Cross Site Scripting (XSS)

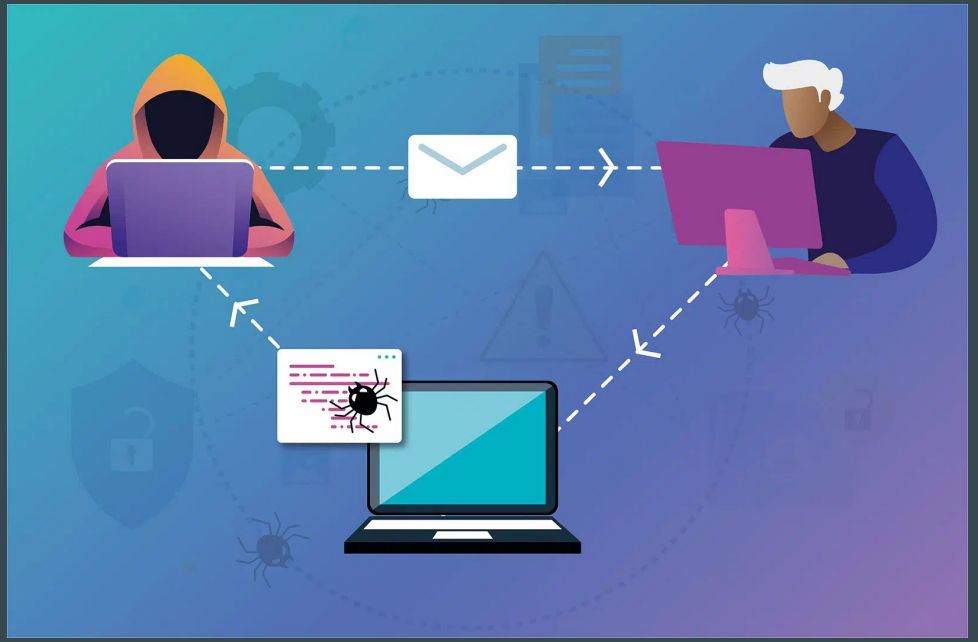
Seguridad e Integridad de Sistemas

Prof. GIANNI, Fernando Julian

¿Qué es un XSS?

Es una vulnerabilidad que permite a un atacante **inyectar código arbitrario** (JavaScript o similares) a través de un **parámetro** vulnerable en el sitio web afectado, con el objetivo de que se ejecute **en el navegador de los usuarios** que visiten el enlace especialmente diseñado.

Client side attacks



TIPOS DE XSS

Reflejado

- No persiste en el servidor.
- Es necesario la interacción del usuario final.
- Ingeniería social.

Persistente

- Persiste en el servidor indefinidamente.
- Comentarios, blogs, chats.
- Mayor criticidad.

DOM

- Mayor complejidad de explotación.
- Document Object Model.
- Abusa de funciones JS como innerHTML().

Reflected XSS

Un XSS reflejado surge cuando una aplicación recibe datos en una solicitud HTTP y los incluye en la respuesta inmediata de forma insegura.

Impacto

La explotación de esta vulnerabilidad, permitiría, entre otras cosas, acceder a cookies, tokens de sesión u otra información confidencial retenida por el navegador y utilizada en el sitio web.

- Recopilación de datos personales.
- Robo de credenciales (usuarios y contraseñas).
- Robo de cookies.
- Redireccionamiento a sitios maliciosos.
- Acceso al control del equipo de la víctima.

Laboratorio

https://portswigger.net/web-security/cross-site-scripting/reflected/lab-html-context-nothing-encoded

Detección XSS

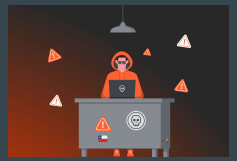
POST /activities/api/notes

Enumeración

Enumerar todos los endpoints con funcionalidades de ingreso de datos. Ej. blog, notas, búsquedas.

Detección

Búsqueda de parámetros que reflejen su contenido en la respuesta del sitio web



Explotación

Encontré un posible XSS. ¿Hasta dónde puedo llegar? ¿Puedo obtener las cookies de sesión?

¿Cuál es el impacto?

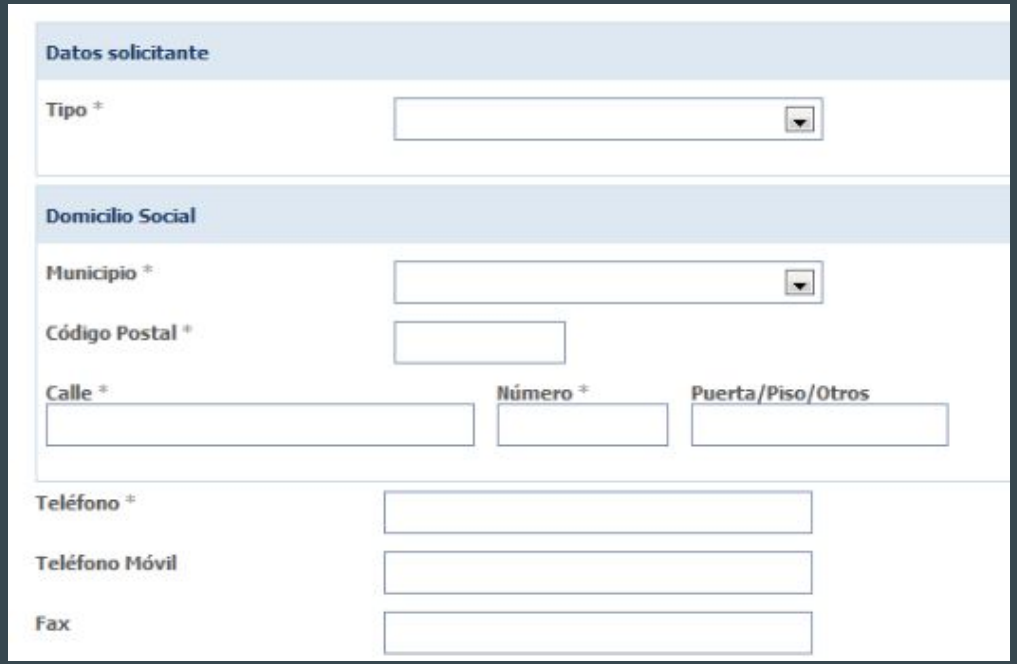
Automatización

Es posible juntar una lista de payloads y probar si alguno de ellos es reflejado o persiste

Validación

Prueba y error, revisar el código del sitio web y tratar de romper el mismo. Ej: etiquetas HTML.

TIPS



Puntos de entrada

- QueryParams
- Formularios
- Cabeceras HTTP
- Cookies
- Archivos y sus contenidos

Todo en el request puede ser un punto de entrada.

Mitigación

Output encoding / HTML, CSS, JS

Characters	Decimal	Hexadecimal	HTML Character Set	Unicode
" (double quotation marks)	&#34;	&#x22;	&quot;	\u0022
' (single quotation mark)	&#39;	&#x27;	&apos;	\u0027
& (ampersand)	&#38;	&#x26;	&amp;	\u0026
< (less than)	&#60;	&#x3C;	&lt;	\u003c

Expresiones regulares

```
String email = request.getParameter("email");
String expression = "^[a-z0-9!#$%&'*+/=?^_`{|}~-]
+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*@(?:[a-z0-9]
(?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?$";
```

¿Qué validamos?

Estructura	40eb37bd-7d94-4506-b4d8-b358fcb80d24	uuid | email | ...
Contenido	¿Qué debo recibir? ¿Es lo que necesito?	int | float | array
Longitud	¿Cualquier tamaño?	<10 | >32 | <600
Contexto	¿Pagos negativos? ¿Fechas anteriores?	Business Logic

¡ LA SUMA DE TODO NOS DA SEGURIDAD !