7. [Análisis Forense] - ¿Qué es Wireshark?

1. Lista todos los parámetros GET que aparecieron durante tu navegación.

   cat: `GET /listproducts.php?cat=1 HTTP/1.1`
   artist: `GET /artist.php?artist=4`
   pic: `GET /product.php?pic=3`
   add: `GET /shoppingcart.php?add=2`
   test: `GET /login.php?test=1`

2. ¿Qué peticiones POST detectaste? ¿Qué información enviaron?

   Petición POST: `POST /userinfo.php HTTP/1.1`
   Solicitud de envío del formulario de inicio de sesión enviado al servidor

   Información enviada: `uname=prueba&pass=1234`
   Enviar el nombre de usuario (uname) y la contraseña (pass)

3. ¿Cómo viajan las credenciales del formulario de login?

   Las credenciales (uname y pass) se transmiten en texto plano a través de HTTP sin cifrar. Sin medidas de seguridad adicionales, Wireshark puede capturar directamente los datos de los campos de formulario dentro de las solicitudes POST sin necesidad de descifrarlos.

4. ¿Qué cookies o identificadores de sesión aparecen durante la captura?

   Cookie principal: PHPSESSID
   `Cookie: PHPSESSID=8d6v93h3k8a...`
   Identificador de sesión de PHP

5. ¿Puedes reconstruir qué páginas visitó el usuario? Describe el flujo.

   1. GET /index.php HTTP/1.1: Visite la página de inicio del web
   2. GET /listproducts.php?cat=1 HTTP/1.1: Explore la lista de productos de la Categoría 1
   3. GET /artist.php?artist=4 HTTP/1.1: Ver la página del artista con el ID 4
   4. POST /userinfo.php HTTP/1.1: Envíe el formulario de inicio de sesión (introduzca el nombre de usuario y la contraseña)

5. GET /product.php?pic=3 HTTP/1.1 : Ver detalles del producto para el artículo ID 3

6. GET /shoppingcart.php?add=2 HTTP/1.1 : Añadir el artículo con el ID 2 a la cesta de la compra

6. ¿Has encontrado algún error HTTP (404, 500…)? ¿En qué rutas?

   GET /nonexist.php HTTP/1.1 → Respuesta HTTP/1.1 404 No encontrado;

   GET /images/invalid.jpg HTTP/1.1 → Respuesta HTTP/1.1 404 No encontrado (solicitando un recurso de imagen inexistente)

7. Explica qué riesgos de seguridad existen al usar HTTP en lugar de HTTPS.

   1. **Exposición de credenciales:** los datos confidenciales, como nombres de usuario y contraseñas, se transmiten en texto sin cifrar, lo que los hace vulnerables a la interceptación por parte de atacantes dentro de la red.

   2. **Secuestro de sesión:** el identificador PHPSESSID se transmite en texto sin cifrar, lo que permite a los atacantes suplantar a los usuarios tras la interceptación.

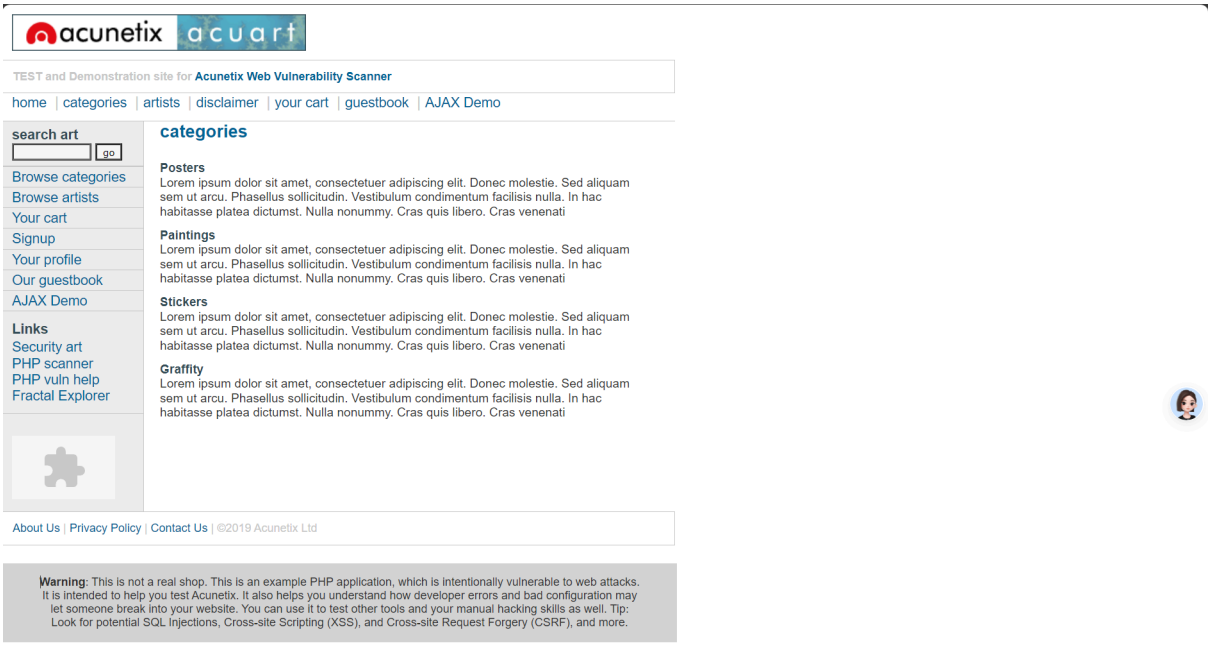
   3. **Manipulación de contenido:** los atacantes pueden modificar las solicitudes/respuestas HTTP (por ejemplo, alterar los precios de los productos o el contenido HTML de las páginas web) debido a la ausencia de mecanismos de verificación de integridad.

   4. **Fallo en la autenticación del servidor:** sin certificados TLS/SSL, los usuarios no pueden verificar que se están conectando al servidor legítimo testphp.vulnweb.com, lo que los expone a riesgos de phishing.

   5. **Exposición del patrón de navegación:** todo el tráfico (visitas a páginas, solicitudes de recursos) se transmite en texto plano, lo que permite a terceros supervisar el comportamiento de navegación de los usuarios y las rutas de acceso.

Práctica: Análisis de Tráfico HTTP con Wireshark en una Web Vulnerable

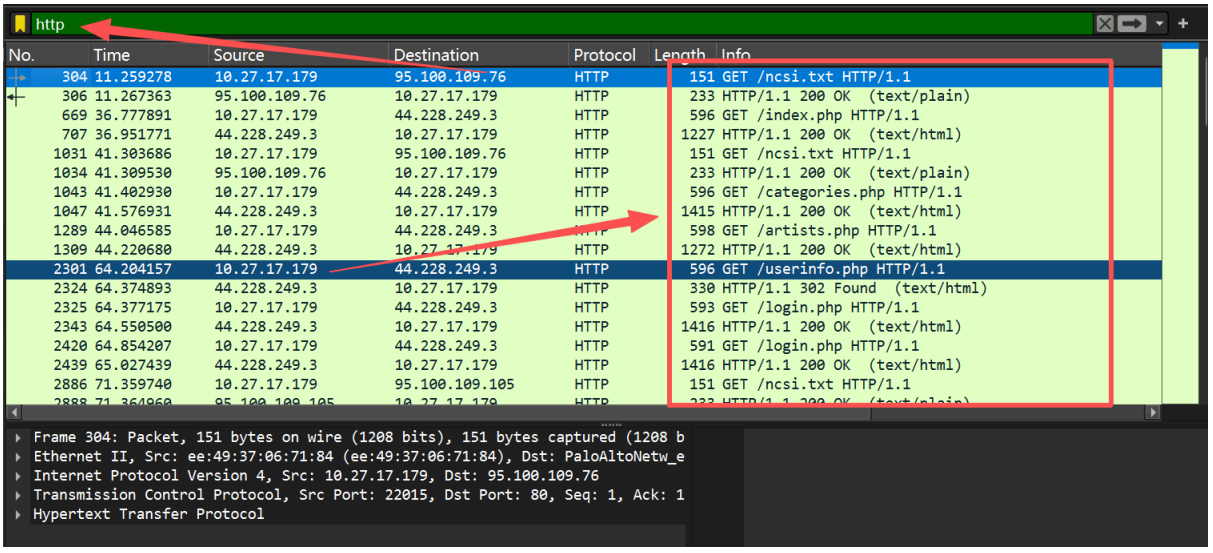
Comience a capturar paquetes con Wireshark y genere tráfico navegando por sitios web:
http://testphp.vulnweb.com/



Visualizar únicamente tráfico HTTP

Detenga la captura de paquetes y utilice filtros para aislar el tráfico HTTP.

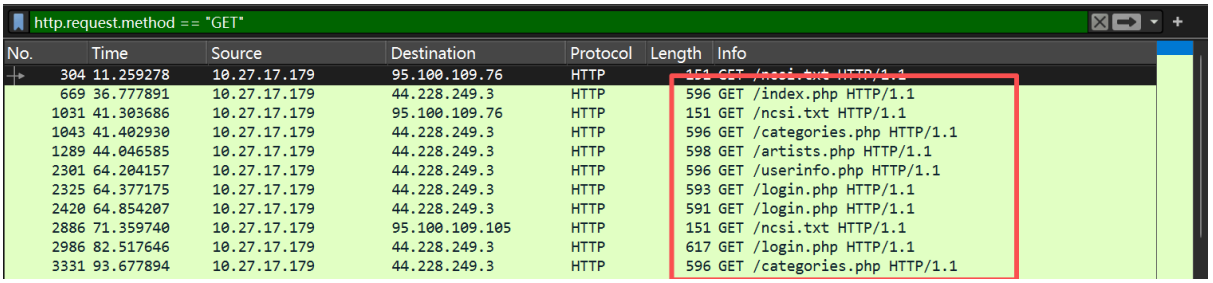
- http



Aquí podemos ver el tráfico generado por nuestra navegación web.

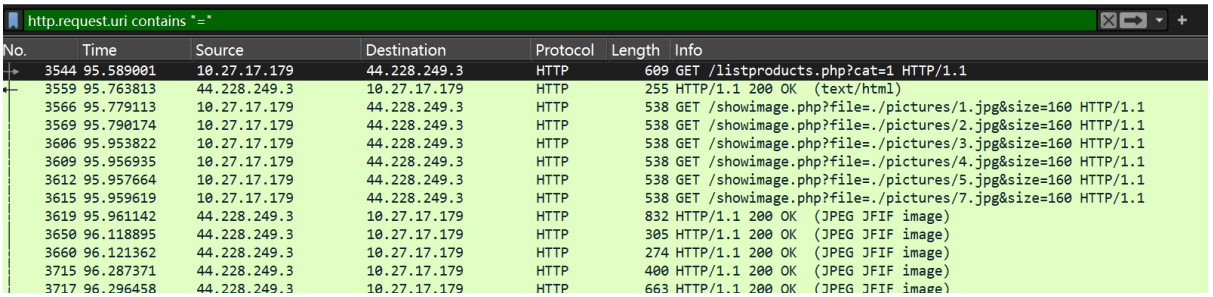
Listar únicamente peticiones GET

- http.request.method == "GET"



Detectar peticiones con parámetros

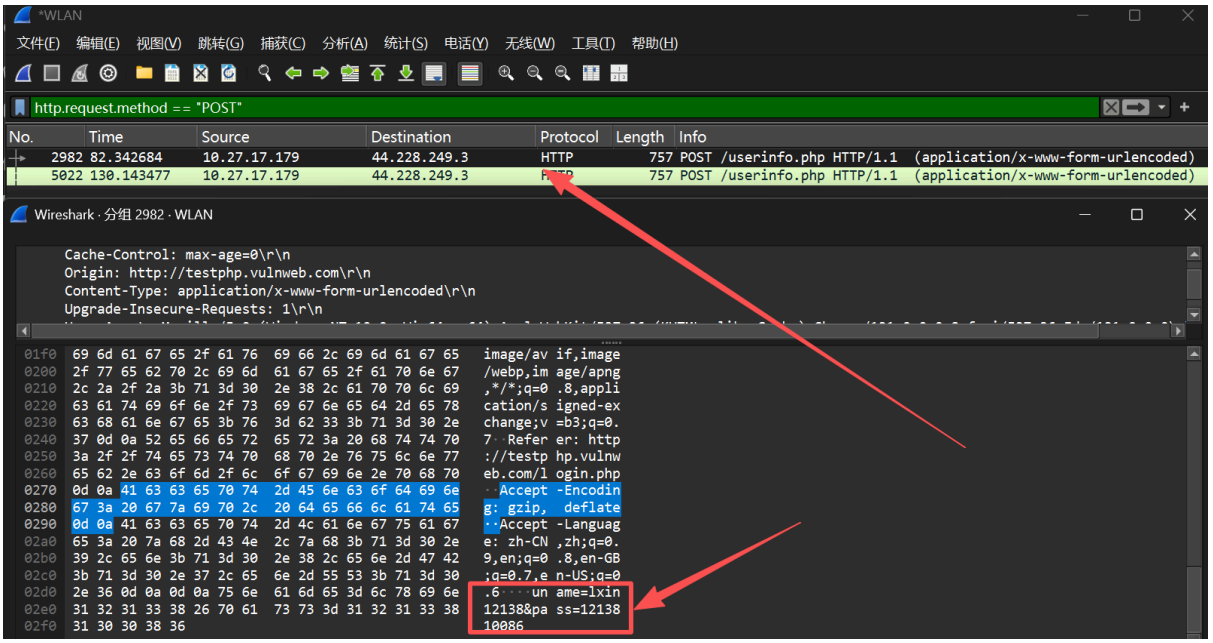
- http.request.uri contains "="



Analizar el login (peticiones POST)

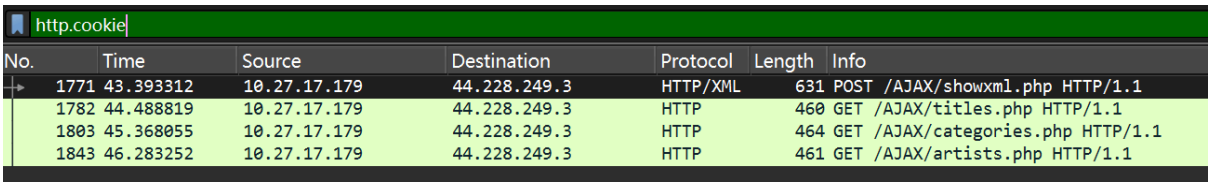
Introduzca cualquier nombre de usuario y contraseña en el formulario de inicio de sesión del sitio web y, a continuación, introduzca el siguiente comando utilizando un filtro en Wireshark:

- http.request.method == "POST"



Visualizar cookies y sesiones

- http.cookie



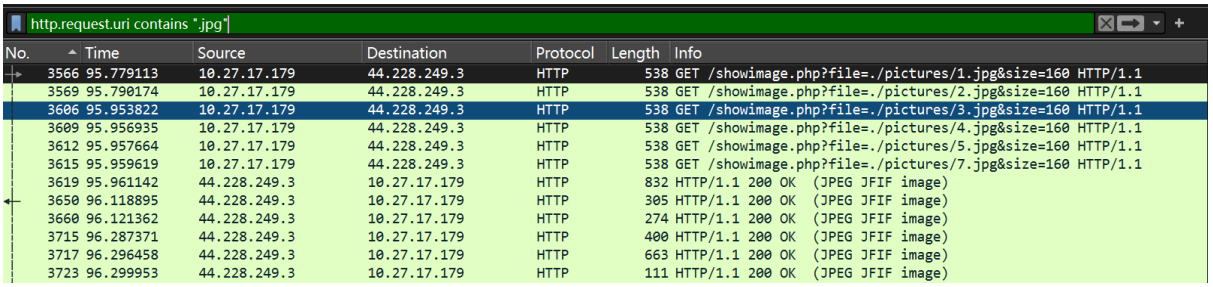
Encontrar errores en la web

- http.response.code >= 400

Localizar recursos concretos: imágenes, JS y CSS

Imágenes (JPG):

- http.request.uri contains ".jpg"



Scripts:

- http.request.uri contains ".js"

Hojas de estilo:

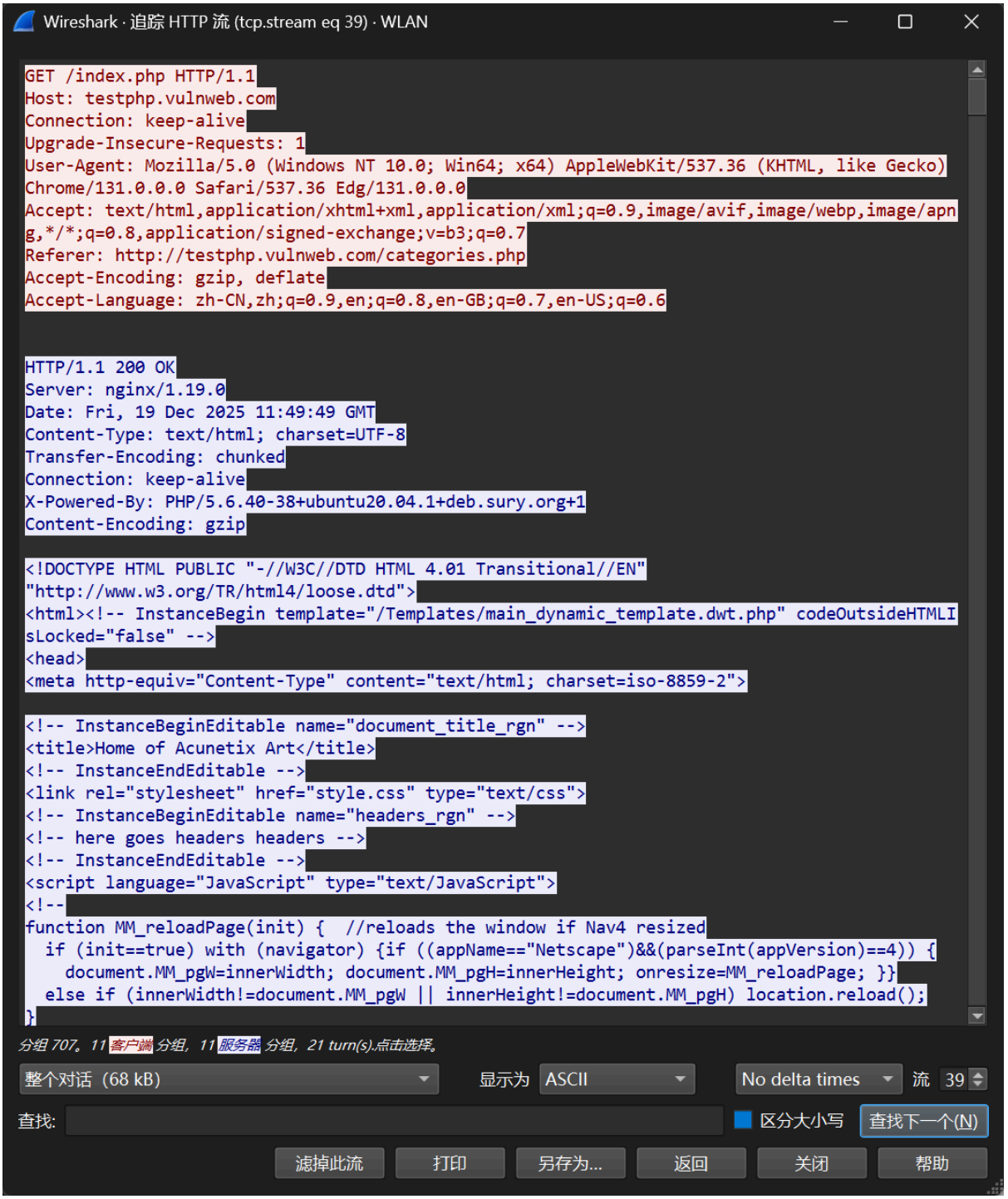
- http.request.uri contains ".css"

Seguir una conversación completa

Selecciona cualquier paquete HTTP → clic derecho →
Follow → HTTP Stream

Verás la conversación completa entre cliente y servidor:

- Petición completa
- Respuesta completa
- HTML enviado


Wireshark · 追踪 HTTP 流 (tcp.stream eq 39) · WLAN
GET /index.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/categories.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Fri, 19 Dec 2025 11:49:49 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Encoding: gzip
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMLI
sLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>Home of Acunetix Art</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
}
分组 707。11 客户端 分组，11 服务器 分组，21 turn(s).点击选择。
整个对话 (68 kB)
显示为 ASCII
No delta times
流 39
查找:
区分大小写
查找下一个(N)
滤掉此流
打印
另存为...
返回
关闭
帮助