Introducción y Contexto

- **Objetivo del Documento:** Presentar una guía práctica y profesional que detalle el **flujo de trabajo óptimo** para la transición de la fase de **Reconocimiento Activo** a la **Explotación Inicial** utilizando las herramientas Nmap y Metasploit Framework (MSF).
- **Enfoque Profesional:** Este recurso está diseñado para analistas *junior*, buscando minimizar errores comunes y ahorrar tiempo mediante la automatización de tareas de recolección de datos y la correcta **importación de información** entre las dos herramientas.
- **Metodología:** El recurso se estructura como un *CheatSheet* avanzado que prioriza la **calidad del dato** y la **integración de herramientas** para una explotación precisa y metodológica.

---

3. Nmap: El Paso Cero (Calidad del Dato)

Esta sección se enfoca en cómo usar Nmap para obtener la información crítica que Metasploit necesita (versión del servicio y SO).

3.1. Comando Maestro y Recolección de Datos

El procedimiento comienza con el escaneo más completo posible. El uso de la opción `-oA` es crucial, ya que genera el archivo **`.gnmap`**, el único formato que Metasploit puede importar directamente a su base de datos.

Comando	Intención Técnica	Resultados Clave para MSF	Procedimiento
`nmap -sS -sV -O -p- -T4 --open <IP> -oA fullscan`	**Escaneo Agresivo y Versátil:** Escaneo SYN (`-sS`), Detección de Versión (`-sV`), Detección de SO (`-O`), todos los puertos (`-p-`). Genera salidas en formato *grepable*(`.gnmap`) .	**Versión de Servicios** (ej. *PostgreSQL 9.3*), **Versión de SO** (ej. *Linux Kernel 3.x*).	Ejecuto este comando una vez que he identificado la dirección IP de la máquina objetivo (por ejemplo, después de un `arp-scan` exitoso) para obtener toda la información necesaria para el *targeting*en MSF.

3.2. Scripts Esenciales para Enumeración Previa

Una vez que el escaneo maestro confirma qué puertos están abiertos, se ejecutan *scripts* específicos para obtener información de *login* o detalles de configuración que ayuden a la explotación:

- **`--script ftp-anon`:** Prueba si el servidor FTP abierto permite el acceso **anónimo** (Ejercicio 7).
- **`--script pgsql-info` / `pgsql-brute`:** Obtiene información detallada del servidor PostgreSQL y prueba credenciales comunes, esencial para el **ataque de fuerza bruta** necesario en el Ejercicio 8.
- **`--script ssh-enumusers`:** Enumeración de usuarios a través del protocolo SSH, intentando identificar nombres de usuario válidos (Relevante para el Ejercicio 6 y 7).

---

4. Flujo de Trabajo y Transición (Nmap → Metasploit)

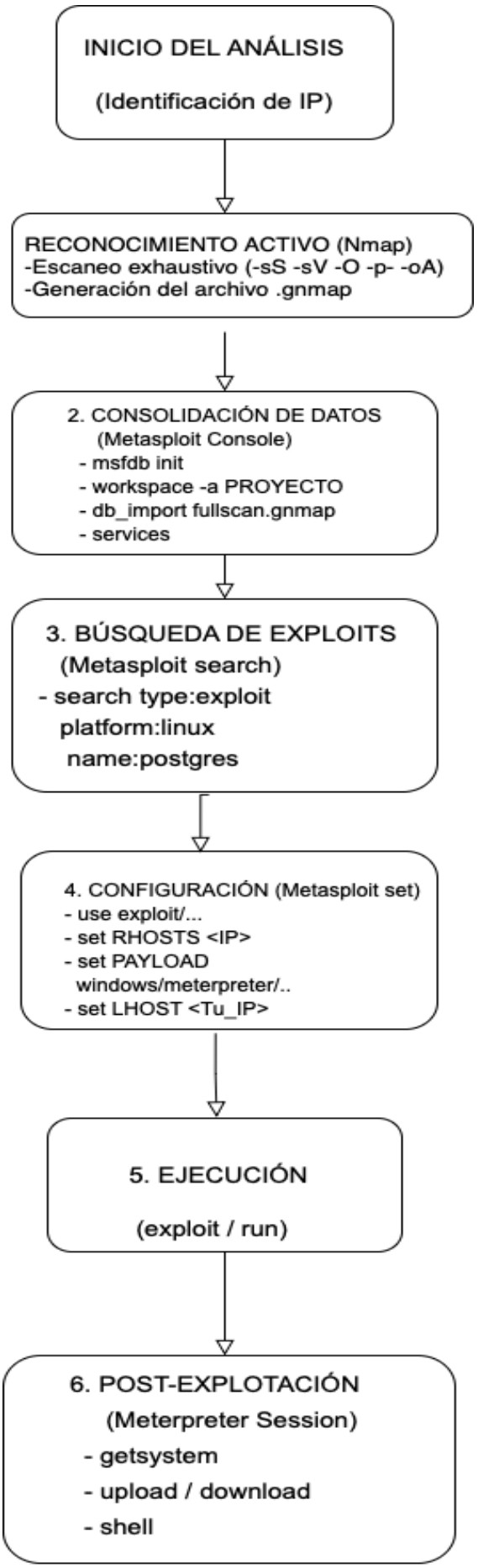
Esta sección explica el procedimiento para importar los datos de Nmap a la base de datos de Metasploit, automatizando el proceso de *targeting*.

4.1. Integración de la Base de Datos

- **Razón:** Evitar errores de *typing* y acelerar la configuración de *exploits*. Al importar los resultados, MSF conoce la IP, puertos y servicios de forma automática.
- **Pasos en `msfconsole`:**
    1. **Iniciar Base de Datos:** `msfdb init` (Asegura que la base de datos PostgreSQL de MSF esté lista).
    2. **Crear Workspace:** `workspace -a [Nombre_Proyecto]` (Procedimiento para aislar los datos de este proyecto, manteniendo la organización).
    3. **Importar Resultados:** `db_import /ruta/a/fullscan.gnmap` (El archivo **`.gnmap`** generado por Nmap se importa. Esto registra el host, los puertos abiertos y las versiones detectadas en la base de datos de MSF).

4.2. Flujo Visual del Proceso

El flujo de trabajo demuestra la **integración metodológica** de ambas herramientas:


INICIO DEL ANÁLISIS
(Identificación de IP)
RECONOCIMIENTO ACTIVO (Nmap)
-Escaneo exhaustivo (-sS -sV -O -p- -oA)
-Generación del archivo .gnmap
2. CONSOLIDACIÓN DE DATOS
(Metasploit Console)
- msfdb init
- workspace -a PROYECTO
- db_import fullscan.gnmap
- services
3. BÚSQUEDA DE EXPLOITS
(Metasploit search)
- search type:exploit
platform:linux
name:postgres
4. CONFIGURACIÓN (Metasploit set)
- use exploit/...
- set RHOSTS <IP>
- set PAYLOAD
windows/meterpreter/..
- set LHOST <Tu_IP>
5. EJECUCIÓN
(exploit / run)
6. POST-EXPLOTACIÓN
(Meterpreter Session)
- getsystem
- upload / download
- shell

[INICIO DEL ANÁLISIS] → **Nmap** (Recolección) → **Importación** (MSF db_import) → **Búsqueda**(MSF search) → **Configuración** (MSF set RHOST/PAYLOAD) → **Explotación** → **[METERPRETER]**

---

5. Profundidad Técnica y Consejos Avanzados

Esta sección demuestra el criterio técnico, la experiencia de uso y la capacidad de ir más allá de la documentación básica.

5.1. Búsqueda Inteligente de Módulos

- **No solo `search`:** Utilizo la indexación de la base de datos de MSF para realizar búsquedas específicas y ahorrar tiempo, en lugar de revisar una lista kilométrica:
  - `search type:exploit platform:linux name:postgres` (Busca solo *exploits* para PostgreSQL en Linux).
  - `search type:auxiliary name:smb_enum` (Busca módulos auxiliares para enumeración SMB).

5.2. Post-Explotación (Meterpreter)

- **El Meterpreter:** En el Ejercicio 9, el *payload* preferido es Meterpreter (una *shell* avanzada) porque no es una *shell* simple. Permite realizar **tareas de post-explotación** sin salir de la sesión:
  - `upload` / `download` de archivos.
  - `migrate` (migrar el proceso a otro más estable para evitar caídas).
  - `getsystem` (Intento automatizado de escalada de privilegios).

5.3. Limitaciones y Criterio del Analista

- **Limitación:** Nmap puede fallar en la detección de versión si hay un *firewall* o un *load balancer*. En estos casos, el criterio técnico exige que la **inspección manual** (usando `telnet` o `netcat`) para leer los *banners* de servicio sea obligatoria.
- **Criterio:** Si Nmap detecta una versión de servicio y Metasploit no tiene un *exploit* directo, demuestro mi capacidad para usar la herramienta como *listener*: se utiliza **`exploit/multi/handler`** en MSF para configurar un *listener* y luego se lanzan *exploits* manuales de **Exploit-DB** o de terceros, manteniendo la sesión de *post-explotación* dentro del *framework*.