

Auditoría Web

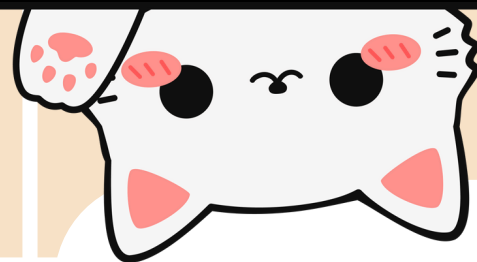


---Propuesta---

AGENDA

- Metodologías más comunes
- Fases de una auditoría web y algunas Herramientas
 - Planificación
 - Reconocimiento y Enumeración
 - Análisis de vulnerabilidades
 - Explotación
 - Post Explotación
 - Documentación
- ¿Dónde Entrenar?
- Laboratorios recomendados





Metodologías más comunes

- OWASP Web Security Testing Guide (WSTG) [www](#)
- PTES (Penetration Testing Execution Standard) [www](#)
- NIST SP 800-115 [www](#)
- Mixtas o Personalizadas



Fase - Planificación

- Definición del Alcance (Scope) - Lo que sí.
- Objetivos del Pentesting - ¿Qué esperamos?
- Tipo de Prueba - ¿Caja negra, gris o blanca?
- Límites y reglas - Lo que NO
- Firmas de documentos o Autorización expresa - Respaldo
- Tiempos y entregables - Cuando y qué



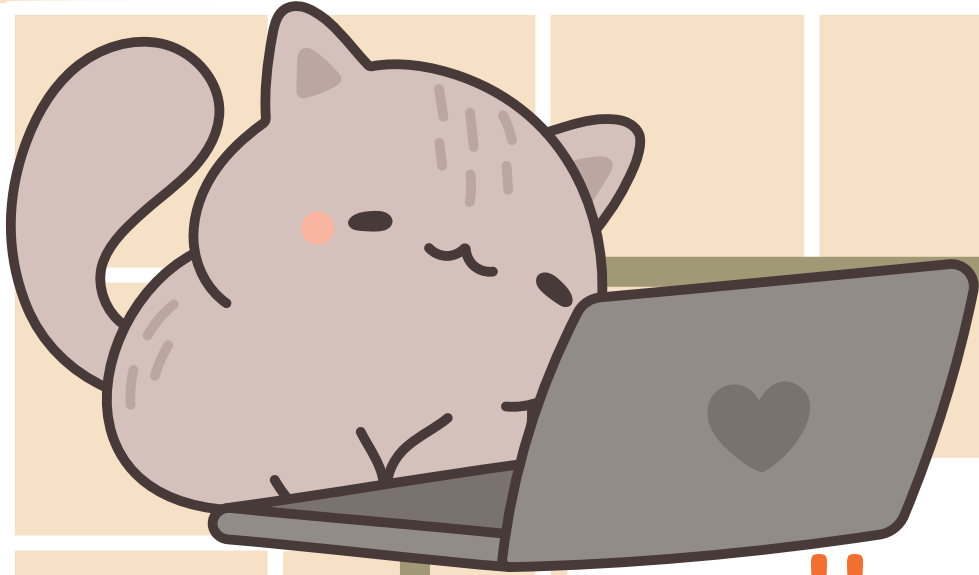
Nunca te fies de un acuerdo verbal, siempre trata de respaldar las solicitudes que te realizan, para de esta manera evitar futuros problemas.

Fase - Reconocimiento y Enumeración

Recopilación de información (Reconocimiento): Obtener información del objetivo como nombres de dominio, IPs, subdominios, puertos abiertos, tecnologías visibles, certificados, etc. Puede ser pasiva (sin tocar el objetivo) o activa (interacción directa con el sistema).

Enumeración: Interactuar activamente con el sistema para descubrir tecnologías utilizadas, versiones, usuarios válidos, archivos, directorios, endpoints, parámetros y servicios internos accesibles que puedan ser vulnerables.





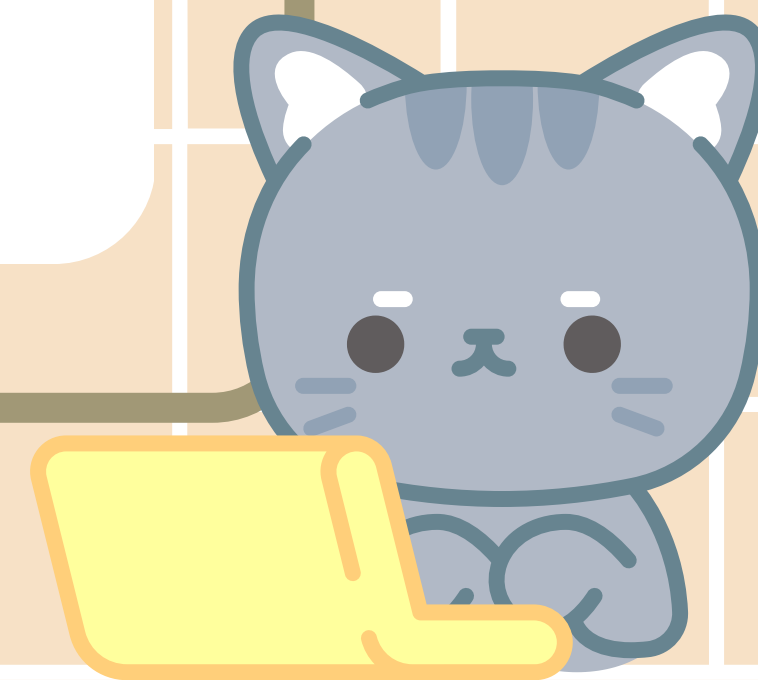
Herramientas – Reconocimiento y Enumeración

Reconocimiento Activo: Nmap, WhatWeb, Wappalyzer, builtwith, Gobuster, FFUF, Dirb, Dirbuster

Reconocimiento Pasivo: Whois, Dnsdumpster, Shodan, Google Dorking, Sublist3r, Amass, theHarvester, Subfinder, Wayback Machine

Enumeración: Nmap, ParamSpider, Burp Suite, ZAP, Gobuster, FFUF, WPScan

Puede que Recon Activo y Enumeración puedan ser similares para algunos, pero la diferencia radica en la profundidad de lo buscado

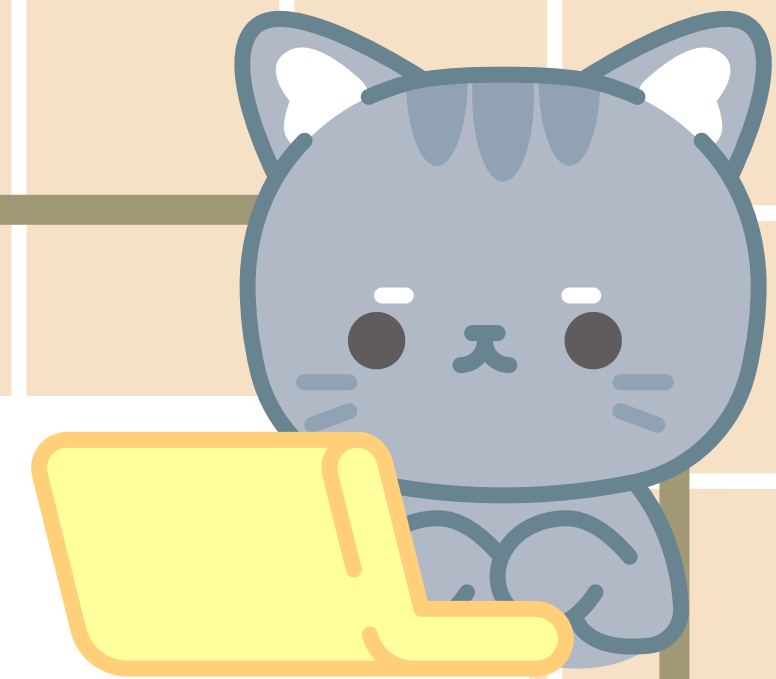
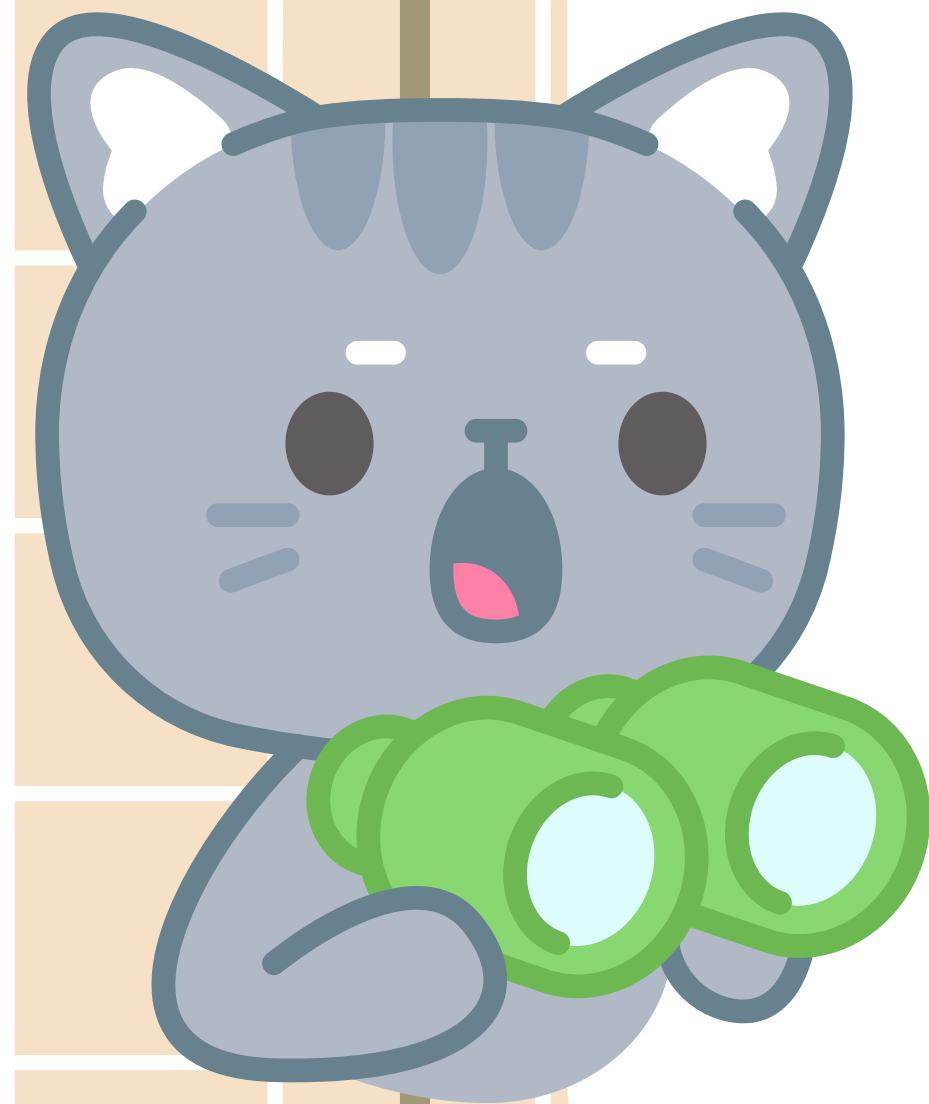


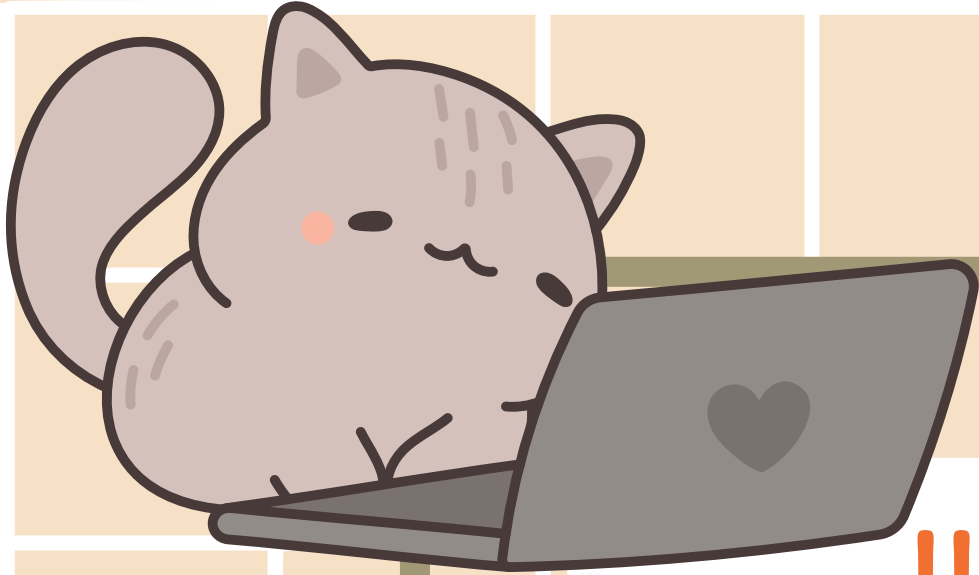
Fase - Análisis de Vulnerabilidades

En esta fase evaluamos el sistema en busca de vulnerabilidades técnicas y lógicas, determinando su riesgo, impacto y facilidad de explotación.

Se validan fallas como:

- Puertos abiertos expuestos (ej: 22, 445)
- Inyecciones: SQLi, XSS, SSTI, Command Injection
- CSRF, LFI, IDOR, Broken Access Control (BAC)
- Path Traversal, Directory Listing, Sensitive File Disclosure (.git, .env)
- Vulnerabilidades en la lógica de negocio o procesos mal diseñados





Herramientas – Análisis de Vulnerabilidades

Escaneo automatizado:

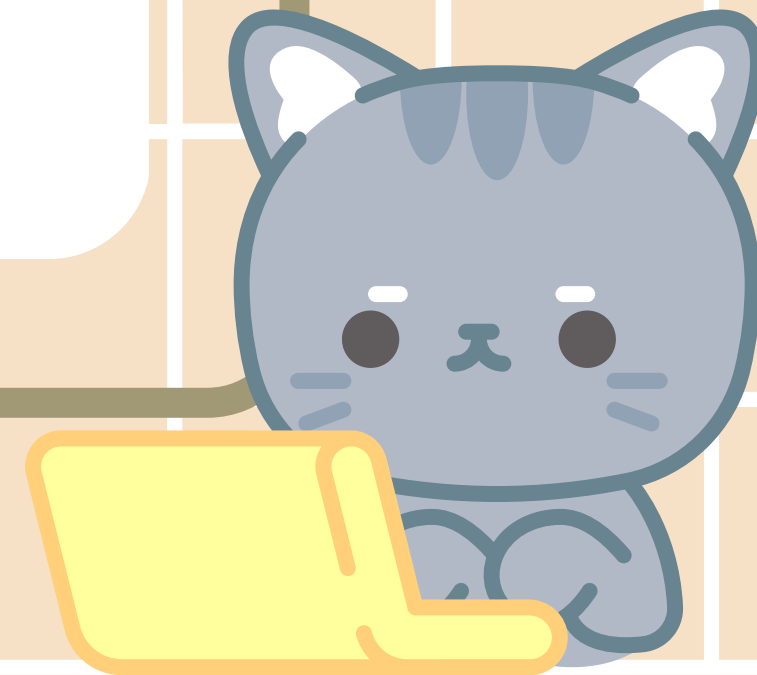
- Nikto → configuración, headers, CGI, etc.
- Nuclei → escaneo de vulnerabilidades conocido
- ZAP, Burp Suite Scanner → escaneo activo/pasivo
- Dalfox → XSS automático
- XSSStrike → análisis avanzado de XSS
- ParamSpider → descubrir parámetros ocultos (para fuzzing)

Análisis manual:

- Postman, curl → Peticiones HTTP personalizadas
- DevTools (Chrome) → Análisis JS, parámetros ocultos, endpoints

Inyección y prueba de fallos:

- sqlmap → SQL Injection
- joomscan, wpscan → CMS vulnerables
- ffuf, feroxbuster → Fuzzing de rutas y parámetros (con payloads)



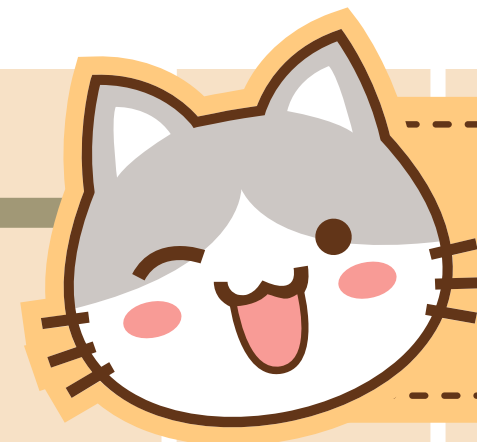
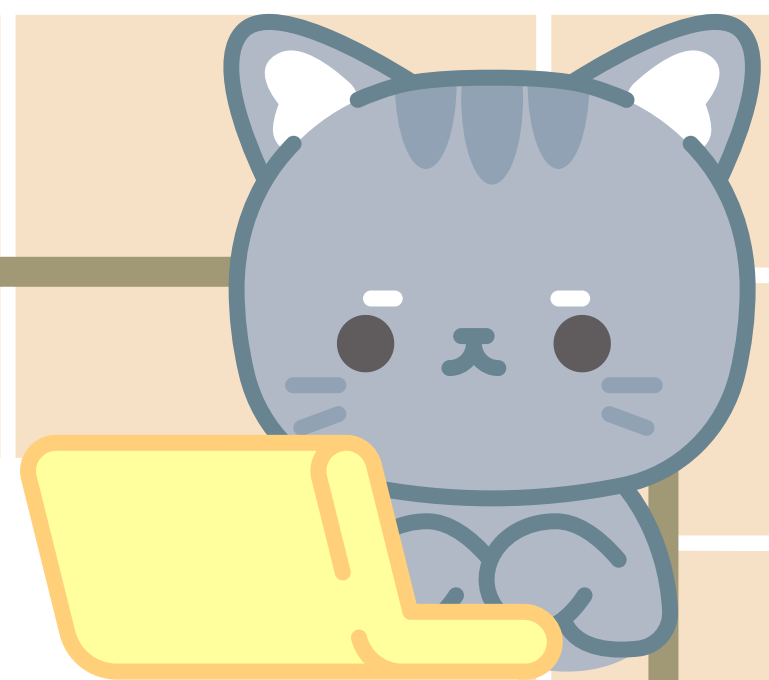
Fase - Explotación

En esta etapa aprovechamos las vulnerabilidades detectadas para demostrar su impacto real sobre el sistema.

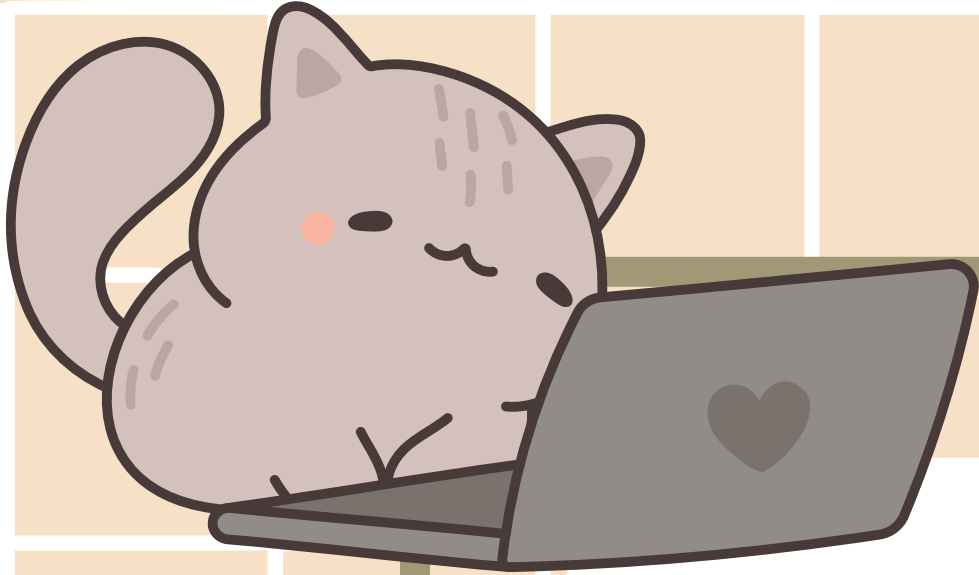
Se ejecutan acciones como:

- Robo de información sensible (tokens, credenciales, sesiones)
- Escalada de privilegios
- Acceso a funciones restringidas
- Ejecución remota de comandos (RCE)
- Manipulación de parámetros (IDOR, Broken Access Control)
- Envío de payloads XSS, SQLi, LFI, SSTI, etc.

El objetivo no es causar daño, sino validar hasta dónde puede llegar un atacante si explota esa vulnerabilidad.

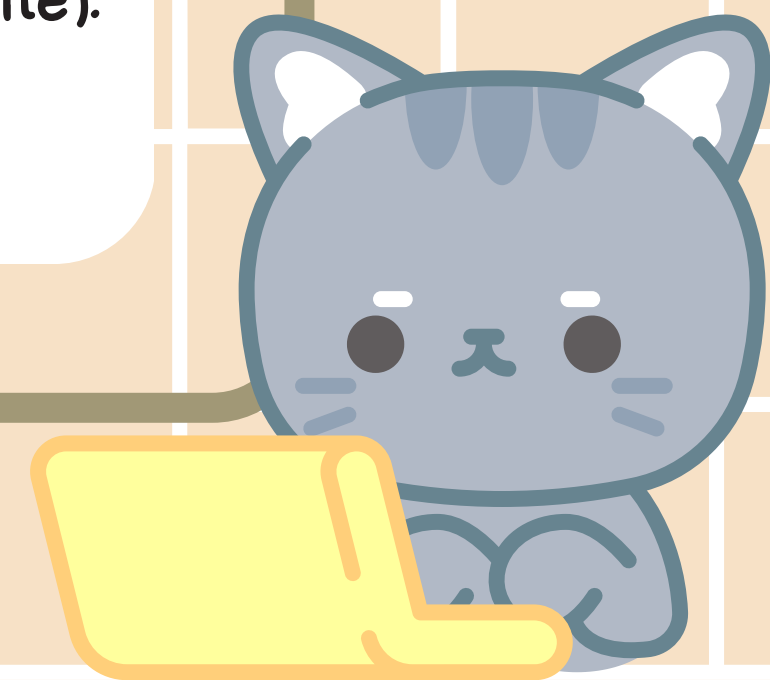


Toda explotación debe quedar documentada con evidencias (PoC, capturas, request/response) para respaldar el informe final.



Herramientas – Explotación

- Sqlmap → Para explotar inyecciones SQL y extraer datos.
- Commix → Explota Command Injection (cmd=).
- XSS Hunter → Para ataques XSS tipo blind (admin que ejecuta tu payload).
- jwt-tool → Para analizar y modificar tokens JWT (falsificación, decodificación, etc.).
- Burp Suite → Repeater + Intruder para explotar XSS, IDOR, Auth Bypass, etc.
- Curl / Postman → Explotación manual de APIs con peticiones personalizadas.
- Cookie-Editor o ModHeader → Para modificar headers, cookies, etc.
- Hydra / patator / Medusa → Ataques de fuerza bruta a login.
- Metasploit → Para exploits más complejos o post-explotación (solo si el entorno lo permite).
- XSSStrike / Dalfox → XSS testing avanzado y automático.



Fase - Post Explotación

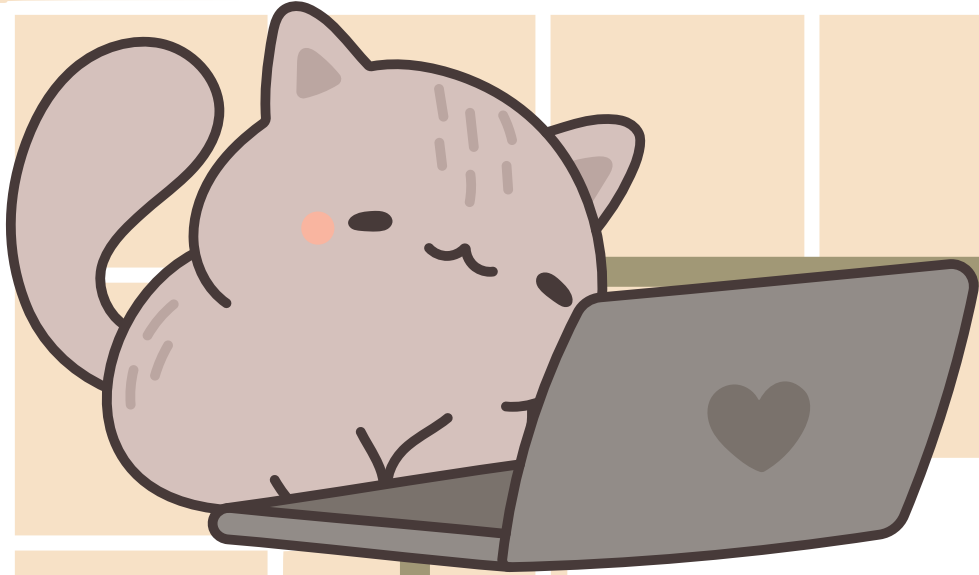
Esta fase se aplica solo si el alcance lo permite.

Se evalúa qué tan grave es el compromiso después de una explotación exitosa.

Se ejecutan acciones como:

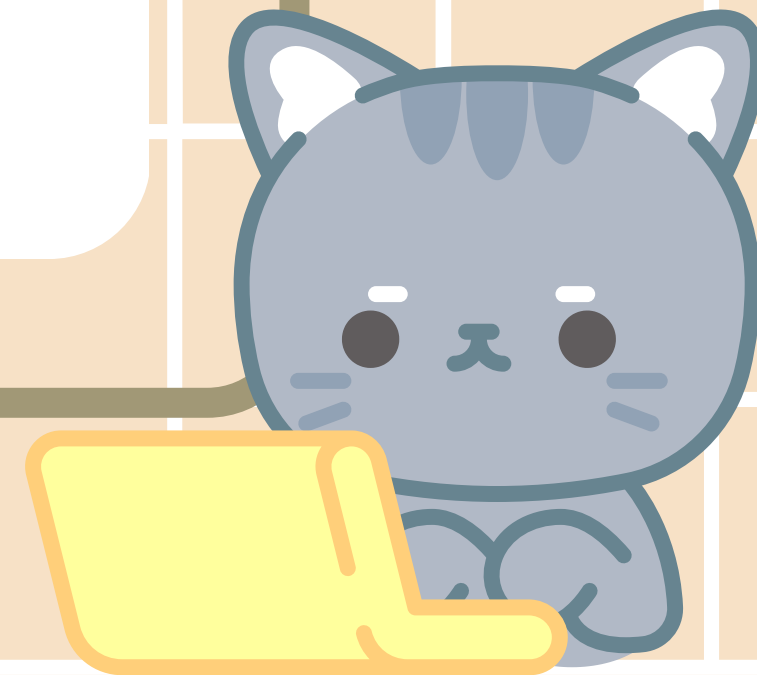
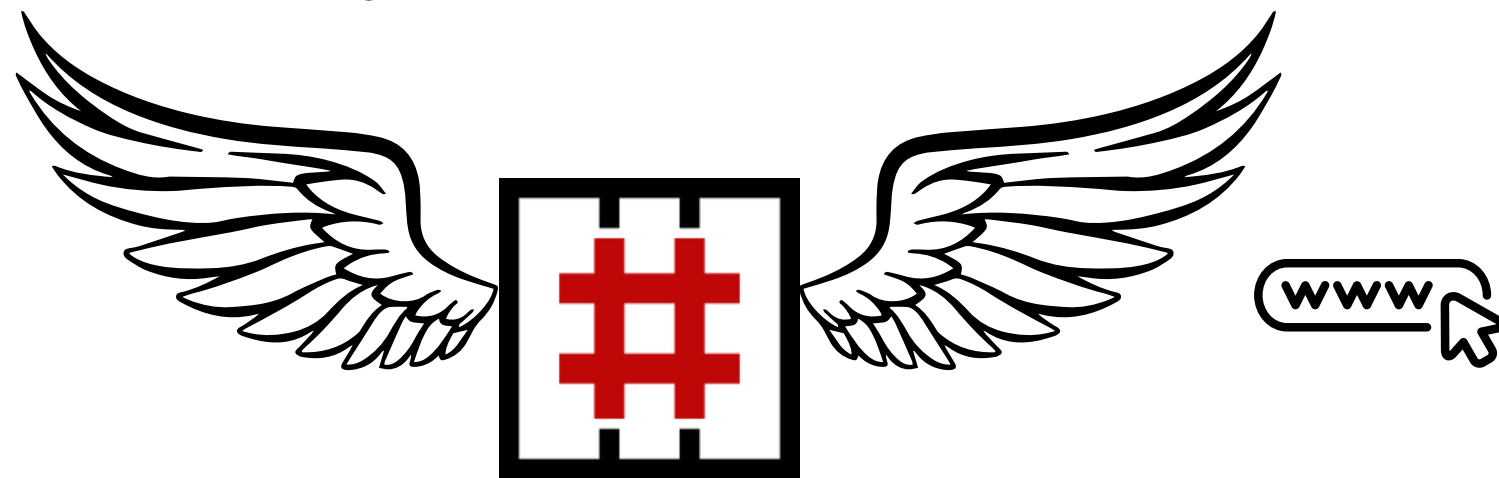
- Acceso y extracción de información crítica (bases de datos, archivos sensibles)
- Validación de acceso a otras cuentas o sistemas (lateral movement)
- Escalada de privilegios dentro del sistema
- Verificación de persistencia (agregar backdoors, sólo si está autorizado)
- Análisis de impacto real para el negocio

Toda acción debe estar justificada, registrada y nunca causar daño. Si no está autorizado, no se hace.



Herramientas – Post Explotación

- Metasploit → Módulos de post-explotación
- LinPEAS / WinPEAS → Enumeración post-compromiso
- pspy, sudo -l, getcap, linenum.sh → Para escalar privilegios
- netstat, arp, ifconfig, ip, ping, curl → Enumeración de red
- sqlite3, cat, less, strings → Extracción de info sensible

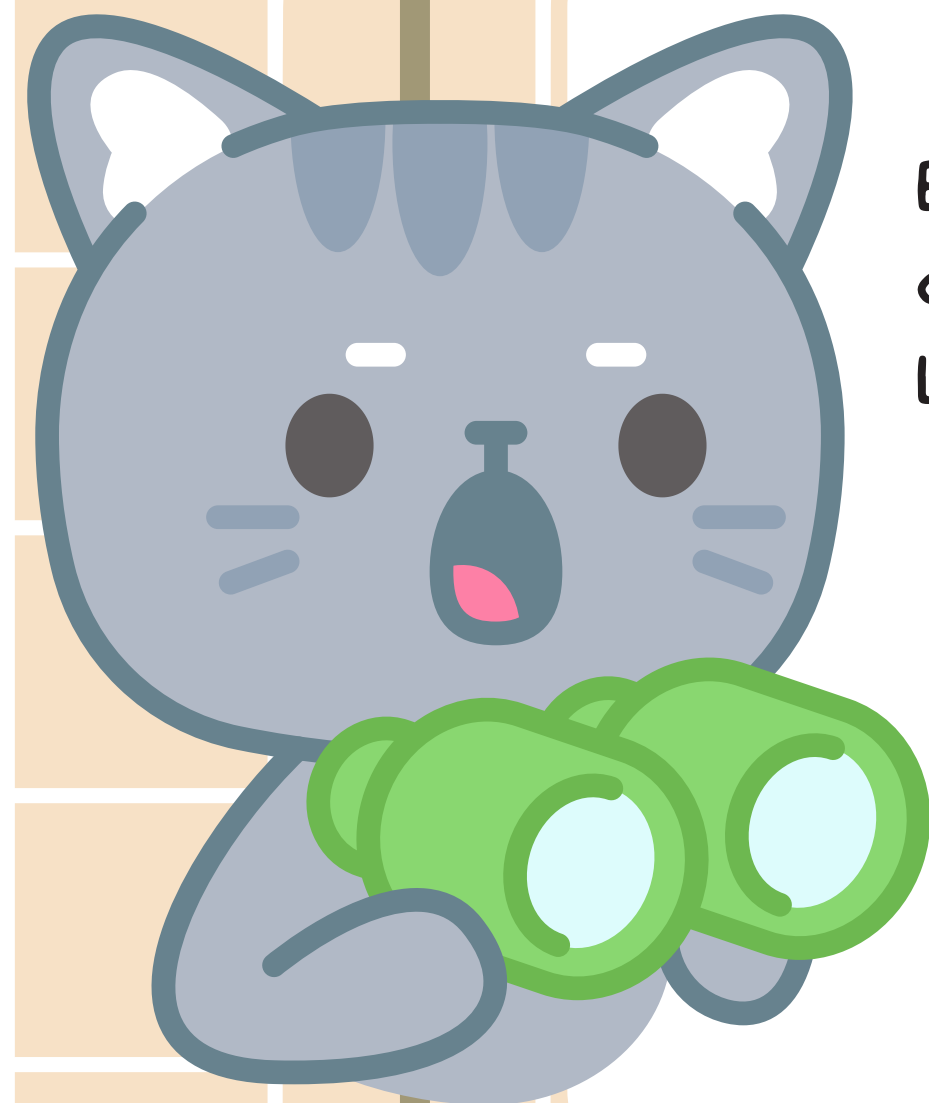
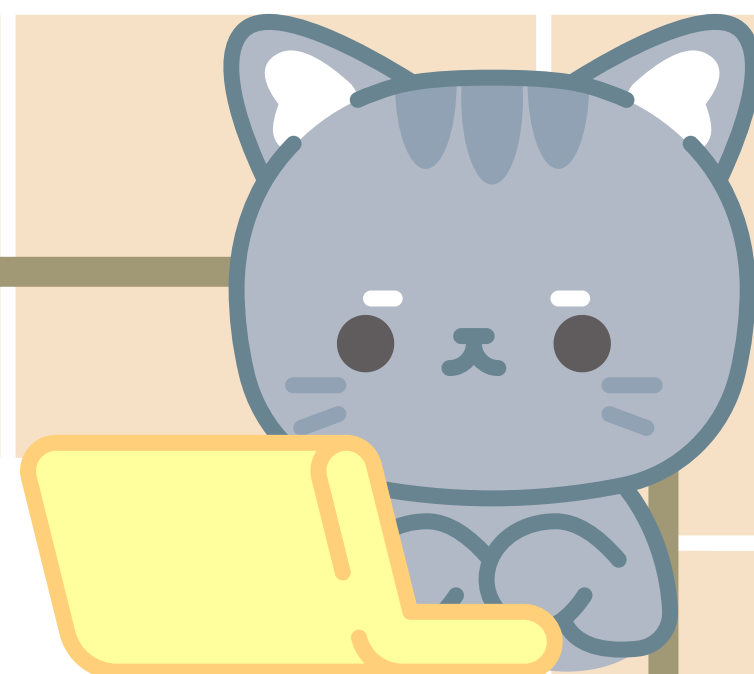


Fase - Documentación

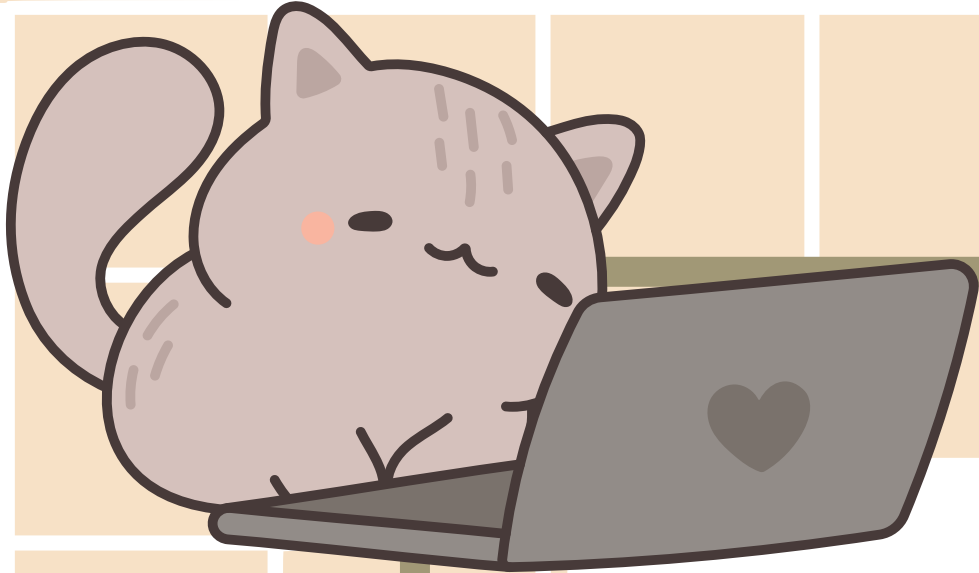
Etapa final, donde se registra todo lo encontrado, probado y demostrado durante el pentest.

Lo mínimo que se espera que contenga el informe técnico:

- Vulnerabilidad detectada.
- Prueba de concepto (PoC).
- Impacto estimado (crítico, alto, medio, bajo).
- Evidencias (capturas de pantalla, request/response, payloads).
- Recomendación técnica para mitigación.
- Riesgo asociado (puede incluir CVSS o riesgo cualitativo).
- Una conclusión del estado general de la seguridad del sistema auditado.

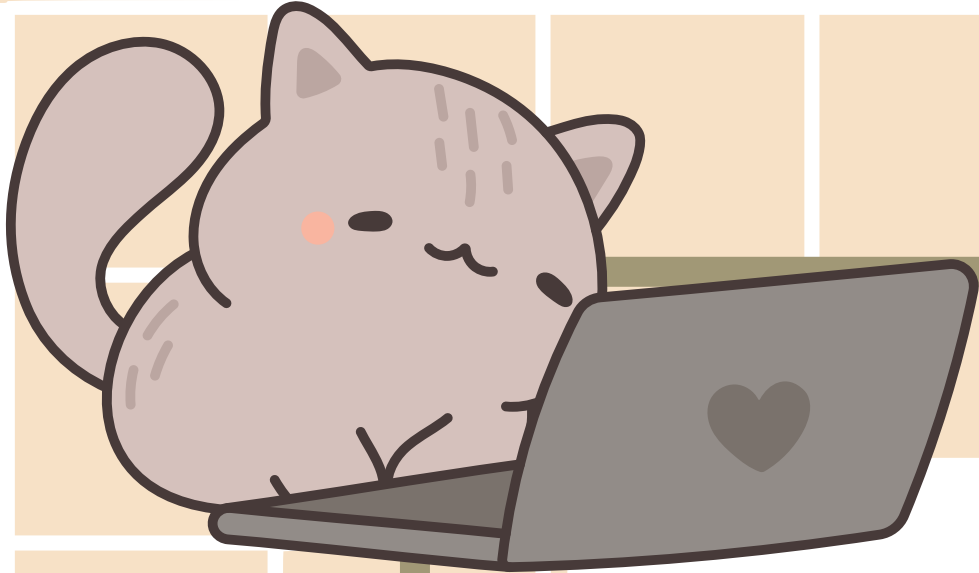


Toda acción debe estar justificada, registrada y nunca causar daño. Si no está autorizado, no se hace.



Herramientas – Documentación

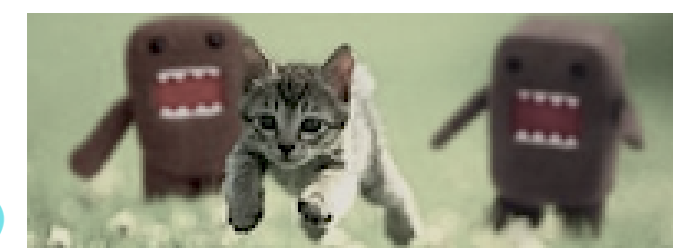




¿Dónde Practicar?

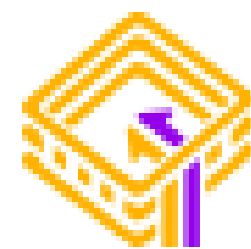


Try
Hack
Me




HackMyVM

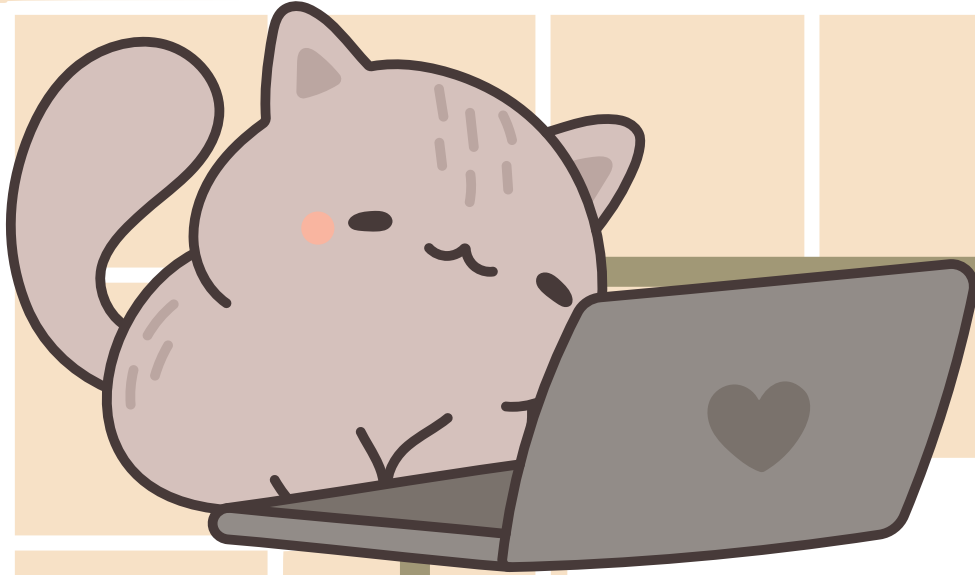
 PortSwigger



VULNHUB
VULNERABLE BY DESIGN

 VULNX





Laboratorios recomendados


WalkingCMS



Autor: El Pingüino de Mario

Dificultad: Fácil

Fecha de creación:
09/04/2024



Basic

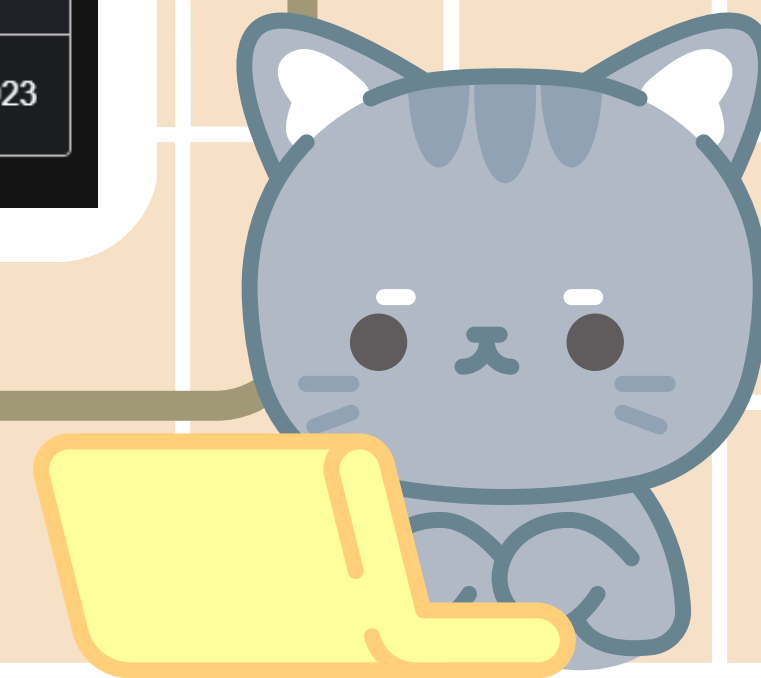
OS: 🐧 Linux

Creator: m0w

Difficulty: 🔵 Low

Release: 26 Oct 2023

VULNEX





¡Gracias!

