sustantiva
CON•SENTIDO

Explotación de Vulnerabilidades Comunes con Python

Automatización Ética de Pruebas de SQLi y XSS

Introducción al Contexto de Riesgo Web

Introducción al Contexto de Riesgo Web

- Las aplicaciones web son puntos frecuentes de exposición y ataque.
- Vulnerabilidades como SQLi y XSS siguen presentes en entornos reales.
- Automatizar su detección es clave para pruebas de seguridad frecuentes, éticas y efectivas.
- Python se destaca por su versatilidad, simplicidad y poder para scripting ofensivo.

Fundamentos de SQL Injection (SQLi)

Fundamentos de SQL Injection (SQLi)

- Técnica que manipula consultas SQL desde entradas no validadas.

- Permite:
  - Bypass de autenticación
  - Acceso a datos sensibles
  - Modificación o eliminación de información

- Variantes:
  - SQLi clásica
  - Ciega
  - Basada en tiempo
  - De segundo orden

Fundamentos de Cross-Site Scripting (XSS)

- Inserción de código malicioso en páginas legítimas.

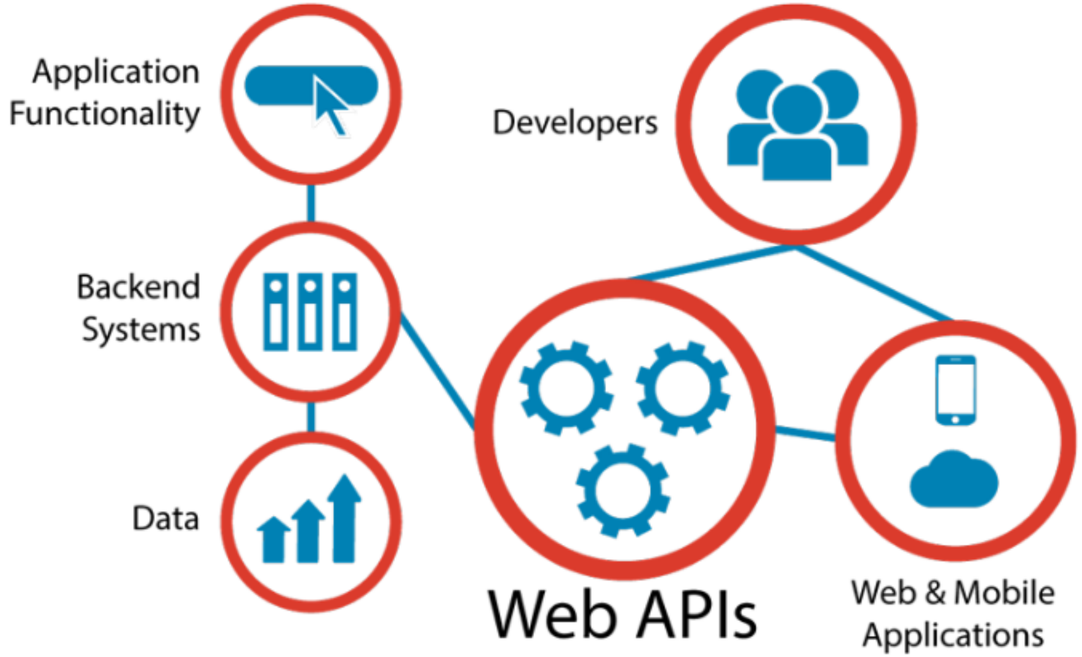
- **Impacto:**
  - Robo de cookies
  - Secuestro de sesiones
  - Redirecciones maliciosas

- **Tipos:**
  - Reflejado
  - Almacenado
  - Basado en DOM

Principios Éticos en el Desarrollo de Scripts

Principios Éticos en el Desarrollo de Scripts

- Modularidad y mantenimiento del código
- Manejo robusto de errores y logs
- Consumo controlado de recursos
- Declaración de límites legales y éticos
- Uso exclusivamente autorizado en contextos de prueba



Bibliotecas de Python para Pentesting Web

- **requests:** envío de solicitudes HTTP
- **BeautifulSoup:** parsing HTML
- **sqlmapapi:** conexión con motor de SQLMap
- **OWASP ZAP API:** automatización de análisis
- **Selenium:** automatización de validación en navegador

SQL Injection Simple

SQL Injection Simple

url = "http://example.com/vulnerable.php?user="
payload = "' OR '1'='1"
response = requests.get(url + payload)

- Si el resultado contiene información sensible o acceso no autorizado, puede indicar SQLi.
- El ejemplo puede escalarse con diccionarios, fuzzing o integración con SQLMap.

XSS Básico

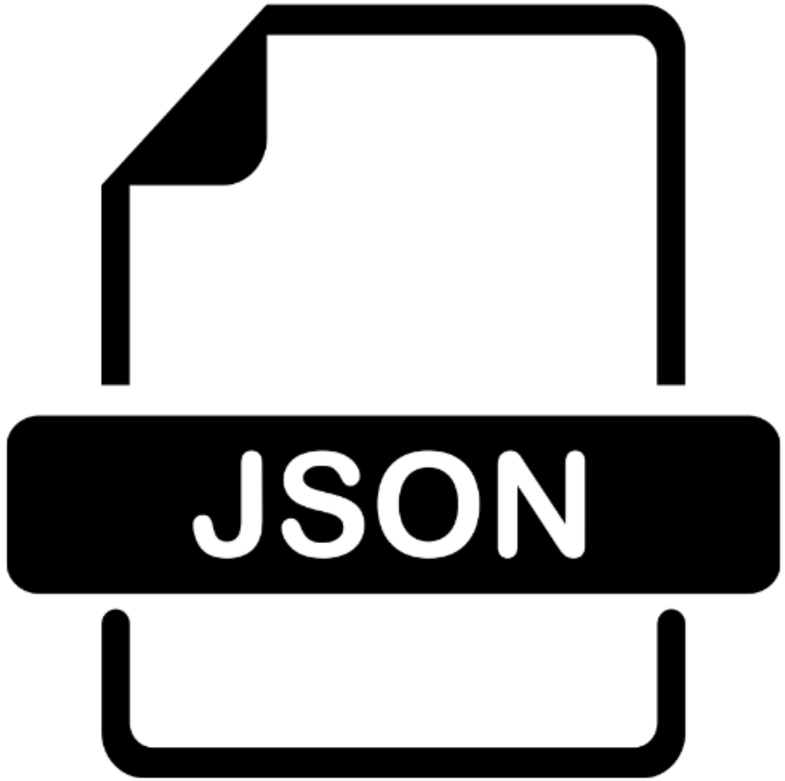
XSS Básico

url = "http://example.com/search?query="
payload_xss = "<script>alert('xss')</script>"

- Se evalúa si el input es reflejado sin sanitización.
- La automatización puede ampliarse con Selenium para observar ejecución real del script.

APIs RESTful – Superficie Moderna de Ataque

- Las APIs permiten inyecciones en JSON si no validan entradas.
- Ejemplo de SQLi sobre JSON POST:
  - **payload = {"username": "' OR 1=1 -- ", "password": "irrelevante"}**
- Importancia de validar respuestas, errores y códigos de estado HTTP.




Energiza!