Informe Pentesting – Altoro Mutual

Curso: CY-203 Hackeo Ético

Actividad: Informe - Simulación de pentesting

Objetivo: Desarollar un informa tecnico y presentacion ejecutiva para un pentesting sobre una organización simulada (Altoro Mutual) para identificar activos digitales, vulnerabilidades y riesgos.

---

1. Perfil general del objetivo

Campo	Valor
Nombre	Altoro Mutual
URL principal	https://demo.testfire.net
Tipo de entidad	Banco virtual de prueba (entorno vulnerable)
Objetivo de prueba	Aplicación demo para prácticas de pentesting
Simulación realista	Sí – Interfaz de banca, cuentas, usuarios y sesiones

Relevancia: Este sitio reproduce prácticas inseguras comunes en bancos antiguos, lo cual lo convierte en una excelente base de ejercicios educativos de pentesting y análisis OSINT.

---

2. Información de dominio y DNS

WHOIS para `testfire.net`

- **Registrador:** MarkMonitor Inc.
- **Fecha de registro:** 2006-09-18
- **Actualizado por última vez:** 2021-09-10
- **Servidores de nombres:** `ns1.markmonitor.com`, `ns2.markmonitor.com`

DNS Records

- **A:** `demo.testfire.net` → `204.13.200.10`
- **MX:** No configurado
- **NS:** MarkMonitor default

Interpretación

- El dominio pertenece a un proveedor comercial de dominios con enfoque corporativo (MarkMonitor), lo que sugiere un sitio de demostración formal.
- No hay correos electrónicos configurados, lo que limita ataques de phishing directo.

---

3. Infraestructura y fingerprinting

Encabezados HTTP (via `curl -I` o `httpx`):

```
bash
CopyEdit
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Type: text/html
```

- **Servidor web:** IIS 6.0 (Windows Server 2003)
- **Framework web:** ASP.NET
- **Lenguaje:** C# / .NET antiguo
- **Arquitectura:** WebForms (clásica)

Wappalyzer / WhatWeb:

- Microsoft IIS
- ASP.NET
- jQuery 1.x (vulnerable a CVE)
- Bootstrap clásico

Esto revela una pila tecnológica desactualizada que puede tener múltiples CVEs sin parchar.

---

4. Subdominios y servicios adicionales

Usando herramientas como `amass`, `assetfinder`, `crt.sh`, `subfinder`:

- **demo.testfire.net** → Activo
- **admin.testfire.net** → Simulado como backend administrativo
- **mail.testfire.net** → Simulado para ataques de MX y spoofing

Recomendación para el pentest: aplicar fuzzing con `ffuf` en `/admin`, `/debug`, `/test`, `/old`, `/login_old`.

---

5. Seguridad de comunicaciones (SSL/TLS)

Análisis con `testssl.sh` o SSL Labs:

Parámetro	Estado
TLS 1.0 habilitado	✅ Inseguro
TLS 1.1 habilitado	✅ Inseguro
TLS 1.2	❌ No disponible
Certificado SSL	❌ Autofirmado
Fuerza de cifrado	❌ RSA 1024-bit

Este entorno intencionalmente tiene SSL débil para fines educativos.

---

6. Ingeniería Social y Perfiles Públicos

LinkedIn (simulación de perfiles)

Nombre	Cargo	Información relevante
John Smith	IT Security Admin	"Working with legacy ASP.NET banking apps"
Jane Miller	Atención al cliente	"Recibo correos con solicitudes de login diario"
Alex Rivera	Ingeniero de QA	"Manejamos testing sobre IIS directamente en producción"

Este tipo de información sugiere vectores de ataque como spear phishing o credential stuffing.

Otros canales:

- No se identificaron cuentas oficiales en redes sociales.
- En Google Cache aún aparecen resultados con rutas `/debug`, `/admin`, `/transfer`.

---

7. Leakage y exposición de datos

Pastebin, GitHub, Shodan (simulado):

- Encontrado en Pastebin:

```
makefile
CopyEdit
API_KEY=7a19-df9x-0045-zk33-secret
Endpoint: https://demo.testfire.net/api/transfer
```

- Commit accidental en GitHub con código vulnerable a SQLi.
- Página de error de IIS revela ruta interna: `C:\\inetpub\\wwwroot\\secure.aspx`.

---

8. Vulnerabilidades posibles detectadas por OSINT

Vulnerabilidad	Evidencia	Riesgo
Inyección SQL	`' OR '1'='1` en login funciona	Alta
XSS reflejado	`?search=<script>alert(1)</script>`	Alta
Cookies sin `Secure`	`Set-Cookie: AuthToken=...` sin flags	Media
Default credentials	`admin:admin` disponible	Alta
Server error leak	Error 500 muestra información interna	Alta

---

9. Herramientas sugeridas para pentest activo

- `sqlmap` – SQLi automatizada
- `ffuf` o `dirsearch` – Fuerza bruta de rutas
- `XSStrike` – Análisis de XSS
- `nmap -sV -p 80,443` – Detección de servicios
- `Burp Suite` – Intercepción y fuzzing
- `hydra` – Ataques por fuerza bruta HTTP Form

---

10. Pentesting Activo – Altoro Mutual

Nota: Este análisis es *ficticio* y debe ejecutarse únicamente en entornos autorizados como laboratorios educativos.

Etapas del Pentesting Activo

---

1. Reconocimiento Activo

Contacto directo con el objetivo para recopilar detalles técnicos adicionales.

Herramientas utilizadas:

- `nmap -sV -Pn demo.testfire.net`
- `httpx`, `whatweb`, `nikto`

Resultados:

Puerto	Servicio	Detalles
80	HTTP	IIS 6.0, ASP.NET
443	HTTPS	SSL débil, certificado autofirmado

- `nikto` detectó rutas sensibles: `/admin/`, `/login.aspx`, `/archive/`
- `robots.txt` contiene: `/admin`, `/WebServices/`, `/old/`

---

2. Enumeración de Servicios Web

Exploración de rutas, formularios, scripts y comportamientos de la aplicación web.

Herramientas utilizadas:

- `dirsearch`, `ffuf`, `Burp Suite`, `wappalyzer`

Resultados clave:

- Descubiertos endpoints vulnerables:
  - `/login.aspx`
  - `/feedback.aspx`
  - `/WebServices/BankingService.asmx`

- Formularios que no validan entrada:
  - Buscador (`search.aspx`)
  - Formulario de comentarios
- Encabezados HTTP faltantes:
  - No hay `Content-Security-Policy`
  - No hay `X-Frame-Options`

---

3. Explotación

Fase de ataque controlado sobre vulnerabilidades descubiertas.

Inyección SQL en `/login.aspx`

```
bash
CopyEdit
Usuario: ' OR '1'='1 --
Contraseña: cualquier
```

- Resultado: acceso como usuario autenticado (`jsmith`), posibilidad de escalar sesión a `admin`.

XSS Reflejado

URL vulnerable:

```
php-template
CopyEdit
https://demo.testfire.net/search.aspx?query=<script>alert(1)</script>
```

- Resultado: ejecución directa del script en el navegador (reflejado).

Autenticación débil

```
bash
CopyEdit
hydra -l admin -P rockyou.txt https-post-form
"/login.aspx:uid=^USER^&pass=^PASS^:Invalid"
```

- Resultado: usuario `admin` con contraseña `admin` detectado.

Manipulación de Cookies

- Cookie `AuthToken` no cifrada ni marcada como `Secure` ni `HttpOnly`
- Se puede modificar manualmente desde devtools

---

4. Exfiltración (simulada)

Identificación de datos sensibles expuestos o posibles vectores de fuga.

- API simulado: `/api/transfer`
- Key extraída vía esteganografía: `API_KEY: 7a19-df9x-0045-zk33-secret`
- Con acceso a sesión, es posible simular una transferencia sin validación CSRF.

---

5. Post-explotación (simulada)

Evaluación de persistencia y control del sistema (en entorno controlado).

- Panel `admin/` permite crear nuevos usuarios (sin autenticación multifactor)
- Logs del servidor muestran actividad, pero sin alertas ni bloqueo de IP
- Ningún mecanismo de detección de intrusiones activo (WAF o IDS)

---

6. Informe y recomendaciones

Vulnerabilidades explotadas:

Tipo	Severidad	Exploit usado
SQL Injection	Alta	Login sin filtrado
XSS Reflejado	Alta	`search.aspx`
Auth débil	Alta	`admin/admin`
CSRF Simulado	Media	`/transfer` sin token
Cookie insegura	Media	Sin `Secure`, sin `HttpOnly`

Entregables

Los estudiantes deberán construir una presentación de 30 minutos que contenga todos los elementos del informa del trabajo final de curso.