CTFs Learning Path: Hacking Web

Sesión 4

CSRF y SSRF

David Ramírez Acero
Carlos Freire Caballero

28 de marzo de 2023

Aula de
Ciber seguridad
y Redes

El Aula de Ciberseguridad y Redes presenta

CTFs Learning Path Hacking Web

David Ramírez Acero

Carlos Freire Caballero

Prepárate para hacer frente a todo tipo de CTFs sobre Hacking Web a través de **5 sesiones independientes** en las que ofreceremos los **conceptos clave** de esta categoría mientras resolvemos **multitud de CTFs** entre todos.

Sesión 1
SQL Injection
07/03

Sesión 2
XSS Injection
14/03

Sesión 3
Directory traversal, RFI y LFI
21/03

Sesión 4
CSRF y SSRF
28/03

Sesión 5
Repaso y ejercicios finales
11/04

Aula de Ciberseguridad y Redes

Todas las sesiones se realizarán a las **18:00** en el **aula B1 del Da Vinci**.

EL CONTENIDO DE ESTE TALLER SE IMPARTE ÚNICAMENTE CON

FINES EDUCATIVOS

HACED USO DE ESTE CONOCIMIENTO DE MANERA RESPONSABLE

QUIÉNES SOMOS

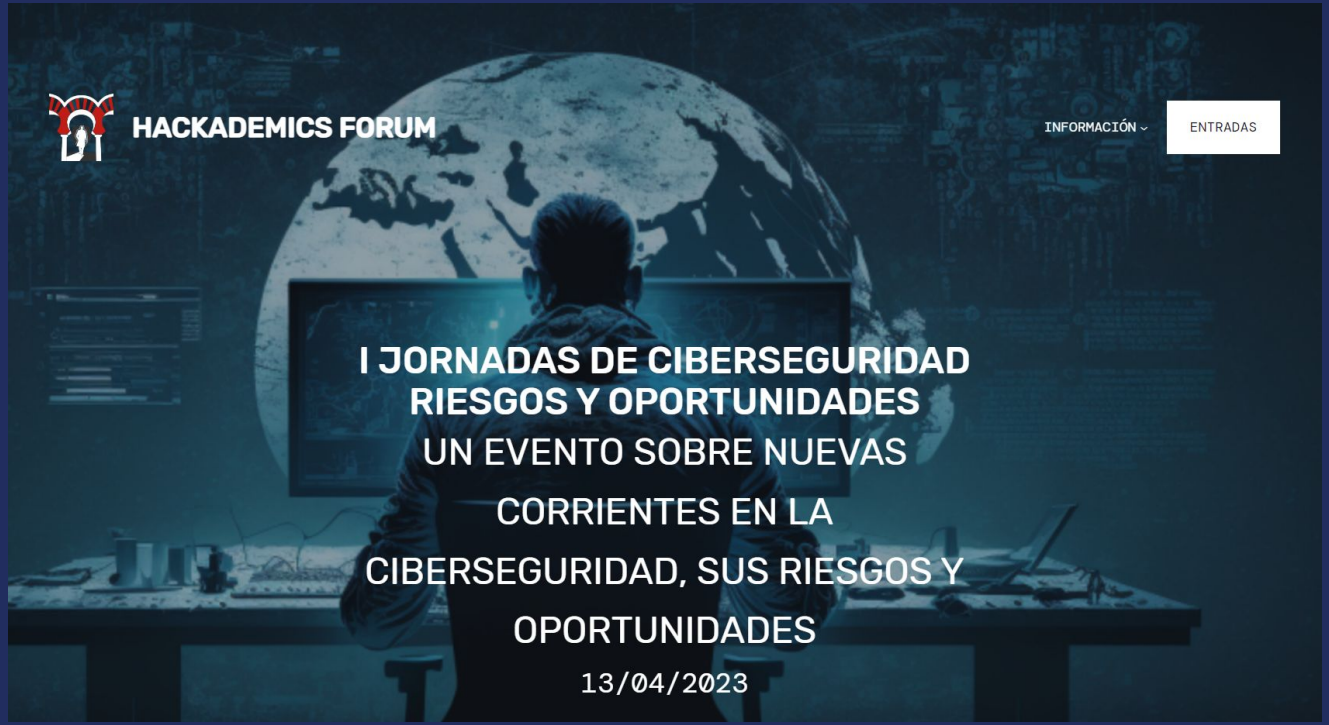
DAVID RAMÍREZ ACERO

CARLOS FREIRE CABALLERO

AULA

Aula de
Ciber seguridad
y Redes

Noticias


HACKADEMICS FORUM
INFORMACIÓN
ENTRADAS
I JORNADAS DE CIBERSEGURIDAD
RIESGOS Y OPORTUNIDADES
UN EVENTO SOBRE NUEVAS
CORRIENTES EN LA
CIBERSEGURIDAD, SUS RIESGOS Y
OPORTUNIDADES
13/04/2023

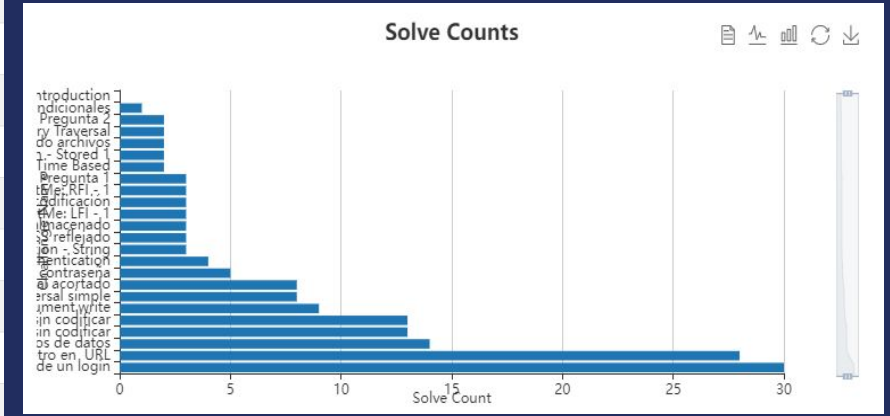
Noticias

HackTheBox Academy

Repaso de las sesiones anteriores

¿Dudas?

Place	User	Score
1	Carlos Marin	4414
2	naibu3	4160
3	Alvaro Palacios Criado	3662
4	Ebb4n	712
5	Txrrxs	661
6	i12mocab	638
7	serranito	388
8	pedro	252
9	pablo_dominguez_alcaide	168
10	Miguel Orzáez	168



Fechas

- **Viernes 7 de abril 12:00**. Apertura de los 5 ejercicios finales. Valdrán 1000 puntos cada uno.
- **Martes 11 de abril 18:00**. Cierre de todos los ejercicios de sesión y extra (los finales no incluidos). Se explicarán. También se anunciarán los premios.
- **Miércoles 12 de abril 23:59**. Cierre de los ejercicios finales. Termina la competición.
- **Jueves 13 de abril 14:00**. Entrega de premios en las jornadas Hackademics Forum.

Tabla de repaso

Client-Side	Server-Side
XSS Injection	SQL Injection
DOM-based	Directory Traversal
CSRF	File Related Vulnerabilities (LFI y RFI - RCE)
-	SSRF

¿Qué voy a necesitar?

- Ordenador (preferible Kali Linux).
- Estar conectado al wifi de uconet.

- Conocimientos básicos de HTML.
- Conocimientos básicos de páginas web.

¿Qué vamos a hacer?

- Falsificar peticiones de clientes para cambiar un correo electrónico de un servidor web usando CSRF.
- Bypassear defensas débiles de CSRF.
- Falsificar peticiones de servidores como APIs para obtener acceso a información privilegiada y funciones restringidas usando SSRF.

Ataques:

- CSRF.
- SSRF.

¿Por qué puede ser vulnerable?

Para CSRF se requieren 2 condiciones:

- Una cookie identifica al usuario.
- No existen parámetros impredecibles en la petición a atacar.

En SSRF a veces se necesitan ciertos requerimientos en los que un cliente emite peticiones directas al servidor. Y esto es peligroso.

Cross-Site Request Forgery (CSRF)

Se induce al usuario a hacer una petición maliciosa en un sitio web vulnerable.

Debe ser vulnerable el sitio web en el que se explote el ataque.

Por ejemplo: suponiendo que Facebook es vulnerable a CSRF, el usuario entra en un servidor web infectado y automáticamente se le hace una petición POST a su Facebook, que tiene su sesión abierta, para cambiar el correo por el del ciberdelincuente. Pasa a tener todo el control.

CSRF Defensas

- Usar un token CSRF identificador del usuario.
- Usar Same-Site Cookies.
- Usar validación del Referer.

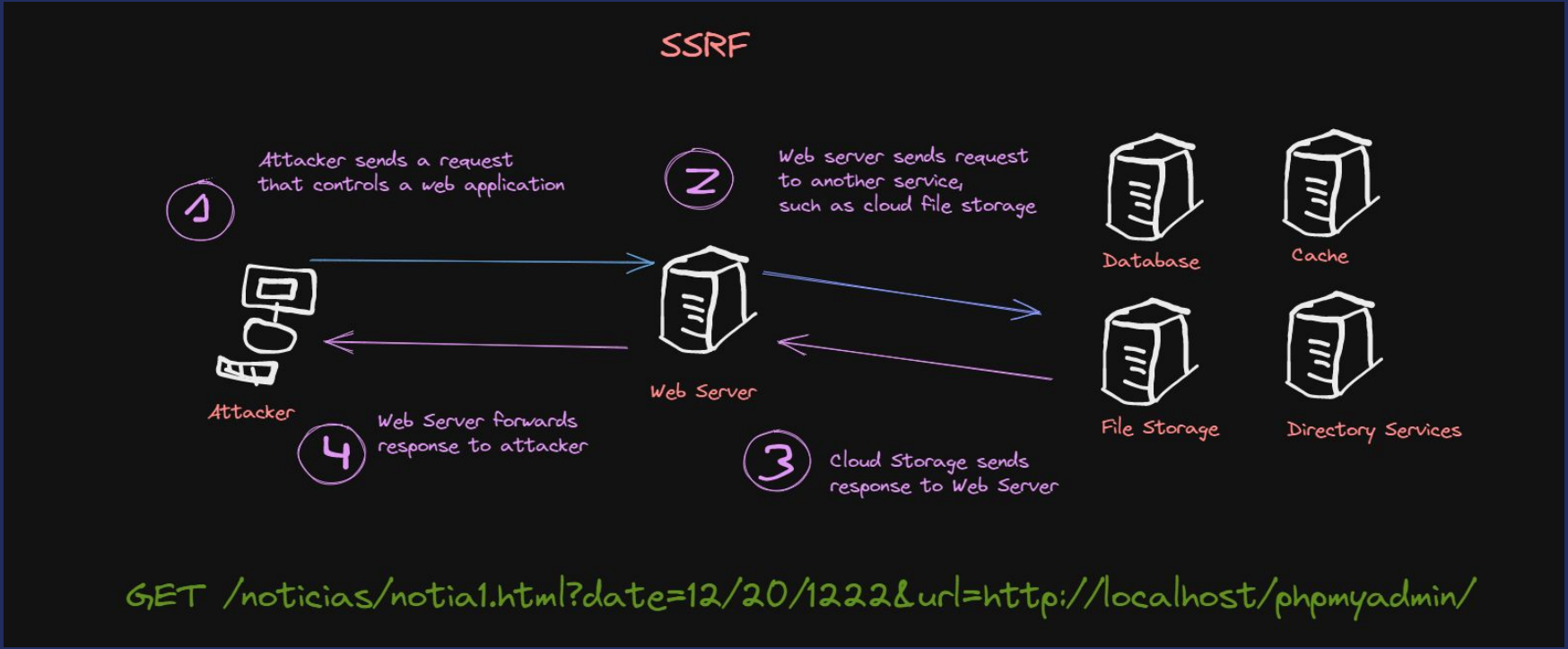
Server-Site Request Forgery (SSRF)

Permite hacer acciones desde el lado del servidor, lo que puede suponer ejecutar tareas con privilegios elevados.

Suele ocurrir en peticiones que debería el servidor hacer consigo mismo pero el cliente de forma indirecta interviene.

Por ejemplo: el servidor utiliza una API para obtener un dato, y este dato lo solicita el cliente. Si el atacante tiene acceso a la petición GET del dato puede cambiar https://API-URL por http://localhost/admin y visualizar así el panel de administrador.

Server-Site Request Forgery (SSRF)



Teoría

CSRF: https://portswigger.net/web-security/csrf

XSS vs CSRF: https://portswigger.net/web-security/csrf/xss-vs-csrf

CSRF Bypassing token validation:
https://portswigger.net/web-security/csrf/bypassing-token-validation

CSRF Bypassing SameSite cookie restrictions:
https://portswigger.net/web-security/csrf/bypassing-samesite-restrictions

CSRF Bypassing Referer-based defenses:
https://portswigger.net/web-security/csrf/bypassing-referer-based-defenses

SSRF: https://portswigger.net/web-security/ssrf

CTF PortSwigger 1: CSRF

Este ejercicio se realiza de manera guiada.

Pasos:

1. Acceded a la plataforma CTFd: http://150.214.112.168/
2. Entrad en: https://portswigger.net/web-security/csrf/lab-no-defenses
3. Responded la pregunta de la plataforma de CTFd.

CTF PortSwigger 1: SSRF

Este ejercicio se realiza de manera guiada.

Pasos:

1. Acceded a la plataforma CTFd: http://150.214.112.168/
2. Entrad en: https://portswigger.net/web-security/ssrf/lab-basic-ssrf-against-localhost
3. Responded la pregunta de la plataforma de CTFd.

Ejercicios para hacer

- PortSwigger 1. CSRF sin defensas.
- PortSwigger 2. CSRF con token asociado a cookie.
- PortSwigger 3. CSRF con Same-Site Lax sobreescribiendo método.
- PortSwigger 4. CSRF con validación de Referer.
- PortSwigger 5. SSRF sin defensas.
- PortSwigger 6. SSRF con blacklist.
- PortSwigger 7. SSRF con whitelist.

- RootMe. Web - Client. CSRF - token bypass.