Black System

Seguridad Informática

Seguridad Informática

FPTD/2021/019/2080

Metadatos

Pueden definirse como unos datos que describen a otros datos. Por ejemplo en las propiedades de un documento de Office, se puede encontrar información clasificada como Metadatos.
Algunos ejemplos de Metadatos son:

Creador del Documento
Nombre del equipo donde se creo
Sistema Operativo
Una nombre de una impresora.
Una Ruta (Path)

Existen muchas herramientas para la extracción de los METADATOS, la FOCA es una de ellas. Es una herramienta para el S.O Windows, escrita por el consultor de seguridad español “Chema Alonso”.

Otra opción para poder ver los metadatos de los archivos es: https://metashieldclean-up.elevenpaths.com/
El inconveniente de esta opción es que tenemos que ir de un archivo en uno.

nmap

Nmap nos permitirá obtener una gran cantidad de información sobre los equipos de nuestra red, es capaz de escanear qué hosts están levantados, e incluso comprobar si tienen algún puerto abierto, si están filtrando los puertos (tienen un firewall activado), e incluso saber qué sistema operativo está utilizando un determinado objetivo.

- Hay 65535 puertos en cada IP. Para escanear todos los puetos tendríamos que realizar el siguiente escaneo:

nmap ip -p-

- Para que solo nos muestren los puertos que están abiertos:

nmap ip -p- --open

- Otro parámetro para indicarle que vaya más rápido pero que haga más ruido o que vaya más lento y sigiloso es:

nmap ip -p- --open -T5 (con este iria rápido pero ruidoso)

- Para que no aplique diferencial DNS (para ahorrar tiempo en el escaneo)

nmap ip -p- --open -T5 -n

- Para exportar los resultados a un archivo que podamos grepear

nmap ip -p- --open -T5 -n -oG NombreArchivo

Una vez que sabemos que puertos están abiertos vamos a lanzarle unos scripts básicos para saber la versión y servicio que corren en esos puertos y exportarlo a un tipo de archivo Nmap

- nmap -sC -sV -pPuertosOpen(separados por comas) ip -oN
  ejemplo: nmap -sC -sV -p22,80 192.168.1.190 –oN targeted

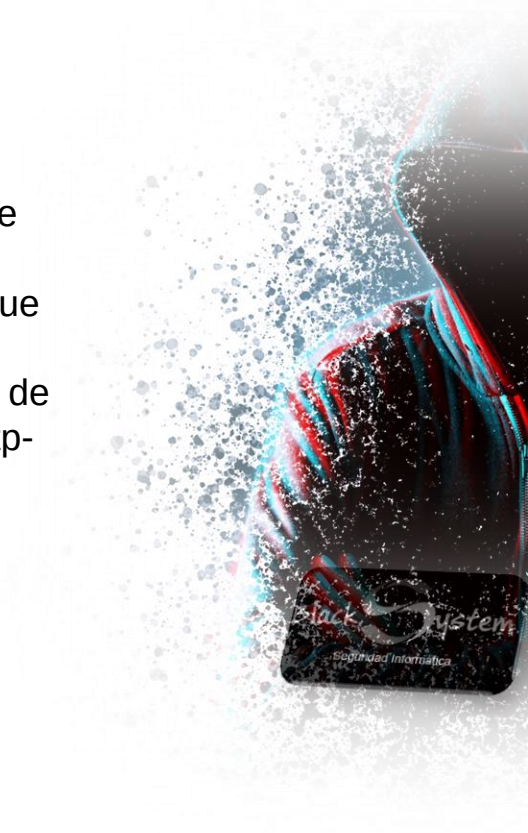
nmap

- Comando para hacer UDP SCAN:
  Nmap –sU IP
- Comando para identificar la versión del sistema operativos del equipo que se esta analizando:
  nmap –O –p(Puerto) IP
- Comando nmap usado para escaneo sigiloso y completo:
  Nmap –sS –O –sV ip

Nmap scripts

Para localizar los scripts de nmap ponemos: locate .nse
Hay 14 tipos de categorías en estos scripts:

- auth: Estos scripts se ocupan de las credenciales de autenticación (o las omiten) en el sistema de destino. Los ejemplos incluyen x11-access, ftp-anony oracle-enum-users.
- broadcast: Los scripts de esta categoría normalmente detectan hosts que no aparecen en la línea de comandos mediante la difusión en la red local. Utilice el argumento de la newtargetssecuencia de comandos para permitir que estas secuencias de comandos agreguen automáticamente los hosts que descubren a la cola de exploración de Nmap.
- brute: Estos scripts utilizan ataques de fuerza bruta para adivinar las credenciales de autenticación de un servidor remoto. Nmap contiene scripts para fuerza bruta en docenas de protocolos, incluidos http-brute, oracle-brute, snmp-brute, etc.
- default: scripts que se ejecutan automáticamente con las opciones -sC o -A
- discovery: scripts que intentan adquirir más información sobre la red de destino
- dos: secuencias de comandos que pueden bloquear la aplicación de destino y, por lo tanto, causar una denegación de servicio al destino
- exploit: scripts que pueden explotar la aplicación de destino
- external: scripts que envían datos a un servidor de terceros a través de la red (whois)
- fuzzer: scripts que envían datos aleatorios no válidos al objetivo para encontrar errores no descubiertos
- intrusive: scripts que pueden hacer que el objetivo falle
- malware: scripts que prueban si el objetivo está infectado por malware o puertas traseras
- safe: scripts que se pueden ejecutar de forma segura, por lo que no bloquearán un servidor
- version: scripts que pueden determinar la versión de la aplicación que se ejecuta en un destino (se ejecutan solo cuando se especifica la opción -sV)
- vuln: scripts que pueden verificar si el objetivo es vulnerable a ataques específicos



Nmap scripts por categoria

Ejercicio ---- listar por pantalla la categoría que pertence cada script de nmap (categories)

Ejemplo para usar un script de nmap por categoria
Nmap -p(numero puerto) ip --script “nombre de la categoría”
Nmap -p80 10.2.0.18 --script “vuln and safe”

- se puede fusionar categorías con and o or

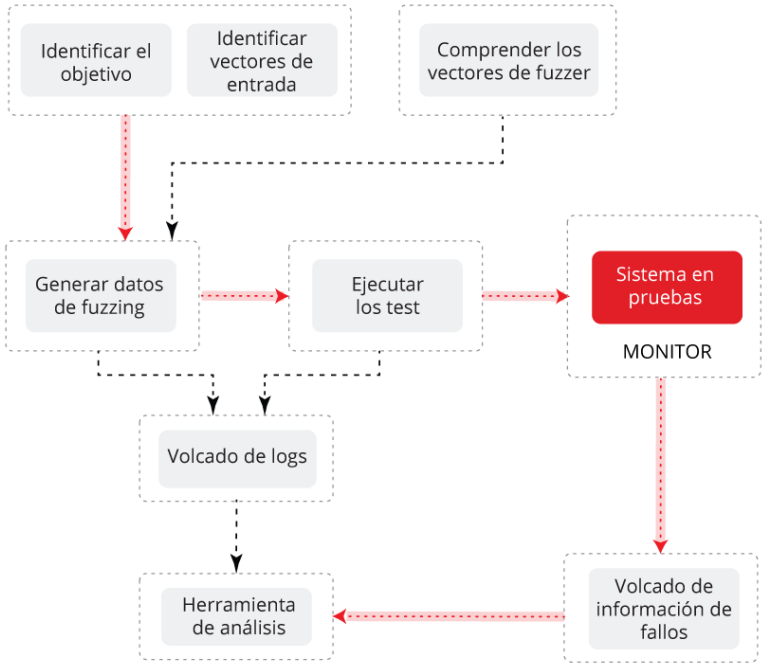
Nmap un solo scripts

Ejemplo para usar un script de nmap
Nmap -p(numero puerto) ip --script “nombre del script”
Nmap -p80 10.2.0.18 –script http-enum

- este script hace parecido a dirb

Fuzzing

Se conoce como fuzzing o técnicas de fuzzing al conjunto de pruebas de caja negra que permiten descubrir errores en los programas o protocolos mediante la introducción de datos al azar, inválidos y malformados. El fin último de estas pruebas es provocar comportamientos inesperados, como fallos que lleguen a hacer que el dispositivo, aplicación o servicio, dejen de funcionar. Este tipo de pruebas permiten, de forma automática o semiautomática, detectar potenciales vulnerabilidades de una forma rápida. Además, en caso de disponer del código fuente, arreglar la vulnerabilidad será mucho más fácil. Las herramientas utilizadas para llevar a cabo estas pruebas se conocen con el nombre de fuzzers.



wfuzz

hidden code

wfuzz -c -hc=404 -w /usr/share/wordlist/disbuster/directory-list-2.3-médium.txt http://10.2.0.19/FUZZ

colores es para que aquí ponga la palabra del diccionario

Si queremos que trabaje con un numero de consultas definidas a la vez le metemos -t:
wfuzz -c -t 400 -hc=404 -w /usr/share/wordlist/disbuster/directory-list-2.3-médium.txt http://10.2.0.19/FUZZ

Si queremos que nos muestre los que tienen estado 200:
wfuzz -c -t 400 -sc=200 -w /usr/share/wordlist/disbuster/directory-list-2.3-médium.txt http://10.2.0.19/FUZZ

Desde wfuzz podríamos hacer un doble fuzzing, para buscar dentro de cada palabra del diccionario extensiones de arvhivos desde otro diccionario que nos crearemos nosotros

- Primero creamos un archivo extensiones.txt y le metemos php, txt, html
- Despues para hacer el doble fuzzing pondríamos lo siguiente:

wfuzz -c -t 400 -hc=404 -w /usr/share/wordlist/disbuster/directory-list-2.3-médium.txt -w extensiones.txt http://10.2.0.19/FUZZ.FUZ2Z

Dirbuster

Es una aplicación con interface gráfica, por lo que es mucho mas amigable.
Solo tenemos que poner dirbuster en la consola
Se abrirá en una ventana y seleccionamos el target que será la ip
Cuantos peticiones queremos lanzar a la vez
Seleccionar el diccionario
Quitar recursivo
Y seleccionar que tipos de archivos queremos encontrar.

Gobuster

Es otra aplicación para realizar fuzzing. Su línea de comando es:
gobuster dir -t 100 -w /usr/share/wordlist/disbuster/directory-list-2.3-médium.txt --url http://10.2.0.19

Dirsearch

No viene instalada por defecto