Mi primer Laboratorio para Pentest

Instrucciones de Uso

Bienvenidos

En esta charla, vamos a aprender temas básicos a considerar dentro de una metodología de pruebas de penetración para aplicaciones web de forma estándar y ortodoxa, recuerda que siempre el ingrediente *secreto* es pensar fuera de la caja y que estos son únicamente los pasos para empezar.

Materiales

Metodología

Aplicación vulnerable

Herramienta

Utensilios

WebApp Vulnerable: http://192.168.147.103:8000/

posadev:p0s4d3v1

Otra app vulnerable:

https://ginandjuice.shop/

Antes de empezar

- La información colectada en una etapa, puede hacer que sea necesario volver a una etapa anterior y realizar ataques más focalizados, es decir, dar vueltas a veces no es estar perdido.
- Encontrar una vulnerabilidad clave en un área de la aplicación, puede acortarnos trabajar en otras áreas de la misma.
- El resultado de nuestras pruebas en algunas partes de la aplicación, probablemente nos muestre cosas que inmediatamente debemos probar en otras partes.

Instrucciones

Paso 1: Reconocimiento y análisis

1. Mapear el contenido de la aplicación

https://ginandjuice.shop/

Instrucciones

1. Mapear el contenido de la aplicación

Contenido Ligado	**Otro contenido**	**Métodos de acceso no estándar**
Explorar contenido visible	Descubrir contenido oculto	funciones especificadas por identificador
Consultar los recursos públicos	Descubrir contenido por defecto	Debug sobre los parámetros

Instrucciones

Paso 1: Reconocimiento y análisis

1. Mapear el contenido de la aplicación
2. Analizar la aplicación

Instrucciones

2. Analizar la aplicación

Identificar la funcionalidad

Identificar los puntos de entrada

Identificar las teconolgías

Superficie de ataque

Instrucciones

Paso 2: Búsqueda y explotación

1. Mapear el contenido de la aplicación
2. Analizar la aplicación

Instrucciones

Paso 2: Búsqueda y explotación

1. Mapear el contenido de la aplicación
2. Analizar la aplicación

Lógica de la aplicación

- 3 Client-side controls
- Logic Flaws 9

Manejo de Acceso

- Authentication 4
- 5 session management
- Access Controls 6

Manejo de Entradas

- 7 Fuzz all parameters
- Issues with specific fuctionality 8

Hosting de la aplicación

- Shared hosting issues 10
- 11 Test the web server

Instrucciones

Client-side controls

LOGIC Flaws

Instrucciones

- Busca en los mecanismos multietapa (multiproceso)
- Busca cualquier tipo de mal funcionamiento que permita ejecución
- Busca entender cada parámetro enviado en la solicitud y el mal funcionamiento que ocasiona.
- Busca deshabilitar los mecanismos de validación antes de que sean enviados
- Busca fallos en la confianza de los límites
- Busca fallos en la lógica de las transacciones

Test #1

Side Controls

Logic Flawes

Instrucciones

Authentication

Session management

Access Controls

Instrucciones

- Busca por fallos en las condiciones abiertas (open-conditions)
- Busca fallos en la calidad de los passwords, adivinación y posible enumeración de usuario
- Busca transmisiones sin redirección
- Prueba las herramientas como “recuerdame” y “Recuperación de cuentas”
- Busca almacenamiento inseguro
- Busca impesonización

Test #2


PyGoat
Django Vulnerable Web Application
(made by OWASP)


Management


Session Control

Instrucciones

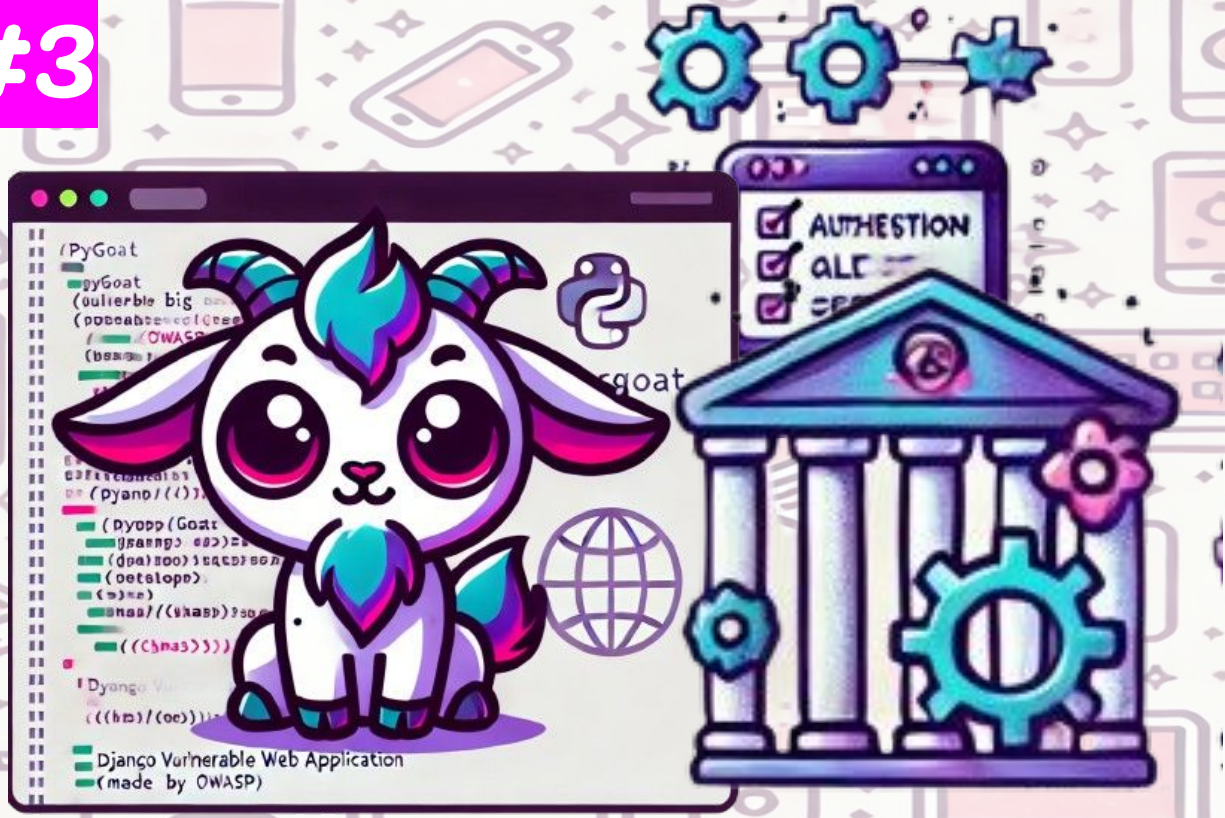
Fuzz all parameters

Issues with specific functionality

Instrucciones

- Entender la causa raíz y el funcionamiento para SQL Injection
- Entender y detectar el tipo de XSS que se nos presenta
- Buscar path traversal, script injection, OS Command Injection en la manipulación de parámetros.

Test #3


PyGoat
Django Vulnerable Web Application
(made by OWASP)
AUTHESTION

Instrucciones

Shared hosting issues

Test the web server

Test #4

Instrucciones

Paso 3: Es solo el comienzo

1. Mapear el contenido de la aplicación
2. Analizar la aplicación

Lógica de la aplicación

- 3 Client-side controls
- Logic Flaws 9
- Miscellaneos check
- Information Leakage

Manejo de Acceso

- Authentication 4
- 5 session management
- Access Controls 6

Manejo de Entradas

- 7 Fuzz all parameters
- Issues with specific fuctionality 8

Hosting de la aplicación

- Shared hosting issues 10
- 11 Test the web server

MiAlegría

```
# AFL (American Fuzzy Lop)
sudo apt install -y afl

# w3af
sudo apt install -y python3-pip
pip3 install w3af-api-client

# Nikto
sudo apt install -y nikto

# Nuclei
sudo apt install -y nuclei

# Completion message
echo -e "\nAll tools have been installed successfully!\n"

# Menu display
clear
echo -e "${PURPLE}=================================================${RESET}"
echo -e "${PINK}          Vulnerability Assessment Tools         ${RESET}"
echo -e "${CYAN}=================================================${RESET}"
echo -e "${PURPLE}1.${RESET} BeEF (Client-Side Controls)"
echo -e "${PINK}2.${RESET} Hydra (Authentication)"
echo -e "${CYAN}3.${RESET} Medusa (Authentication)"
echo -e "${PURPLE}4.${RESET} wfuzz (Fuzz All Parameters)"
echo -e "${PINK}5.${RESET} AFL (Fuzz All Parameters)"
echo -e "${CYAN}6.${RESET} w3af (Issues with Specific Functionality)"
echo -e "${PURPLE}7.${RESET} Nikto (Shared Hosting Issues, Test Web Server)"
echo -e "${PINK}8.${RESET} Nuclei (Shared Hosting Issues)"
echo -e "${CYAN}=================================================${RESET}"
echo -e "\nTo run a tool, simply type its name in the terminal."
```

¡Sirva & Disfrute!

MiAlegría

WOMCY

LATAM Women in Cybersecurity

De la speaker

- 8 años de experiencia en Ciberseguridad
- 3 en seguridad ofensiva
- 2 años en ESET Latinoamérica
- Entrega de servicios ofensivos para todos los países de LATAM
- 3 años de voluntariado en diferentes comunidades de infosec
- Actualmente líder del Programa WOMCY TECH
- Speaker para varias universidades/congresos/conferencias
- Mentora en WOMCY Latam Women In Cybersecurity
- Maestra en Seguridad de la Información & Ciencias de la computación
- Top Women In Cybersecurity 2023
- En curso: Master WomenCISO de Google

MiAlegría