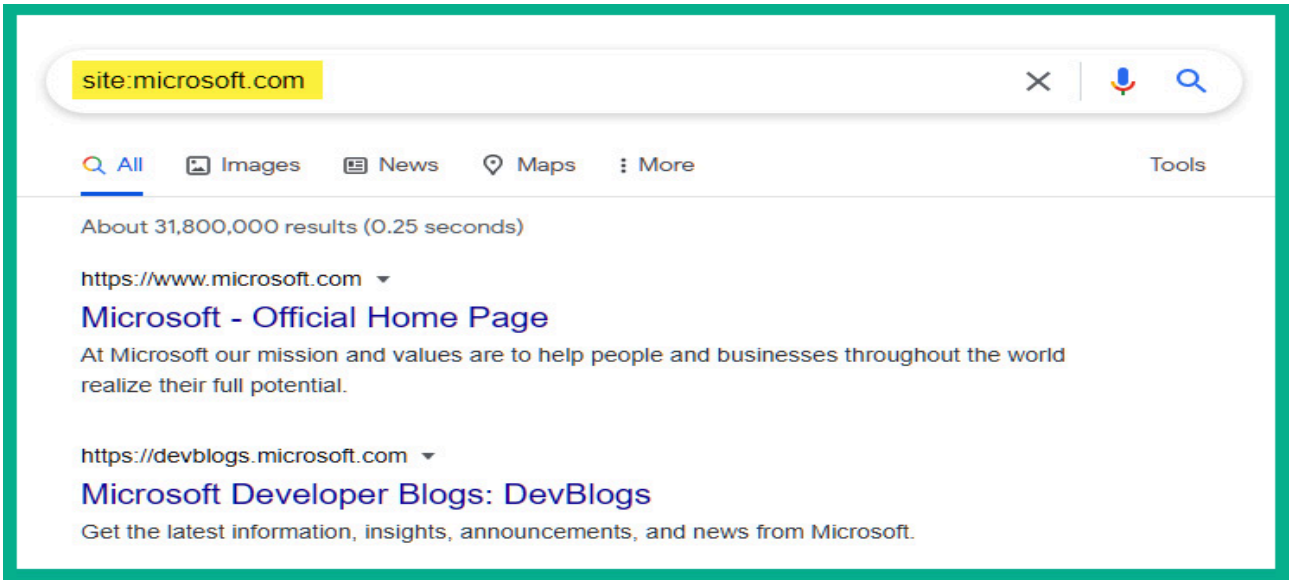


Fig. 1 Filtrado de resultados de un dominio

2. Para realizar la búsqueda que contengan una palabra clave pero solo del dominio de destino. Aquí, puede usar keyword site:domain.com :

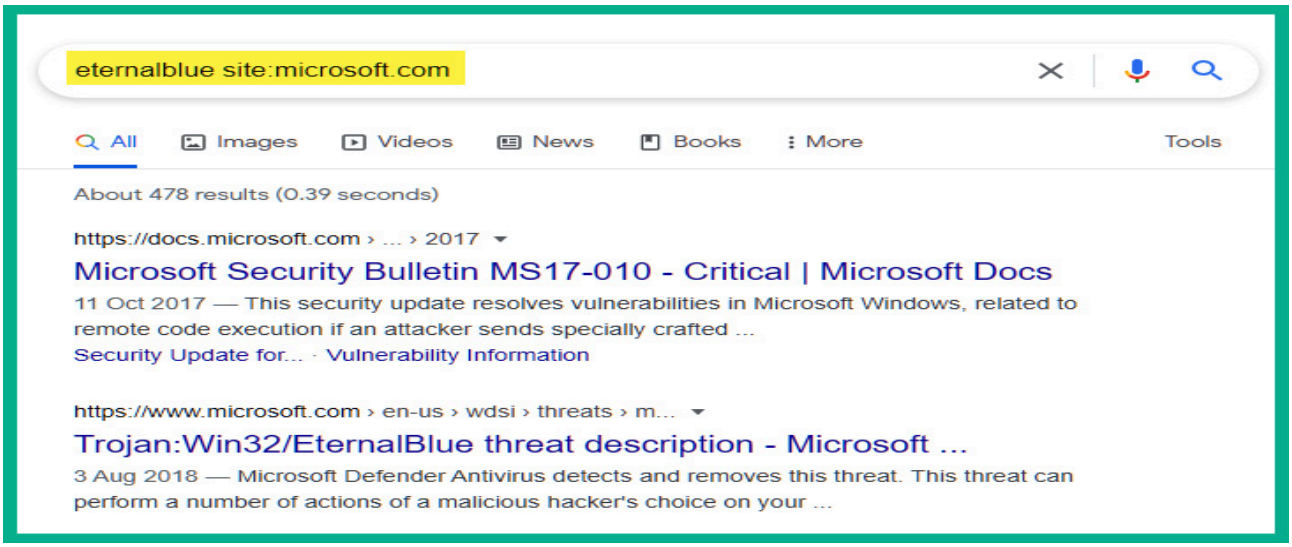


Fig 2. Búsqueda por palabra clave

3. Para filtrar sus resultados de búsqueda para que incluyan dos palabras clave específicas, puede usar keyword1 AND keyword2 site:domain.com :

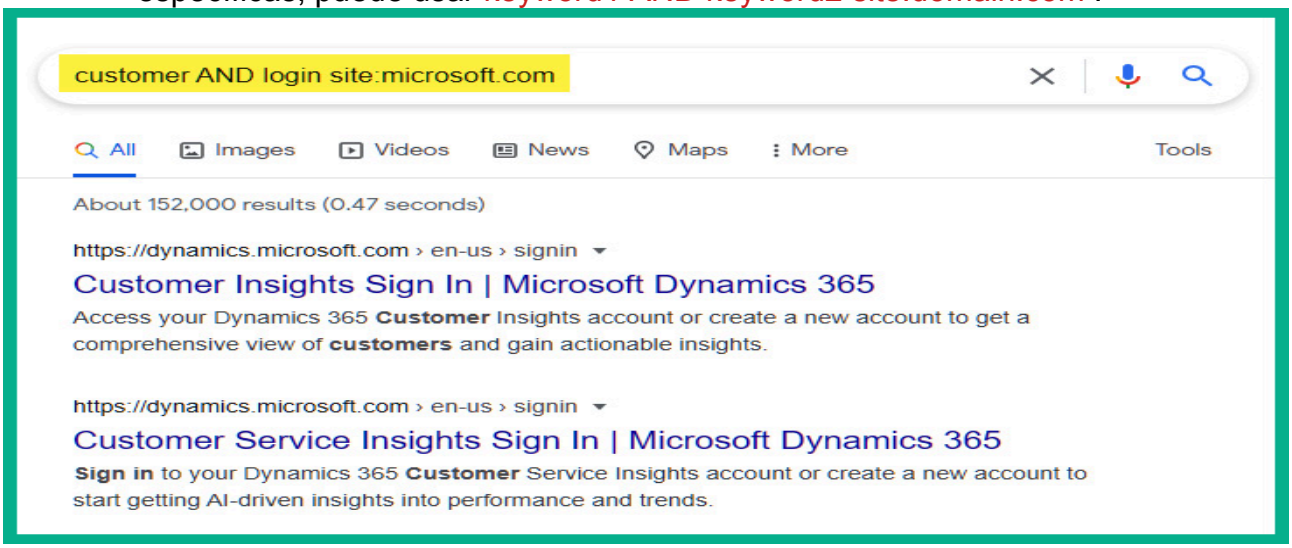


Fig 3. Búsqueda con varias palabras clave

4. Para filtrar los resultados de búsqueda para mostrar un tipo de archivo específico de un dominio de destino, use la sintaxis site:domain.com filetype:file type

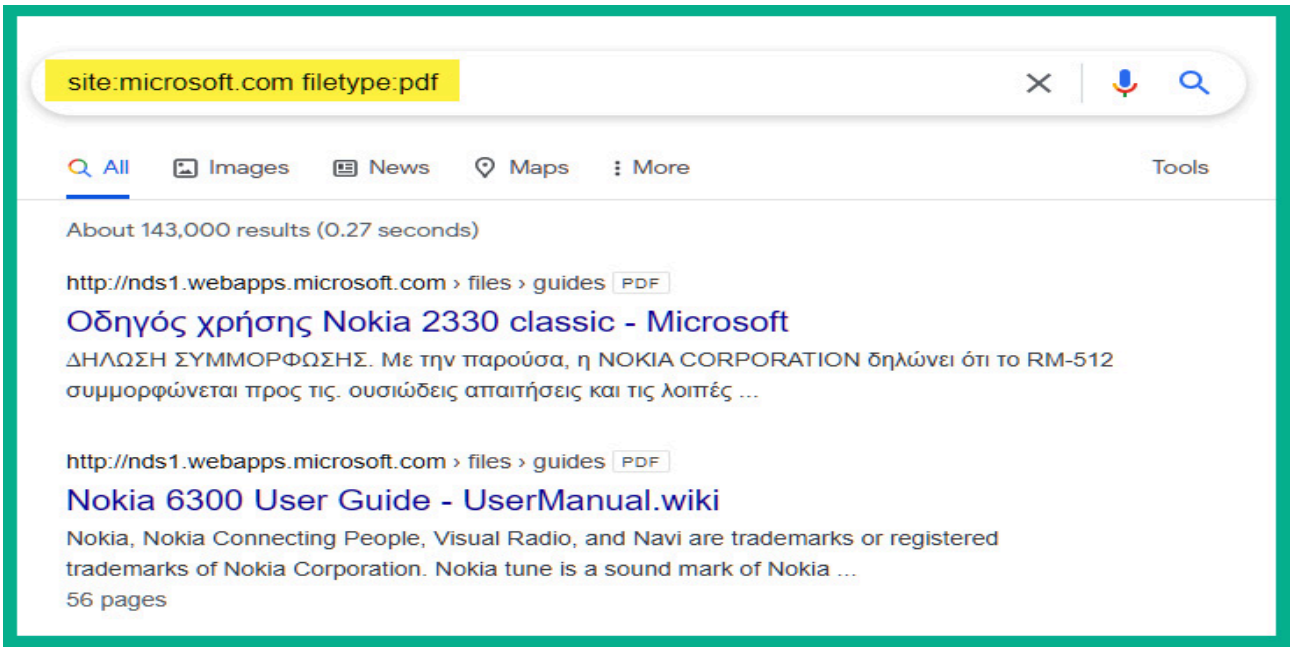


Fig 4. Filtrado por tipo de archivos.

5. Para descubrir direcciones URL específicas que contienen una palabra clave específica dentro del título de su página, use site:domain.com intitle:keyword

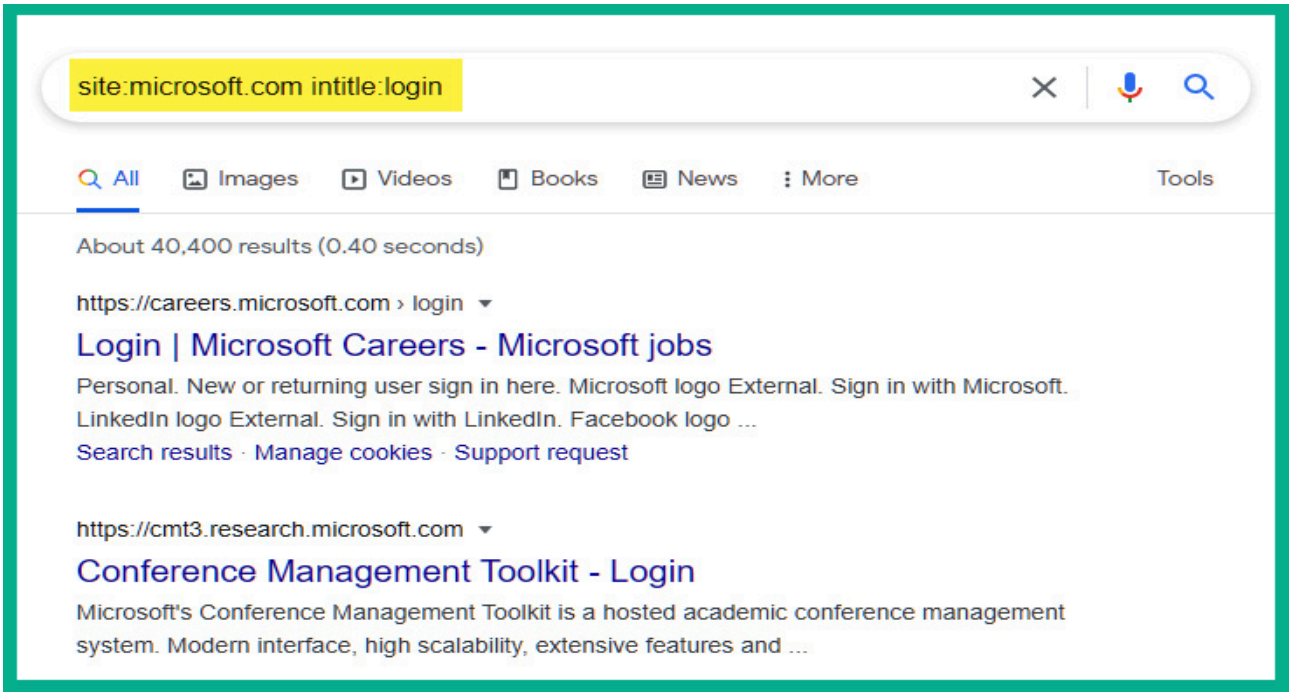


Fig 5. Filtrado por URL específicas

6. Para eliminarlos resultados de visualización de las URL para un dominio de destino que no incluye una palabra clave específica, use:
site:domain.com –keyword

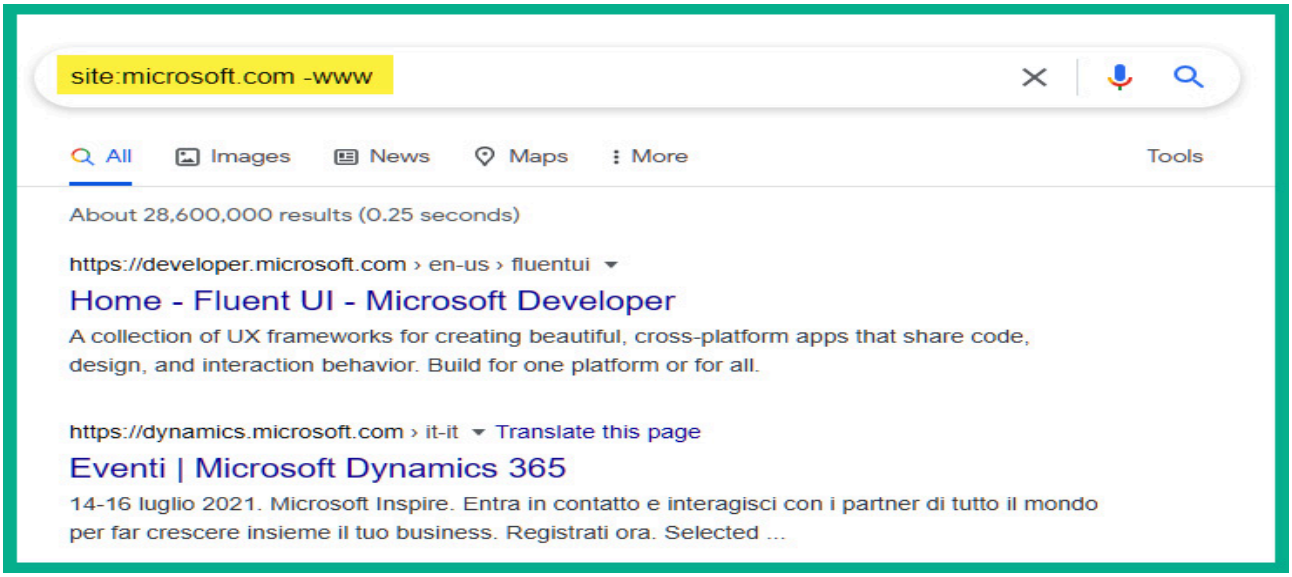


Fig 6. Visualización de las URL para un dominio

7. Ejemplo para ver diferentes cámaras, entre una de las técnicas que utilizaría un “hacker”

   Existen miles de códigos para ver las cámaras, aquí le mencionaré varios:

   Código: inurl:/view.shtml

   Código: inurl:ViewerFrame?Mode=

   Código: inurl:ViewerFrame?Mode=Refresh

   Código: inurl:axis-cgi/jpg

   Código: inurl:axis-cgi/mjpg (motion-JPEG)

   Código: inurl:view/indexFrame.shtml (motion-JPEG)

   Código: inurl:view/index.shtml

   Código: inurl:view/view.shtml

   Código: intitle:axis intitle:”video server”

Observe la cantidad de resultados que puede arrojar una búsqueda sencilla como inurl:/view.shtml. Hay miles de resultados de posibles cámaras que probablemente estén abiertas y que pueden entrar, claro esto no necesariamente es interesante, pero qué pasaría si se encuentra una cámara que constantemente esté grabando una oficina, quizás puedan rastrear algún código editado por algún usuario que está siendo visto por la cámara.

8. Ejemplo de búsqueda de contraseñas

Algunas personas guardan sus contraseñas en archivos de texto plano. Si no se tiene una buena protección podrán aparecer en la siguiente búsqueda:

filetype:xlsx intext:password intext:username

3.3. EXPLORANDO EL RECONOCIMIENTO DE DNS

El DNS es un protocolo de capa de aplicación que permite a una computadora (sistema computacional) resolver un nombre de host en una dirección IP. Si bien hay tantos dispositivos en una red, especialmente en Internet, recordar la dirección IP de cada servidor web es todo un desafío. Usando DNS, un administrador del sistema puede configurar cada dispositivo con una dirección IP y un nombre de host. Usar un nombre de host es mucho más fácil de recordar, como www.packtpub.com o www.google.com . Sin embargo, ¿conoce las direcciones IP de los servidores que alojan esos sitios web para Packt y Google? Probablemente no, y está bien porque en todo el mundo en Internet existe una jerarquía de servidores especiales que contienen los registros de nombres de host públicos y sus direcciones IP. Estos se conocen como servidores DNS.

Un servidor DNS es como una guía telefónica tradicional, con una lista de personas y sus números de teléfono. En un servidor DNS, puede encontrar registros de los nombres de host de las personas, así como sus direcciones IP asociadas, que son similares a los números de teléfono.

Muchas empresas populares de Internet, como Cisco, Google, Cloudflare y otros han establecido muchos servidores DNS públicos en Internet que contienen registros de casi todos los nombres de dominio/host públicos en Internet. Para entender el DNS, veamos un ejemplo simple.

Para visitar un sitio web, como www.example.com:

Cada vez que una computadora o dispositivo necesita resolver un nombre de host en una dirección IP, envía un mensaje de consulta DNS a su servidor DNS, como se indica en el siguiente diagrama:

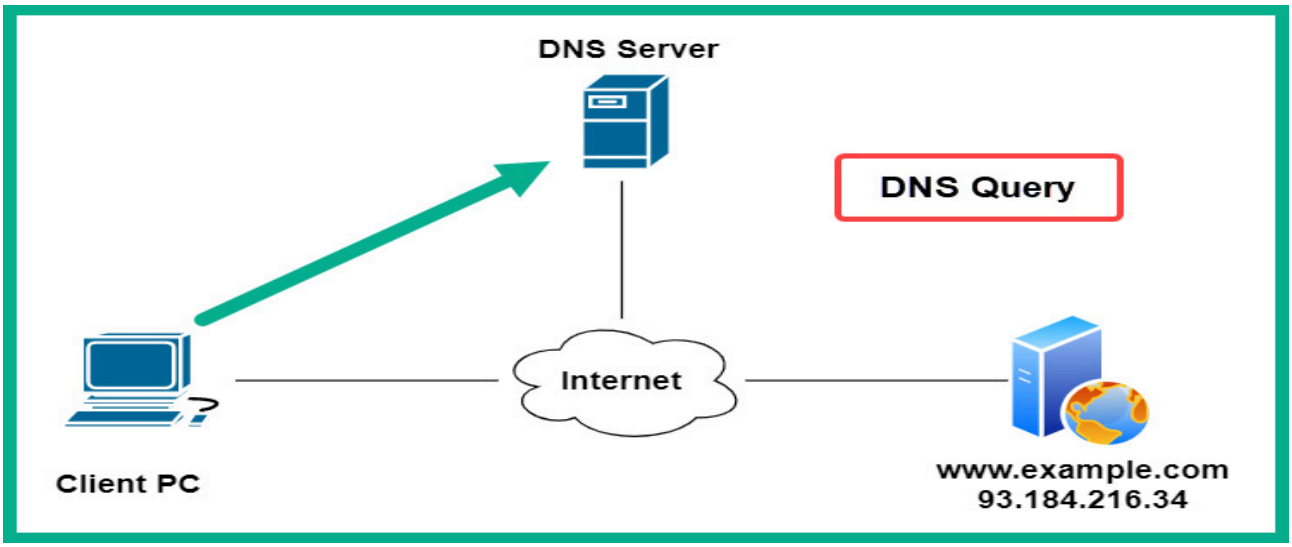


Fig.7. Consulta de DNS

El servidor de dns verificará sus registros y responderá con una respuesta DNS, proporcionando a la computadora cliente la dirección IP del dominio, como se muestra en el siguiente diagrama:

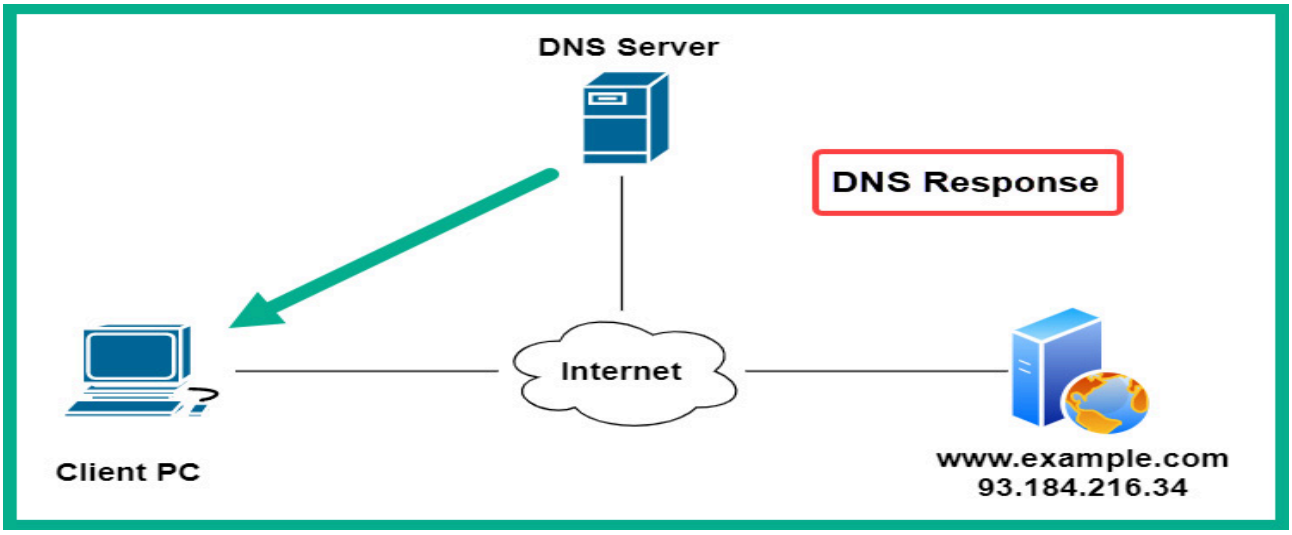


Fig.8. Respuesta DNS

Finalmente, el cliente recibe la dirección IP y establece una sesión entre él y el dominio https://www.example.com/, como se muestra en el siguiente diagrama:

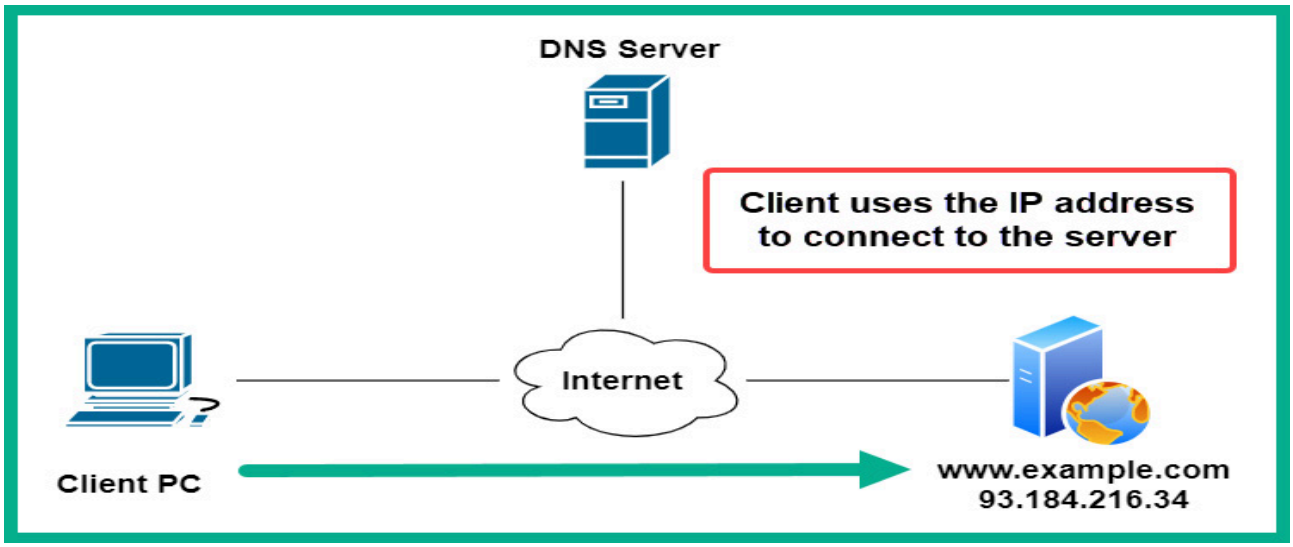


Fig. 9. Cliente estableciendo una conexión.

Existen muchos servidores DNS públicos en Internet; algunos son creados con intenciones maliciosas, como redirigir a usuarios desprevenidos a sitios web maliciosos. Como resultado, es recomendable usar un proveedor de DNS confiable en todos sus dispositivos de red y computadoras para mejorar su seguridad en línea. Además, los servidores DNS no solo resuelven un nombre de host en una dirección IP, sino que también contienen varios registros que se utilizan para varios tipos de resolución. Los siguientes son los diferentes tipos de registros:

- A: Resolves a hostname to an IPv4 address.
- AAAA: Resolves a hostname to an IPv6 address.
- NS: Contains the name servers' information.
- MX: Contains the mail exchange (email) servers.
- PTR: Resolves an IP address to a hostname.
- CNAME: Provides a canonical name or an alias.
- RP: Specifies the person that's responsible for the domain.
- SOA: Contains information about the administrator of the domain.
- SRV: Contains a service port number for a specific service of the domain.

La enumeración de DNS es una técnica de sondear registros de DNS específicos para el dominio de una organización específica. En otras palabras, se le pregunta a un servidor DNS sobre las direcciones IP y los nombres de servidor para una organización de destino. Es decir, puede recuperar tanto el nombre de host como las direcciones IP de los servidores públicos de un objetivo, como sus servidores de correo electrónico.

3.4. PRÁCTICA DE SERVIDORES DNS

Dentro de Kali Linux, encontrará muchas herramientas que pueden realizar la enumeración de DNS de un dominio de destino.Realización de la enumeración de DNS con DNSRecon. Para automatizar la tarea de realizar la enumeración de DNS, así como comprobar si el servidor DNS del objetivo se ha configurado incorrectamente para permitir transferencias de zona no autenticadas.

1. Ejecute:

```
┌──(apogon㉿kali-1)-[~]
└─$ dnsrecon -h
```

2.

```
┌──(apogon㉿kali-1)-[~]
└─$ dnsrecon -d microsoft.com
```

3.5 PRÁCTICA DE OBTENCIÓN DE SUBDOMINIOS CON PYTHON

Inicialmente se debe crear un ambiente virtual de python con los siguientes pasos
1 virtualenv seguridad
2. source seguridad/bin/activate

Una vez ya en en el ambiente virtual de Python se debe instalar las librerías dnspython, requests y argparse:
pip install dnspython
pip install requests

En el archivo subdominios.txt se encuentra un listado grande de posibles subdominios que usualmente se configuran al interior de un dominio. El objetivo es hacer un “testing” de que subdominios se pueden encontrar detrás de un dominio dado. En esta primera fase recuerden que la idea es gestionar información que pueda ser útil para posteriormente analizar vulnerabilidades e incluso hacer algún ataque. También es importante recordar que vulnerar una máquina es un proceso muy complejo y a veces

casi imposible, pero, si hay vulnerabilidades puede ser muy fácil hacer un hackeo, por lo tanto, una de las fases importantes es obtener información que pueda ser útil para hacer una explotación de las vulnerabilidades.

Ejecutar el programa por ejemplo con el dominio wikipedia.com.

3.6 PRÁCTICA DE INTRODUCCIÓN A CRUNCH E HYDRA

Esta práctica es una introducción inicial a técnicas de fuerza bruta, en realidad el primer paso es recolectar información, pero vamos a ir alternando las aplicaciones poco a poco. Posteriormente, haremos una introducción al encriptamiento y el uso de las claves y por ejemplo el ssh.

En el mundo del hacking tenemos una técnica llamada fuerza bruta, bien vista por algunos y mal vista por otros, pero no es este el lugar de debatir sobre la calidad de la técnica, pues según wikipedia la fuerza bruta es: *"La forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso."*

Crunch es un programa que basándose en criterios establecidos por el usuario (input) es capaz de generar diccionarios para ser usados en fuerza bruta (output), el resultado de Crunch puede ser visto en pantalla, puede ser guardado en un archivo .txt ó puede enviarse a otro programa en tiempo real para su uso.

```
┌──(apogon㉿kali-1)-[~]
└─$ crunch 2 3 -o 0.txt
```

Observe la salida, con la cual se puede saber cómo funciona la herramienta.

Hay muchas opciones y combinaciones para obtener una gran posibilidad de posibles contraseñas, sin embargo recuerden que el objetivo principal es poder hacer inicialmente una búsqueda que nos de indicios de vulnerabilidades. Este ejercicio es muy simple y partimos de que conocemos las contraseñas de un usuario.

Hydra es un muy, pero muy conocido en el mundo del “hacking” y esta herramienta intenta crakear por fuerza bruta la contraseña de una cantidad impresionante de protocolos: TELNET, FTP, HTTP, HTTPS, HTTPPROXY, SMB, SMBNT, MS-SQL, MYSQL, REXEC,RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP2, LDAP3, Postgres, Teamspeak, Cisco auth, Cisco enable, AFP, LDAP2, Cisco AAA (incorporado en el módulo de Telnet).

Con la herramienta hydra se realiza un ataque a un determinado host colocando contraseñas en un determinado protocolo con el objetivo de hacer login usando un determinado protocolo. Hay diferentes métodos por los que pueden robar nuestras contraseñas y uno de ellos es lo que se conoce como **fuerza bruta**. Consiste básicamente en probar una tras otra diferentes combinaciones hasta dar con la que realmente permite iniciar sesión en una cuenta.

Estos ataques de fuerza bruta se basan en **diccionarios**, donde se almacenan palabras o números más comunes. De esta forma pueden combinar las posibles claves y tener mayor oportunidad. Si un usuario ha puesto una contraseña débil, como podría ser una palabra, un nombre o una fecha, es mucho más sencillo explotarla. Hay **diferentes programas** que pueden actuar para lograr romper una clave, como es el caso de Hydra. Es una de las aplicaciones más utilizadas y conocidas en hacking ético para poner a prueba contraseñas. No obstante, también puede ser usada por un atacante que realmente tenga el objetivo de robar nuestra clave.

Uso básico de hydra:

```
┌──(apogon㉿kali-1)-[~]
└─$ hydra 192.168.68.XXX ssh -s 22 -l kali -P 1.txt -f -vV
```

- 192.168.68.xxx es la ip de la máquina donde se realizará el ataque
- ssh es el protocolo
- -s 22 índica el puerto
- -l kali el usuario
- -f para que una vez que encuentre una contraseña finalice el testing
- -P 1.txt es nuestro diccionario
- -vV es el verbose con bastante detalle.

Para efectos de realizar la práctica, vamos a cambiar el password del usuario linux por una contraseña corta de 4 letras máximo (actualmente es linux, podemos cambiarla a una contraseña como secu por ejemplo).

Identifiquen una IP de otro grupo (compañeros) y hagan las pruebas de hackeo con las dos herramientas.

```
┌──(apogon㉿kali-1)-[~]
└─$ hydra 192.168.68.XXX ssh -s 22 -L users.txt -P 1.txt -f -vV
```

Otra opción de uso es colocar -L users.txt donde el archivo contiene una lista de posibles usuarios de una máquina, así el testing no se hace sobre un usuario en particular, sino sobre una lista de usuarios posibles.

4. Detección de vulnerabilidades

El motor de secuencias de comandos de Nmap (NSE) es una de las funciones más potentes de Nmap. Permite evaluar la seguridad para crear, automatizar y realizar escaneos personalizados en un sistema o red de destino. Cuando las técnicas de escaneo, éstas suelen ser agresivas y, a veces, pueden provocar la pérdida de datos o incluso bloquear un objetivo del sistema. Sin embargo, NSE permite que un probador de testing identifique fácilmente las vulnerabilidades de seguridad y si el objetivo es explotable.

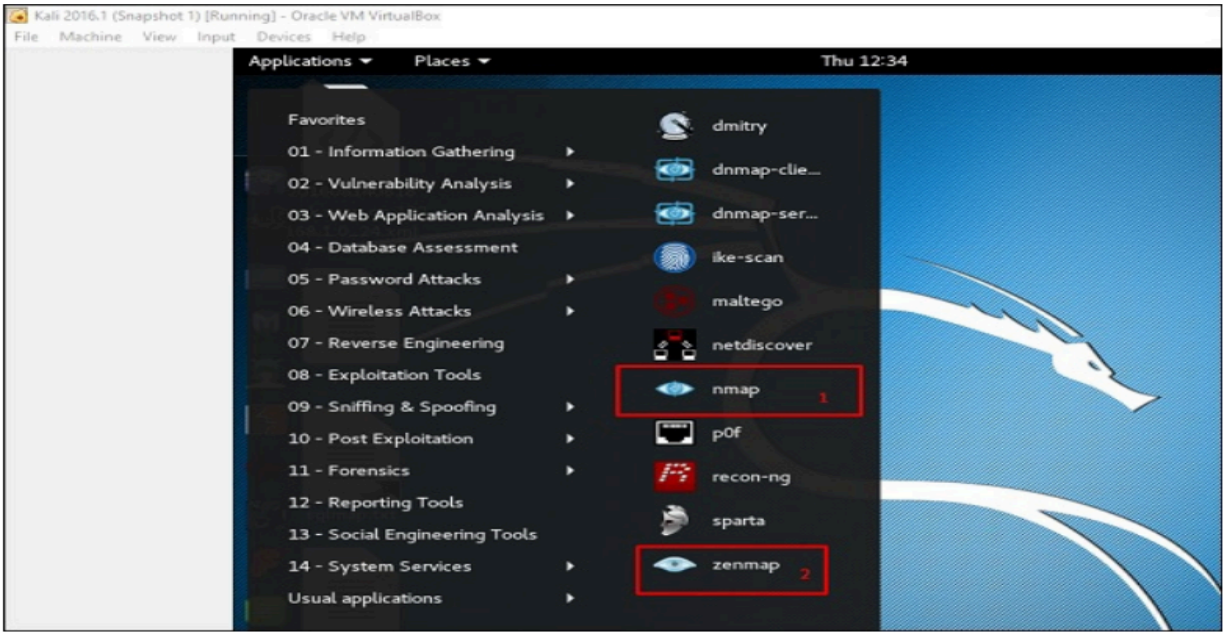
4.1 Sniffers: Wireshark

Las computadoras se comunican usando redes. Estas redes pueden estar en una LAN de red de área local o estar expuestas a Internet. Los Network Sniffers son programas que capturan datos de paquetes de bajo nivel que se transmiten a través de una red. Un atacante puede analizar esta información para descubrir información valiosa, como ID de usuario y contraseñas.
Abrir wireshark y descubrir inicialmente sus funciones, la interfaz es muy sencilla.
Iniciar una aplicación web y abrir alguna página, observe e identifique el funcionamiento de la herramienta.

4.2 Nmap

NMap es el programa para escaneo de puertos por excelencia, sirve para el bien y para el mal. NMAP y ZenMAP son herramientas útiles para la fase de escaneo de Ethical Hacking en Kali Linux. NMAP y ZenMAP son prácticamente la misma herramienta, sin embargo, NMAP usa la línea de comandos mientras que ZenMAP tiene una GUI.

1. Lo primero será ejecutar el comando nmap para ver las opciones que nos brinda.




```
┌──(apogon㉿kali-1)-[~]
└─$ nmap
Nmap 7.92 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL <inputfilename>: Input from list of hosts/networks
```

-iR <num hosts>: Choose random targets
--exclude <host1[,host2][,host3],...>: Exclude hosts/networks
--excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
Hay más opciones pero se las dejo resumidas

2. Lo segundo que vamos a hacer es escanear toda la red para elegir un objetivo a analizar el rango de ip. Observe cómo se vería (es un ejemplo), sin embargo en el laboratorio tendremos otras ip y otras máquinas en ejecución.

```
┌──(apogon㉿kali-1)-[~]
└─$ nmap 192.168.68.0/24 ****(esta red la veremos en el laboratorio)
Starting Nmap 7.80 (https://nmap.org) at 2019-09-15 11:55 EDT
Nmap scan report for 192.168.171.1
Host is up (0.00069s latency).
Not shown: 996 filtered ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
869/tcp open icslap
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.171.2
Host is up (0.00030s latency).
Not shown: 999 closed ports
PORT STATE SERVICE
53/tcp open domain
MAC Address: 00:50:56:F3:A1:03 (VMware)
Nmap scan report for 192.168.171.129
Host is up (0.00076s latency).
Not shown: 991 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
139/tcp open netbios-ssn
143/tcp open imap
443/tcp open https
445/tcp open Microsoft-ds
5001/tcp open commplex-link
8080/tcp open http-proxy
8081/tcp open blackice-icecap
MAC Address: 00:0C:29:51:79:D9 (VMware)
Nmap scan report for 192.168.171.165
Host is up (0.00070s latency).
Not shown: 991 closed ports
```

PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open Microsoft-ds
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp open unknown
49156/tcp open unknown
49157/tcp open unknown
MAC Address: 00:0C:29:EC:45:1E (VMware)
Nmap scan report for 192.168.171.169
Host is up (0.00081s latency).
Not shown: 994 closed ports
PORT STATE SERVICE
81/tcp open hosts2-ns
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open Microsoft-ds
2049/tcp open nfs
8080/tcp open http-proxy
MAC Address: 00:0C:29:46:71:3A (VMware)
Nmap scan report for 192.168.171.254
Host is up (0.00029s latency).
All 1000 scanned ports on 192.168.171.254 are filtered
MAC Address: 00:50:56:E3:E7:9E (VMware)
Nmap scan report for 192.168.171.166
Host is up (0.000013s latency).
Not shown: 999 closed ports
PORT STATE SERVICE
111/tcp open rpcbind
Nmap done: 256 IP addresses (7 hosts up) scanned in 22.44 seconds

3. Ahora vamos a intentar obtener el tipo de sistema operativo que están usando esas máquinas, máquina windows7 ….

┌──(apogon㉿kali-1)-[~]
└─$ nmap -O 192.168.171.165
Starting Nmap 7.80 (https://nmap.org) at 2019-09-15 14:54 EDT
Nmap scan report for 192.168.171.165
Host is up (0.00037s latency).
Not shown: 991 closed ports
PORT STATE SERVICE
135/tcp open msrpc

139/tcp open netbios-ssn
445/tcp open Microsoft-ds
49152/tcp open unknown
49153/tcp open unknown
49154/tcp open unknown
49155/tcp open unknown
49156/tcp open unknown
49157/tcp open unknown
MAC Address: 00:0C:29:EC:45:1E (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1
cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.94 seconds

4. Con este comando vamos a ver qué servicios y versión están ejecutándose en el objetivo maquina windows7

┌──(apogon㉿kali-1)-[~]
└─$ nmap -sV 192.168.171.165
Starting Nmap 7.80 (https://nmap.org) at 2019-09-15 14:51 EDT
Nmap scan report for 192.168.171.165
Host is up (0.00048s latency).
Not shown: 991 closed ports
PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open Microsoft-ds Microsoft Windows 7 - 10 Microsoft-ds
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
49156/tcp open msrpc Microsoft Windows RPC
49157/tcp open msrpc Microsoft Windows RPC
MAC Address: 00:0C:29:EC:45:1E (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 60.62 seconds

5. con el comando -A: Habilita la detección del sistema operativo, la detección de versiones, el escaneo de scripts y el trazado de ruta /máquina windows7 esta opción es muy intrusiva y puede ser detectada por ids o administrador.

```
┌──(apogon㉿kali-1)-[~]
└─$ nmap -A 192.168.171.165
Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-15 15:47 EDT
Nmap scan report for 192.168.171.165
Host is up (0.00083s latency).
Not shown: 991 closed ports
PORT STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds Windows 7 Home Basic 7600 microsoft-ds
49152/tcp open msrpc Microsoft Windows RPC
49153/tcp open msrpc Microsoft Windows RPC
49154/tcp open msrpc Microsoft Windows RPC
49155/tcp open msrpc Microsoft Windows RPC
49156/tcp open msrpc Microsoft Windows RPC
49157/tcp open msrpc Microsoft Windows RPC
MAC Address: 00:0C:29:EC:45:1E (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1
cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008
R2, Windows 8, or Windows 8.1 Update 1
└─$ nmap 192.168.171.165 > nmap.txt
```

y este otro comando lo utilizamos con los parámetros que nos ofrece nmap.

```
┌──(apogon㉿kali-1)-[~]
└─$ nmap 192.168.171.165 -oA nmapNetwork Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
Host script results:
|_clock-skew: mean: 59m59s, deviation: 1h43m55s, median: 0s
|_nbstat: NetBIOS name: nil, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:ec:45:1e
(VMware)
| smb-os-discovery:
| OS: Windows 7 Home Basic 7600 (Windows 7 Home Basic 6.1)
```

| OS CPE: cpe:/o:microsoft:windows_7::-
| Computer name: computacion 2019
| NetBIOS computer name:
| Workgroup: GRU-COMPUTAXION\xE5\x89\x87COMPUTAXION2019\x00
|_ System time: 2019-09-15T16:48:31-03:00
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:
| date: 2019-09-15T19:48:32
|_ start_date: 2019-09-15T18:40:26
TRACEROUTE
HOP RTT ADDRESS
1 0.83 ms 192.168.171.165
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.51 seconds

Con nmap tenemos algunas opciones para guardar el escaneo, el siguiente comando va a guardar el contenido en formato TXT

┌──(apogon㉿kali-1)-[~]

5. Técnicas criptográficas

Algoritmos
• Cifrado simétrico.
• Cifrado asimétrico.
• Funciones de hash

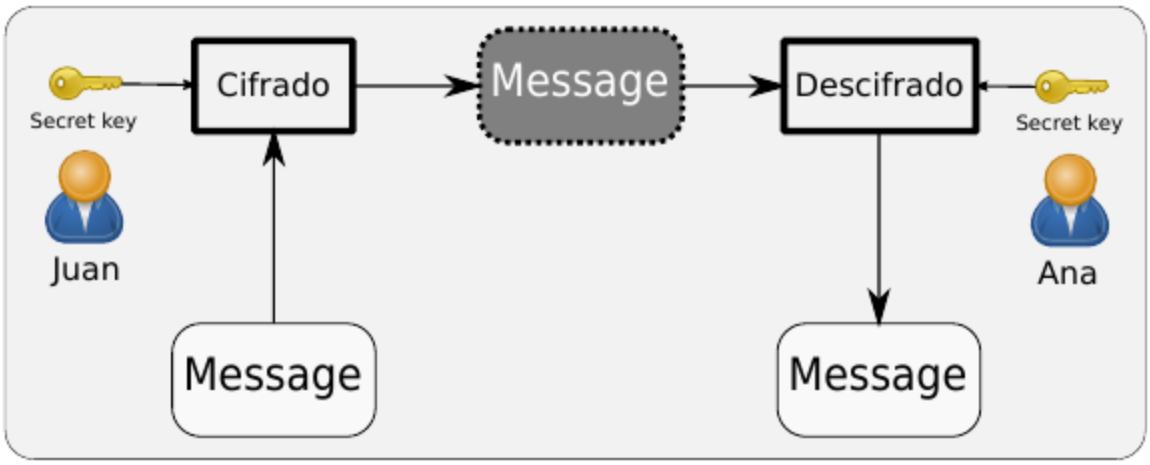


Figura 1.1: La misma clave se utiliza para cifrar y descifrar.
• Generación de números aleatorios.
Servicios
Autenticidad: Poder demostrar la autoría de un mensaje.
Confidencialidad: Sólo los autorizados pueden leer el mensaje.
Integridad: Asegurar la no modificación del mensaje.
No repudio: No poder negar la autoría de un mensaje.

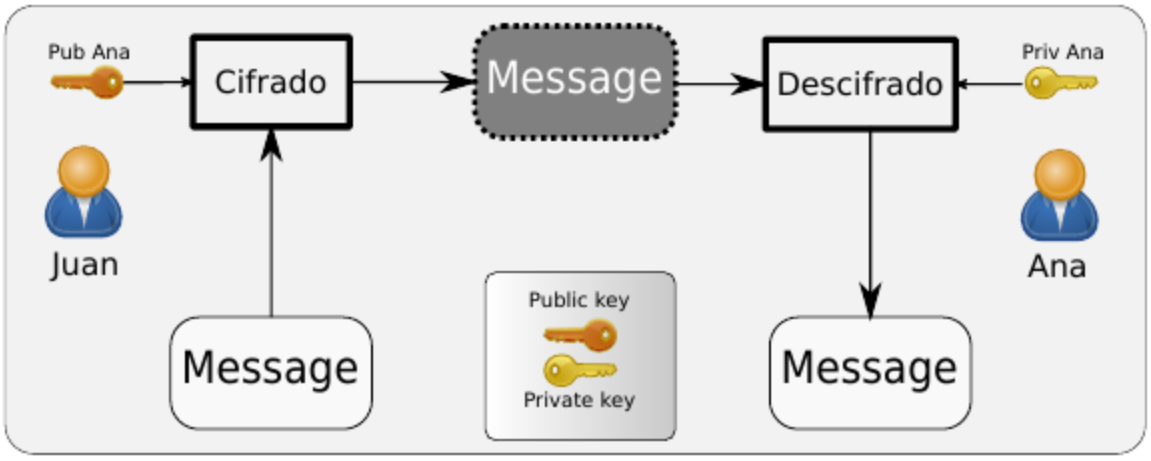
5.1 Criptografía de clave secreta

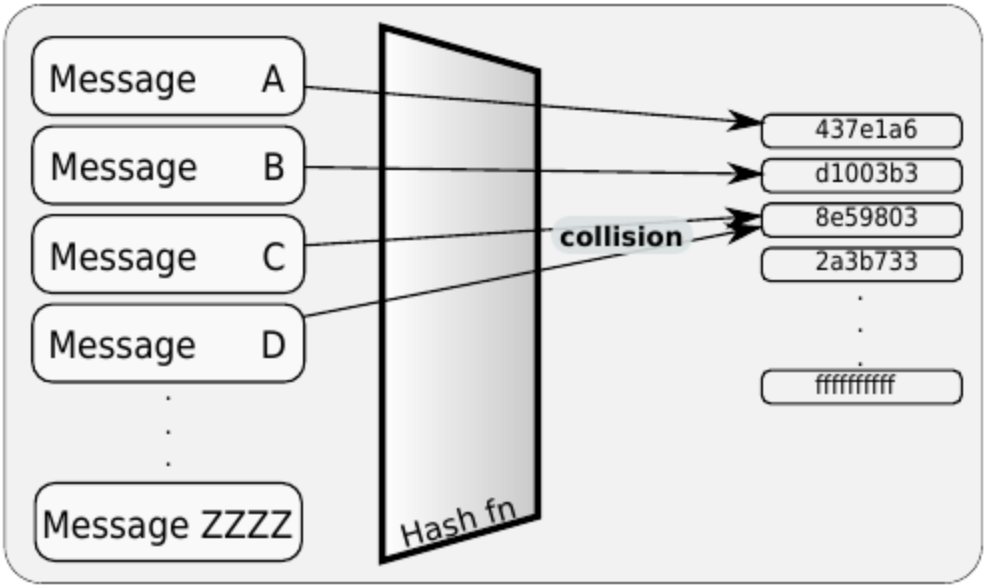
• Conocida como criptografía de clave secreta.
• Por regla general los algoritmos de cifrado y descifrado son públicos. La seguridad reside únicamente
en el secreto de las claves.
Más información en la asignatura de Criptografía.
• Los datos ((plaintext) se cifran con una clave que ha de ser conocida sólo por el emisor y el/los
receptores para garantizar la seguridad del sistema.
• El emisor y el receptor han de conocer la clave de cifrado.

• Es necesario que antes de enviar un mensaje ambos (emisor y receptor) conozcan la clave utilizando
para ello un medio de comunicación seguro.
• Por regla general suelen ser más rápidos que los algoritmos de clave pública.
• Algoritmos de clave secreta: IDEA, ARCFOUR, BLOWFISH, AES

5.2 Criptografía de clave pública

• Conocida también como criptografía asimétrica. Inventada en 1975.
• Cada usuario tiene dos claves. Una de ellas es la clave privada que sólo ha de conocer el propietario de la clave, y la otra la clave pública que ha de ser conocida por el mayor número de personas posible.
• A diferencia de la clave secreta, la clave privada sólo tiene que ser conocida por una sola persona
y no se tiene que compartir con nadie.
• Los datos cifrados con una clave sólo se pueden descifrar con la otra.



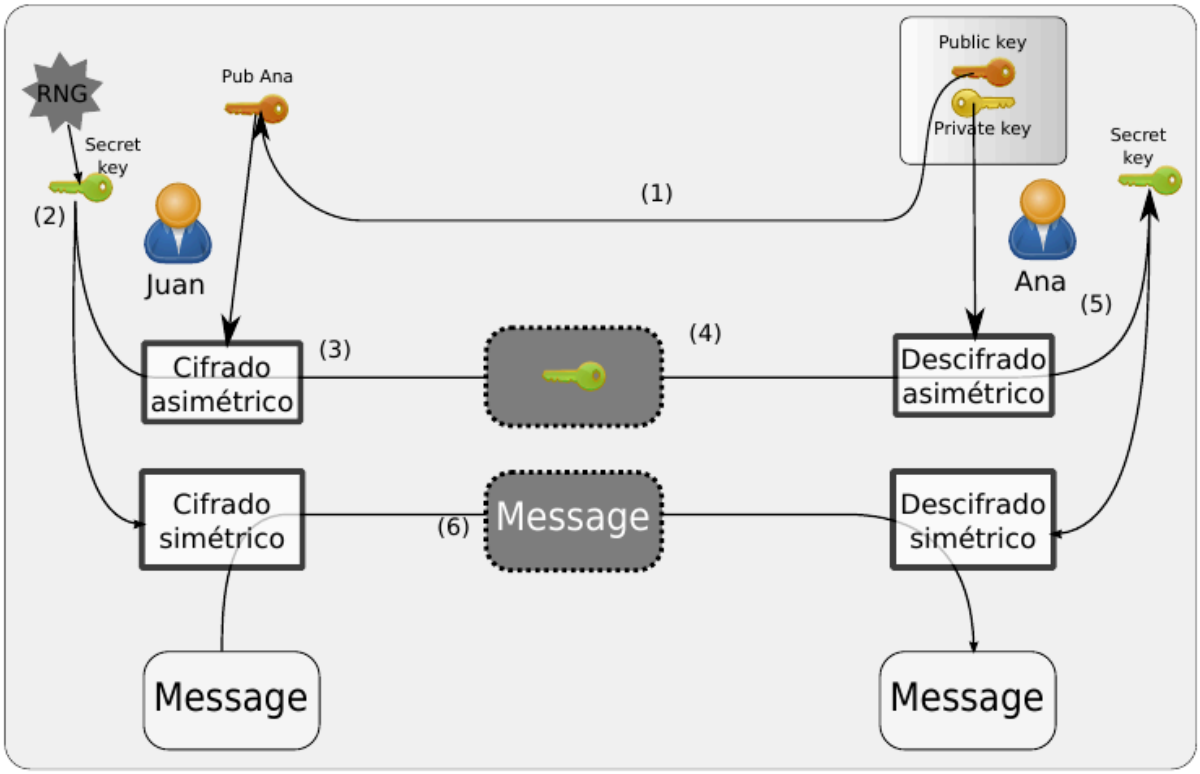


• Son algoritmos sencillos pero de alto coste computacional.
• Los principales algoritmos existentes son: DSA, RSA, ECC.
Cuantas más personas conozcan la clave pública, más seguro será el sistema

5.3 Funciones de hash

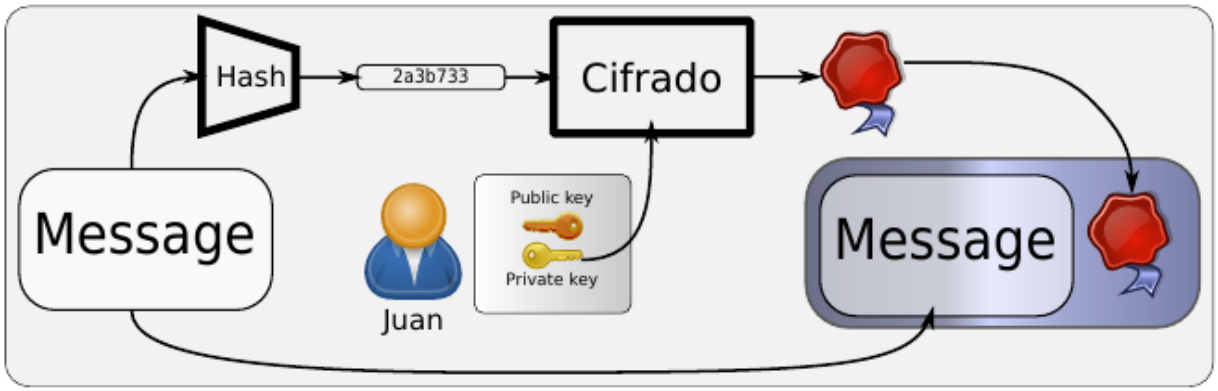
• En inglés se conocen como message digest (MD), o como message hash algorithm (SHA).
• Los algoritmos de hash, producen una salida que es mucho más reducida que el mensaje original y no se puede reconstruir el mensaje original a partir del resumen. Entre 128 y 512 bytes.
• Por contra, los algoritmos de cifrado toman como entrada una serie de datos y producen como salida otros datos a partir de los cuales se puede reconstruir el original. El tamaño del mensaje cifrado similar al original.
• Cada mensaje debería de producir un resumen distinto.

• Funciones de hash: SHA-224 SHA-512, MD5, ...
• Matemáticamente es imposible crear funciones de hash perfectas!
• Hay mayor número de mensajes que de resúmenes.

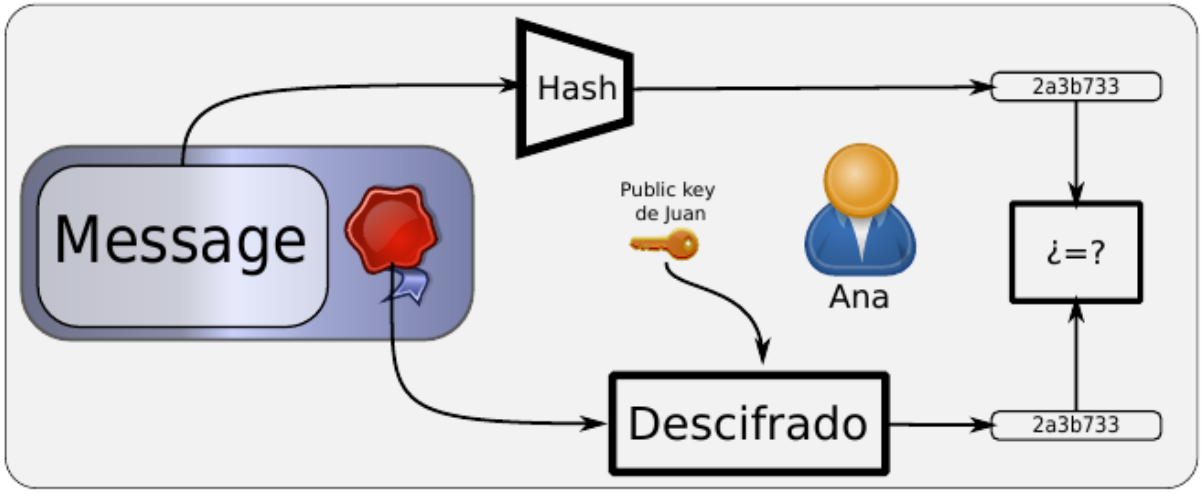


5.4 Firma digital

• Se dice firma digital a un esquema matemático que sirve para demostrar la autenticidad de un mensaje digital [WP].
• Una firma digital garantiza al destinatario que el mensaje fue creado por el remitente.
• Pasos para firmar un mensaje:
1. Calcular un resumen (MD5, SHA, etc.) del mensaje.
2. Cifrar el resumen resultante con la clave privada del remitente.
3. Adjuntar el resultado al mensaje original.



• Pasos para verificar una firma digital:
1. Extraer mensaje original.
2. Calcular un resumen (MD5, SHA, etc.) del mensaje original.
3. Extraer la firma digital.
4. Descifrar la firma digital con la clave pública de Juan.
5. Si coinciden los resultados entonces es correcta la firma.



5.5 La suite SSH (secure SHell)

5.5.1 Introducción

• ssh es un programa para realizar conexiones seguras entre máquinas conectadas mediante una red de comunicaciones insegura.
• Implementa varios algoritmos de encriptación y autentificación. Para el establecimiento de la conexión utiliza encriptación asimétrica. Para la transferencia de datos utiliza encriptación simétrica (más rápida).
• Diseñado para reemplazar a los antiguos programas de comunicaciones como: telnet, rlogin, rsh, rcp, rdist, etc., conocidas como “los programas que empiezan por R”, que tienen graves problemas de seguridad.

• Puede encriptar toda la información que el protocolo X0 transmite entre máquinas; permite realizar transferencias de archivos; para instalarlo no es necesario modificar ningún archivo del sistema.
• Algunos gobiernos creen que es demasiado bueno: Francia, Rusia, Iran, etc. Por lo que está prohibido su uso en estos países. Estados Unidos prohíbe exportarlo. Por tanto, no ningún ciudadano de USA puede trabajar en el código del ssh: Dug Song

5.5.2 Utilización básica de ssh

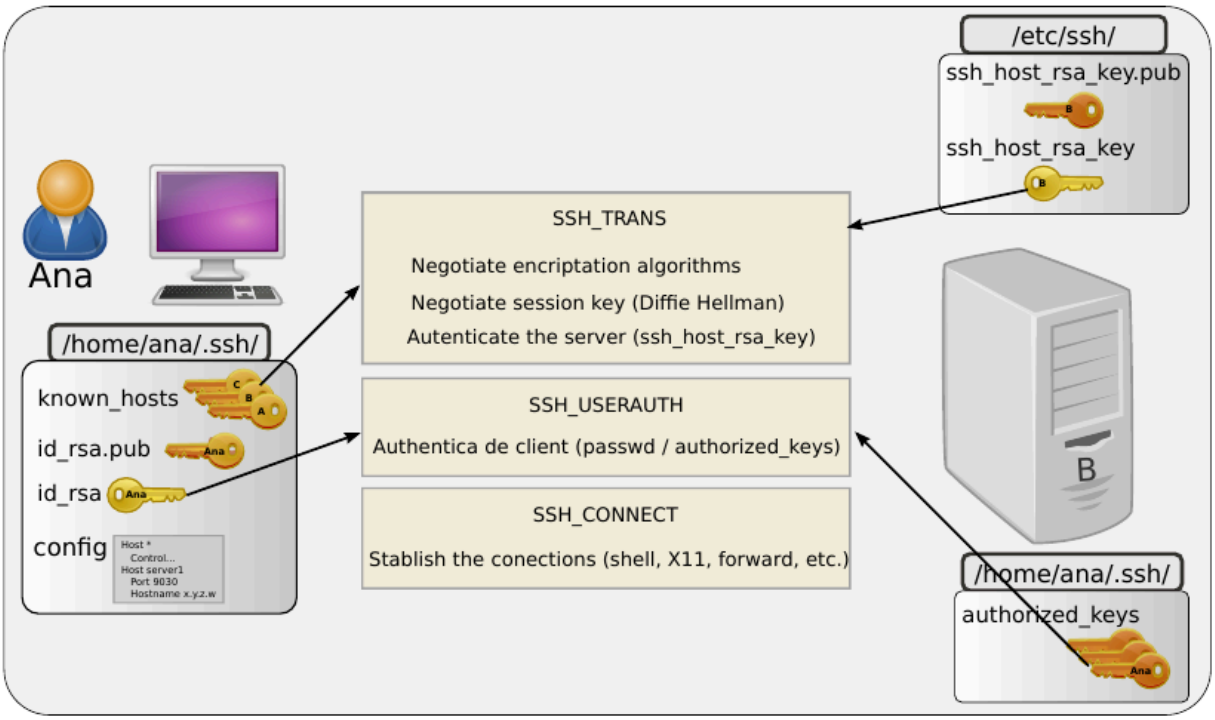
Suponiendo que todo el paquete de programas ssh está instalado tanto en nuestra propia máquina, como en la máquina remota. La forma de utilizar ssh para conectarse a una máquina remota es:
$ ssh maquina_remota
• La primera vez se creará en la máquina local el directorio $HOME/.ssh y dentro de este directorio el archivo known hosts. Éste archivo contendrá la clave pública de la máquina remota a la que nos acabamos de conectar. Cada máquina que ofrezca el servicio ssh tendrá una clave pública y otra privada.
• Ssh nos pedirá el password y si lo conocemos entonces se ejecutará el shell de conexión normal, pero donde todo lo que tecleemos y lo que nos devuelva por pantalla se enviará encriptado.
• Si la conexión la realizamos desde un entorno gráfico en la máquina local (esto es, si la variable de entorno DISPLAY está definida) entonces ssh redireccionará automáticamente todas las aplicaciones gráficas que lancemos en la máquina remota para que se visualicen en la máquina que estamos conectados (local).

• Tanto el establecimiento de la conexión como todos los datos que se transmitan se hacen por canal seguro.

5.5.3 Claves de usuario

• Con el esquema básico de autenticación, la identidad de un usuario se decide utilizando el sistema de seguridad UNIX de la máquina remota, esto es, en función del archivo /etc/passwd, /etc/shadow. Nuestro password viaja hasta el servidor donde el propio servidor lo valida.
• Podemos utilizar el sistema de clave pública de ssh para crear una pareja de claves (una privada y otra pública) que sustituyan a la clave UNIX. Esto sólo se ha de hacer una vez. Es algo así como poner el password a la propia cuenta.
• Pasos a seguir para que el usuario “kali” de la máquina “local” pueda entrar en la cuenta del usuario “prueba” de la máquina “remota”.
1. Con la orden ssh-keygen generamos las dos claves (pública y privada). Cada una de ellas se guarda en un archivo distinto. Ejemplo de creación de claves del usuario kali:
$ ssh-keygen -t dsa
Es conveniente introducir una passphrase para proteger el archivo de clave privada. De esta forma, aunque un atacante obtenga nuestro archivo de clave privada no podrá utilizarlo.
¿Passphrase?

La clave ssh se llama passphrase en lugar de password para diferenciarla de la que aparece en el archivo /etc/shadow, y para “animarnos” a introducir una cadena larga.
2. Luego tenemos que copiar, sin preocuparnos que la capturen (aunque sí que la modifiquen!), nuestra clave pública al archivo $HOME/.ssh/authorized keys de la máquina remota:
$ rcp id_dsa.pub remota:.ssh/authorized_keys
claves públicas, todas ellas concatenadas.
3. Asegurarnos que los permisos del archivo authorized keys son correctos:
$ ssh remota
$ chmod 600 .ssh/authorized_keys
$ chmod 700 .ssh
• Cuando se utiliza este esquema de autenticación: La passphrase no circula por la red.



5.5.4 Copia segura con scp

• Al igual que rcp permite realizar copias de archivos entre distintas máquinas, scp hace lo mismo pero de forma segura.
• scp acepta los mismo parámetros que rcp, pero utiliza un canal seguro creado por ssh para realizar la transferencia de datos.
• scp sigue los mismos esquemas de comprobación que ssh. Por tanto la forma de funcionar dependerá de cómo cada usuario tenga configurado ssh (tenga clave propia o no y si la clave tiene passphrase o no).

5.5.5 Archivos de Configuración

• /etc/ssh/sshd config: Configuración del demonio sshd. Se puede establecer varios parámetros de seguridad (Ver sshd(8)).
/etc/ssh/ssh config $HOME/.ssh/configy : Configuración del cliente.

• /etc/ssh/ssh host key : Contiene la clave privada de la máquina. Sólo root ha de poder leerla.
• /etc/ssh/ssh host key.pub: Parte pública de la clave. Permiso de lectura universal.
• /etc/ssh/ssh known hosts y $HOME/.ssh/known hosts: Contienen la clave pública de máquinas remotas y se utiliza para comprobar que la máquina remota es realmente quien dice ser.

Tarea y reporte: Generar un reporte del uso de ssh cómo sistema de conexión segura entre máquinas:

Secure Shell es una aplicación que mediante el protocolo SSH en el puerto 22 realiza la conexión entre un equipo servidor y sus clientes para proporcionar el servicio de terminal segura, con el cual faculta al cliente de realizar tareas administrativas mediante un usuario previamente definido en nuestro servidor, al ser un servicio remoto estamos facultando el control de acceso por tal motivo es importante definir el nivel de usuario deseado para cada nivel de acc. SSH es uno de los protocolos de comunicación más populares de Internet, ya que nos proporciona la capacidad de acceder de forma segura a sistemas locales y remotos utilizando un canal de comunicación encriptado. Además de ser Open-Source, es ampliamente usado por personal de TI (por ejemplo, Desarrolladores, Webmasters, Administradores de Sistemas, Pentesters, etc). Pasos para habilitar el soporte ssh en cliente y servidor:

Crear uno o varios usuarios para acceder a terminal en el SO, pueden utilizarse usuarios ya existentes, Abrir terminal y ejecutar el comando de instalación en servidor, como superusuario:

#apt-get install openssh-server openssh-client
#service ssh start
Una vez instalado, abrir el archivo del manual de open ssh:
man sshd_config
Con nmap revise los puertos abiertos e identifique el correspondiente al ssh ?
El archivo de configuración de ssh se encuentra en:
/etc/ssh/sshd_config
IMPORTANTE: Antes de cambiar el archivo de configuración, hacer una copia del archivo original y protegerlo contra escritura; para en caso de algún suceso, poder recuperarlo y restablecerlo.
Copiar el archivo /etc/ssh/sshd_config y protegerlo:
#sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
#sudo chmod a-w /etc/ssh/sshd_config.original
Una vez respaldado y protegido el archivo podremos editar el archivo inicial con nano en el entendido de que si se daña o desconfigura podremos reemplazarlo por el que acabamos de crear, una vez abierto vamos a analizarlo línea por línea e identificar la que contiene:

#what portss, IP’s and protocols we listen for
port 22
Aquí ubicaremos entonces la lista de direcciones IP, puertos y protocolos que serán escuchados (Listen) , asi como los archivos Key que contienen la identificación electrónica de los equipos que tienen acceso esto para en un futuro poder implementar VPN’s

1. Cambiar el puerto de escucha en 2222, reiniciar y verificar luego con nmap ?
2. Configura el servidor SSH de forma adecuada para que acepte la redirección X11, de tal forma que se puedan ejecutar aplicaciones gráficas de forma remota. Haz pruebas y comprueba su funcionamiento. Mediante ssh existe la posibilidad de ejecutar aplicaciones gráficas en el equipo remoto (servidor) y manejarlas y visualizarlas en el local (cliente). El servidor ssh deberá tener activada la redirección del protocolo X, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/sshd_config:
// Habilitar la redirección X en /etc/ssh/sshd_config
X11Forwarding yes
// Ejecutar aplicaciones gráficas

Vamos a simular que nuestra máquina no tiene acceso a internet, pero otra en el laboratorio si lo tiene y queremos navegar con firefox:
$ ssh -X usuarioPCremoto@IP_del_equiporemoto
Nota: -X para redirigir Xwindows.
$ firefox // Ejecutamos el Firefox

3. **Realizar una copia remota de archivos entre las dos máquinas (usando *scp*).**
Hacer capturas de pantalla

6. COMPLEMENTOS

6.1 PRÁCTICA DE OBTENCIÓN DE SUBDOMINIOS CON PYTHON

Inicialmente se debe crear un ambiente virtual de python con los siguientes pasos
1. crear un ambiente virtual de python en una carpeta (por ejemplo seguridad)
2. Activar el ambiente
Una vez ya en en el ambiente virtual de Python se debe instalar las librerías dnspython, requests y argparse:
pip install dnspython
pip install requests
En el archivo subdominios.txt se encuentra un listado grande de posibles subdominios que usualmente se configuran al interior de un dominio. El objetivo es hacer un “testing”de que subdominios se pueden encontrar detrás de un dominio dado. En esta primera fase recuerden que la idea es gestionar información que pueda ser útil para posteriormente analizar vulnerabilidades e incluso hacer algún ataque. También es importante recordar que vulnerar una máquina es un proceso muy complejo y a veces casi imposible, pero, si hay vulnerabilidades puede ser muy fácil hacer un hackeo, por lo tanto, una de las fases importantes es obtener información que pueda ser útil para hacer una explotación de las vulnerabilidades. Ejecutar el programa por ejemplo con el dominio wikipedia.com.

6.2 BANNER GRABBING

Uno de los aspectos a la hora de realizar controles sobre una aplicación web es la información que puede obtenerse a través de lo que se conoce como banner grabbing. Este concepto se refiere a la interacción manual en texto plano para obtener información sobre sobre el servidor donde reside la aplicación web, de esta manera se tiene datos sensibles muchas veces (Software que está en producción y posibles vulnerabilidades) que permiten un proceso de ataque web. Los banners exponen el nombre del servidor web, la versión que está corriendo y algunos exponen incluso el backend usado. La forma más simple de obtener el banner de un servidor es utilizando un socket, es decir, se envía una petición get y obtener la respuesta a través del método recv() que devolverá unos valores con el resultado.

6.3 PRÁCTICA DE BANNER GRABBING

Basándose en el programa anterior (subdomino.py) implementar un programa banner.py que reciba por parámetro el target (una dirección IP) y el puerto y que luego llame la función banner(ip,puerto). Que se vea así:def main():

```
# Entrar la IP
# Entrar el puerto
banner(ip,port)
```

6.4 theHarvester

TheHarvester, fue creado con la finalidad de usarse en la fase más importante en el paso de un hackeo, esta fase es la de recolectar información, y theharvester tiene la fortaleza de obtener información muy relevante como por ejemplo email, subdominios,URL,ip, entre otros utilizando varias fuentes públicas. Con esta herramienta podrán seguir reforzando su diccionario de palabras recolectada anteriormente.

Nota importante: theharvester es muy poderoso pero lamentablemente es muy probable que después de que realicemos nuestro primer análisis, posiblemente GOOGLE bloquee su ip de manera temporal y tendrán que esperar al menos 2 o 3 horas, para realizar nuevamente otro análisis

Recomendación: pueden buscar algún dominio de empresas grandes como universidades, hospitales, escuelas entre otros para obtener buenos resultados y comprender el funcionamiento.

Uso: las opciones del recolector

-d: dominio para buscar o nombre de la empresa

-b: fuente de datos: baidu, bing, bingapi, censys, crtsh, dogpile, google, google-certificados, googleCSE, googleplus, google-profiles, cazador, linkedin, netcraft, pgp, grupo de amenaza, twitter, vhost, virustotal, yahoo, todos

-g: usa Google Dorking en lugar de la búsqueda normal de Google

-s: comienza en el número de resultado X (predeterminado: 0)

-v: verifica el nombre del host a través de la resolución DNS y busca hosts virtuales -f:

guarda los resultados en un archivo HTML y XML (ambos)

-n: realiza una consulta inversa DNS en todos los rangos descubiertos

-c: realiza una fuerza bruta de DNS para el nombre de dominio

-t: realizar un descubrimiento de expansión de DNS TLD

-e: usa este servidor DNS

-p: el puerto escanea los hosts detectados y comprueba las adquisiciones (80,443,22,21,8080)

-l: limita el número de resultados con los que trabajar (Bing va de 50 50 a resultados, Google 100 a 100, y PGP no usa esta opción)

-h: utiliza la base de datos SHODAN para consultar hosts descubiertos

Ejemplos:

theharvester -d microsoft.com -l 500 -b google -f myresults.html

theharvester -d microsoft.com -b pgp, virustotal

theharvester -d microsoft -l 200 -b linkedin

theharvester -d microsoft.com -l 200 -g -b google

theharvester -d apple.com -b googleCSE -l 500 -s 300

theharvester -d cornell.edu -l 100 -b bing -h

Para invocar esta herramienta debemos teclear en una consola el siguiente comando:

```
┌──(apogon㉿kali-1)-[~/theHarvester]
└─$ python3 theHarvester.py -d microsoft.com -l 500 -b google -f myresults.html
```

Este comando aparte de traernos la herramienta en la consola, nos muestra la ayuda, lo cual se darán cuenta que es muy intuitiva y fácil de comprender, y también nos muestra varios posibles comandos a ejecutar. Bien, ahora es importante aclararles que esta herramienta buscará en el objetivo analizar TODA LA INFORMACIÓN PÚBLICA QUE PUEDA ENCONTRARSE DENTRO DE INTERNET. Por lo tanto no se constituye como algo ilegal, o un delito informático, cabe destacar que se puede encontrar información de terceros dentro de un análisis por ejemplo email o subdominios ajenos al objetivo, pero contratados por los mismos, con esto me refiero a que pueden

encontrar información de empresas que brindan servicios para la misma, como por ejemplo algún antivirus, o software de microsoft etc, ese tipo de información ustedes las tienen que obviar por que no se los contrató para realizar un análisis a estos servicios, aparte seria perder dinero en tiempo.

Haga la siguiente consulta:

```
┌──(apogon㉿kali-1)-[~/theHarvester]
└─$ python3 theHarvester.py -d ucol.mx -l 500 -b google
```

3.10 CeWL

https://github.com/digininja/CeWL

CeWL Es una herramienta que viene incorporada dentro de kali linux, esta herramienta se caracteriza, por recolectar palabras, números , email entre otras cosas, es una herramienta simple pero muy poderosa, teniendo en cuenta que a la hora de recolectar información que tenga referencia hacia nuestro objetivo es sumamente importante. Con esta herramienta vamos a poder crear nuestro propio diccionario enfocado directamente al objetivo, para posteriormente poder realizar ataques de fuerza bruta utilizando otra herramientas como apoyo, como puede ser john the ripper. El comando CEWL -h nos muestra la ayuda de esta herramienta:

```
┌──(apogon㉿kali-1)-[~]
└─$ cewl -h
```

CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

Usage: cewl [OPTIONS] ... <url>

OPTIONS:

-h, --help: Show help.

-k, --keep: Keep the downloaded file.

-d <x>,--depth <x>: Depth to spider to, default 2.

-m, --min_word_length: Minimum word length, default 3.

-o, --offsite: Let the spider visit other sites.

--exclude: A file containing a list of paths to exclude

--allowed: A regex pattern that path must match to be followed

-w, --write: Write the output to the file.

-u, --ua <agent>: User agent to send.

-n, --no-words: Don't output the wordlist.

-g <x>, --groups <x>: Return groups of words as well

--lowercase: Lowercase all parsed words

--with-numbers: Accept words with numbers in as well as just letters

--convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe, ü-ue, ß-ss)

-a, --meta: include meta data.

--meta_file file: Output file for meta data.

-e, --email: Include email addresses.

--email_file <file>: Output file for email addresses.

--meta-temp-dir <dir>: The temporary directory used by exiftool when parsing files, default /tmp.

-c, --count: Show the count for each word found.

-v, --verbose: Verbose.

--debug: Extra debug information.

Authentication

--auth_type: Digest or basic.

--auth_user: Authentication username.

--auth_pass: Authentication password.

Proxy Support

--proxy_host: Proxy host.

--proxy_port: Proxy port, default 8080.

--proxy_username: Username for proxy, if required.

--proxy_password: Password for proxy, if required.

Headers

--header, -H: In format name:value - can pass multiple.

<url>: The site to spider.

Referencias

Perfecto, aquí te dejo las **referencias en formato APA 7ª edición** para tu documento de Seguridad de la Información:

REFERENCIAS BIBLIOGRÁFICAS

Libros de Seguridad de la Información y Ciberseguridad

Barrett, D. J., Silverman, R. E., & Byrnes, R. G. (2005). *SSH, the secure shell: The definitive guide* (2nd ed.). O'Reilly Media.

Engebretson, P. (2013). *The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy* (2nd ed.). Syngress.

García-Cerecignon, M., & Alegre Ramos, M. (2010). *Seguridad informática*. Paraninfo.

Kaufman, C., Perlman, R., & Spencer, M. (2002). *Network security: Private communication in a public world* (2nd ed.). Prentice Hall.

Kim, K., & Solomon, S. (2016). *Fundamentals of information systems security* (3rd ed.). Jones & Bartlett Learning.

PSILON. (2011). *Seguridad informática - Ethical hacking*. ACISSI.

Stallings, W. (2017). *Network security essentials: Applications and standards* (6th ed.). Pearson Education.

Weidman, G. (2014). *Penetration testing: A hands-on introduction to hacking*. No Starch Press.

Weidman, G. (2017). *Advanced penetration testing: Hacking the world's most secure networks*. Wiley.

Whitman, M. E., & Mattord, H. J. (2018). *Principles of information security* (6th ed.). Cengage Learning.

Criptografía

Aguirre, J. R. (2006). *Libro electrónico de seguridad informática y criptografía*. Universidad Politécnica de Madrid. http://www.criptored.upm.es/guiateoria/gt_m001a.htm

Lucena López, M. J. (2007). *Criptografía y seguridad en computadores* (4ª ed.). http://www.criptored.upm.es/

National Institute of Standards and Technology. (2001). *Advanced encryption standard (AES)* (FIPS PUB 197). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

National Institute of Standards and Technology. (2015). *Secure hash standard (SHS)* (FIPS PUB 180-4). https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

Schneier, B. (1996). *Applied cryptography: Protocols, algorithms, and source code in C* (2nd ed.). John Wiley & Sons.

Stallings, W. (2020). *Cryptography and network security: Principles and practice* (8th ed.). Pearson Education.

Zimmermann, P. (1999). *An introduction to cryptography*. Network Associates. ftp://ftp.pgpi.org/pub/pgp/6.5/docs/english/IntroToCrypto.pdf

Hacking Ético y Metodologías de Pentesting

EC-Council. (2019). *Certified ethical hacker (CEH) official training guide* (11th ed.). Cengage Learning.

ISECOM. (2010). *Open source security testing methodology manual (OSSTMM)* (Version 3). https://www.isecom.org/OSSTMM.3.pdf

OWASP. (2020). *OWASP testing guide* (Version 4.2). https://owasp.org/www-project-web-security-testing-guide/

The Penetration Testing Execution Standard (PTES). (n.d.). *Penetration testing execution standard*. http://www.pentest-standard.org/

Herramientas y Tecnologías

Borges, C. (n.d.). *theHarvester - E-mails, subdomains and names harvester* [Software]. GitHub. https://github.com/laramies/theHarvester

DigiNinja. (n.d.). *CeWL - Custom word list generator* [Software]. GitHub. https://github.com/digininja/CeWL

Lyon, G. F. (2009). *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure.Com LLC.

Offensive Security. (2024). *Kali Linux documentation*. https://www.kali.org/docs/

THC-Hydra Development Team. (n.d.). *THC-Hydra* [Software]. GitHub. https://github.com/vanhauser-thc/thc-hydra

Wireshark Foundation. (2024). *Wireshark user's guide*. https://www.wireshark.org/docs/wsug_html_chunked/

Protocolos Seguros (SSH)

Eastlake, D. (2005). *Domain name system security extensions* (RFC 4033). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc4033

Mockapetris, P. (1987). *Domain names - Implementation and specification* (RFC 1035). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc1035

Ylonen, T., & Lonvick, C. (2006a). *The secure shell (SSH) authentication protocol* (RFC 4252). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc4252

Ylonen, T., & Lonvick, C. (2006b). *The secure shell (SSH) protocol architecture* (RFC 4251). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc4251

Normas y Estándares de Seguridad

International Organization for Standardization. (2013). *ISO/IEC 27001:2013 - Information technology - Security techniques - Information security management systems - Requirements*.

International Organization for Standardization. (2022). *ISO/IEC 27002:2022 - Information security, cybersecurity and privacy protection - Information security controls*.

National Institute of Standards and Technology. (2018). *Framework for improving critical infrastructure cybersecurity* (Version 1.1). https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

Amenazas y Ransomware

Bhardwaj, S. (2016). Ransomware: A rising threat of new age digital extortion. *International Journal of Computer Science and Mobile Computing, 5*(4), 796-803.

Kaspersky Lab. (2024). *What is ransomware? - Definition and explanation*. https://www.kaspersky.com/resource-center/definitions/what-is-ransomware

Python para Seguridad

O'Connor, T. J. (2012). *Violent Python: A cookbook for hackers, forensic analysts, penetration testers and security engineers*. Syngress.

Ortega, J. M. (2018). *Mastering Python for networking and security*. Packt Publishing.

Bases de Datos de Vulnerabilidades

Exploit Database. (n.d.). *Offensive Security's exploit database archive*. https://www.exploit-db.com/

MITRE Corporation. (n.d.). *Common vulnerabilities and exposures (CVE)*. https://cve.mitre.org/

Reportes y Recursos Públicos

Fort, J. C. (n.d.). *Public penetration testing reports* [Repository]. GitHub. https://github.com/juliocesarfort/public-pentesting-reports

Haverickadams. (n.d.). *TCM Security sample pentest report*. GitHub. https://github.com/hmaverickadams/TCM-Security-Sample-Pentest-Report

Pentest Reports. (n.d.). *Penetration testing report templates*. https://pentestreports.com/templates/

Google Hacking

Long, J. (2004). *Google hacking for penetration testers*. Syngress.

Offensive Security. (n.d.). *Google Hacking Database (GHDB)*. https://www.exploit-db.com/google-hacking-database

---