CTFs Learning Path: Hacking Web

Sesión 2

XSS INJECTION

David Ramírez Acero
Carlos Freire Caballero

14 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


/Rootəd©CON
INICIO
NOTICIAS
ROOTEDCON 2023
REGISTRO
EQUIPO
ARCHIVO
LOGIN
HOWTO
EN
#rooted2017
WORLD WAR I
Mikko Hypponen
/Rootəd©CON
Madrid 2023
9-11 Marzo 2023
COMPRA ENTRADAS
FORMACIONES
ROOTEDCON 2023
AGENDA
PONENTES
HACKERNIGHT

Noticias


INICIO
INFORMACIÓN
GALERÍA
PROGRAMA
REGISTRO
PREGUNTAS FRECUENTES
LOCALIZACIÓN
PATROCINADORES
Security
High School
#SHS2k23
VIII Edición
16 y 17 de Marzo de 2023 - Córdoba
Talleres Jueves Día 16 marzo - Lugar: IES Fidiana
Ponencias Viernes Día 17 marzo - Lugar: Salón De Actos Juan XXIII - Campus de Rabanales
VER MÁS
REGISTRARSE

Repaso de la sesión 1

¿Dudas?

Cierre de los ejercicios de sesión

Formato de puntuación de los ejercicios

Repaso

Client-Side	Server-Side
XSS Injection	SQL Injection
DOM-based	
-	

¿Qué voy a necesitar?

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

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

¿Qué vamos a hacer?

- Clasificar las diferentes inyecciones posibles.
- Inyectar en los diferentes contextos.
- Explotación y utilidad en los diferentes contextos.

Ataques:

- Reflejados (Reflected XSS Injection).
- Almacenados (Stored XSS Injection).
- Basados en DOM (DOM-Based XSS Injection).

Introducción a XSS

Inyectar código normalmente de Javascript en campos formularios en los siguientes contextos:

- En el interior del HTML.
- En el interior de un script en Javascript.

<h1> 'Campo que se puede modificar' </h1>

<script> var input = 'Campo que se puede modificar'; </script>

Contexto

El estudio del contexto nos llevará a hacer una inyección u otra.

También es parte del contexto si se codifican los datos introducidos, si existen bloqueos de caracteres o tags…

IMPORTANTE: diferenciar entre el reconocer la inyección a explotarla.

¿Por qué puede ser vulnerable?

La inyección de Javascript a nivel de cliente, aún teniendo restricciones, es poderosa.

- Robo de cookies.
- Robo de credenciales.
- Descarga de programas maliciosos.

Se evita bloqueando la introducción de caracteres como <, >, tags o con herramientas más complejas de sanitización de entradas del usuario.

Reflected XSS

La inyección obtiene una respuesta inmediata en el cliente.

Peligrosidad limitada. Requiere de pasos extra para realizar el ataque.

Por ejemplo, producir un alert() cuando se realiza una búsqueda.

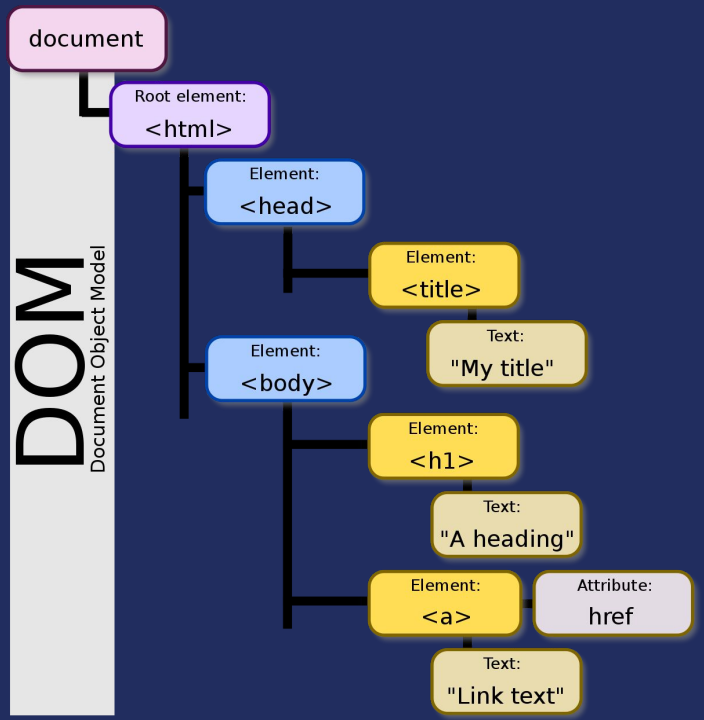
Stored XSS

Almacena la inyección y se realiza para cada cliente que accede al lugar inyectado.

Mucho más peligrosa.

Por ejemplo, el alert() se almacena en un comentario que se ejecuta para cualquier persona que lo vea.

DOM-Based XSS



DOM-Based XSS

Explota métodos que modifican de alguna forma el DOM.

Más complicados de detectar pero muy poderosos.

Por ejemplo, document.write() incluye una imagen en el DOM y se concatena con una entrada del usuario.

Estos ataques pueden ser también reflejados o almacenados.

Teoría

General: https://portswigger.net/web-security/cross-site-scripting

Reflejados: https://portswigger.net/web-security/cross-site-scripting/reflected

Almacenados: https://portswigger.net/web-security/cross-site-scripting/stored

Basados en DOM:
https://portswigger.net/web-security/cross-site-scripting/dom-based

Contextos: https://portswigger.net/web-security/cross-site-scripting/contexts

Explotación: https://portswigger.net/web-security/cross-site-scripting/exploiting

CTF PortSwigger 1: XSS reflejado en contexto HTML sin codificar

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/cross-site-scripting/reflected/lab-html-context-nothing-encoded
3. Cuando tengáis la inyección realizada acceded a CTFd e introducidla como flag.

Ejercicios para hacer

- PortSwigger 1. XSS reflejado en contexto HTML sin codificar.
- PortSwigger 2. XSS almacenado en contexto HTML sin codificar.
- PortSwigger 3. DOM XSS en document.write.
- PortSwigger 4. DOM XSS reflejado.
- PortSwigger 5. DOM XSS almacenado.

- RootMe. Web - Client. XSS - Stored 1.
- RootMe. Web - Client. XSS DOM Based - Introduction.