14 de mayo de 2021
Ficha N° 3 NMAP
CSIRT DE GOBIERNO

Comando de la semana "NMAP"

I. Contexto

Este documento, denominado "comando de la semana", tiene como objetivo ilustrar sobre herramientas que pueden ser de utilidad para el lector, a objeto de ir potenciando las capacidades locales de autochequeo, detección simple de vulnerabilidades que están expuestas a internet en sus activos de información y, a su vez, la obtención de una verificación de la subsanación de aquellas que se les han sido reportadas, facilitando la interacción con el CSIRT de Gobierno. El objetivo no es reemplazar una auditoria de código o evaluación de vulnerabilidades, sino que establecer capacidades básicas de chequeo y obtención de información de manera rápida para temas específicos.

II. Introducción

¿Qué hacer si desde el CSIRT nos llega un ticket señalando que hay problemas de puertos de servicios abiertos (TCP/UDP) y expuestos a internet? ¿Cómo verificamos, una vez que hemos aplicado alguna mitigación y queremos probar si ha tenido efecto, antes de reportarla como problema solucionado al CSIRT o a nuestros auditores internos?

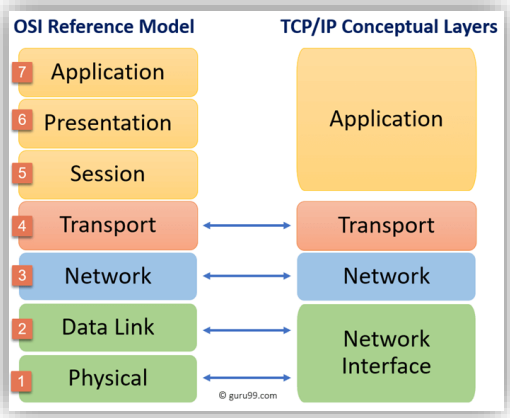
Para este caso existe un comando Linux que nos ayuda a detectar información relevante sobre los puertos TCP o UDP abiertos, entre otros múltiples usos, y tomar decisiones de control de acceso u otras estrategias de mitigación: NMAP.

Pero antes un repaso rápido de algunos conceptos.

IP[1] es el protocolo fundamental para las comunicaciones en Internet, todos los protocolos utilizan IP para transferir datos desde un origen a un destino. IP es un protocolo no orientado a la conexión y basado en el mejor esfuerzo. IP maneja el enrutamiento, la fragmentación y el re-ensamblaje de paquetes grandes cuando sea necesario.

En referencia al modelo OSI los protocolos TCP/IP se homologan de esta manera:

[1] RFC 6864 (2013).



	Concept	Interactive	Async Communication	Remote Access	Resolution	Internal Control	Network Control	Secure Remote Access
Application Layer	Purpose	Browsing Banking Shopping Tunneling	Advertising Personal Official	Tunneling File Transfer Login		Clock Sync	Ping Traceroute	
	Program Interface	Web	E-mail	PuTTY OpenSSH	Bind			
	Communication Protocol	HTTP	SMTP	SSH	DNS	NTP		VPN
Transport Layer		TCP			UDP		ICMP	ESP
Internet Protocol Layer		IP						
Network Access Layer		Ethernet, Wireless, Cable Modem, ISP, Cellular						

Descripción de "UDP": User Data Protocol

- Sin estado y sin conexión
- No garantiza la entrega
- Sin números de secuencia
- Sin bits de bandera
- Sin reconocimientos
- La confiabilidad es responsabilidad del
- aplicación sola
- Rápido debido a la baja sobrecarga
- Usado por: DNS, NTP, SNMP, RIP, DHCP

Voz, audio, video

Descripción de "TCP": Transmission Control Protocol

- Stateful Protocol, las banderas (flags) definen el estado
- Transferencia de datos confiable
- Circuito virtual orientado a la conexión
- Transmisión con búfer
- Re-secuenciación
- Multiplexación
- Transmisión full-duplex eficiente
- Control de flujo
- Usado por: HTTP, FTP, SMTP, BGP, SSH

Los puertos son localizadores numéricos que permiten que los mensajes sean desmultiplexado y entregados al proceso adecuado (servicio). Las conexiones se establecen normalmente mediante puertos conocidos:

- puertos conocidos 1 - 1023
- puertos registrados 1024 – 49151

Algunos de los puertos de servicios que utilizamos comúnmente son:

- FTP = 20, 21, para las transferencias de archivos.
- SSH = 22, para acceso encriptado a las terminales de dispositivos.
- SMTP = 25, para el intercambio de correos electrónicos.

- DNS = 53, para la resolución de nombres.
- TFTP = 69, para transferencia de archivos.
- HTTP = 80, para navegar por internet.
- POP3 = 110, para rescatar los correos electrónicos desde el mail Server.
- BGP = 179, protocolo mediante el cual se intercambia información de encaminamiento entre sistemas autónomos.
- HTTPS = 443, para navegar de manera segura por internet.

Un listado extenso de los puertos de servicios utilizados los podemos encontrar en el sitio web de la IANA:

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Bueno, mucha teoría…ahora lo que nos convoca: NMAP.

Nmap ("Network Mapper") es una utilidad gratuita y de código abierto para el descubrimiento de redes y la auditoría de seguridad. Muchos administradores de sistemas y redes también lo encuentran útil para tareas como el inventario de la red, la gestión de los programas de actualización del servicio y la supervisión del tiempo de actividad del host o del servicio. Nmap utiliza paquetes de IP sin procesar de formas novedosas para determinar qué hosts están disponibles en la red, qué servicios (nombre y versión de la aplicación) ofrecen esos hosts, qué sistemas operativos (y versiones de SO) están ejecutando, qué tipo de filtros de paquetes / firewalls están en uso y decenas de otras características. Fue diseñado para escanear rápidamente redes grandes, pero funciona bien contra hosts únicos. Nmap se ejecuta en todos los principales sistemas operativos de computadoras, y los paquetes binarios oficiales están disponibles para Linux, Windows y Mac OS X.

Algunas características de Nmap son:

Flexible: admite docenas de técnicas avanzadas para trazar redes llenas de filtros IP, cortafuegos, enrutadores y otros obstáculos. Esto incluye muchos mecanismos de escaneo de puertos (tanto TCP como UDP), detección de SO, detección de versiones, barridos de ping y más. Consulte la página de documentación.

Potente: Nmap se ha utilizado para escanear enormes redes de literalmente cientos de miles de máquinas.

Portátil: la mayoría de los sistemas operativos son compatibles, incluidos Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga y más.

Fácil: si bien Nmap ofrece un amplio conjunto de funciones avanzadas para usuarios avanzados, puede comenzar tan simplemente como "nmap -v -A www.example.com". Tanto la línea de comandos tradicional como las versiones gráficas (GUI) están disponibles para adaptarse a sus preferencias. Los binarios están disponibles para aquellos que no deseen compilar Nmap desde la fuente.

Gratis: El objetivo principal del Proyecto Nmap es ayudar a que Internet sea un poco más seguro y proporcionar a los administradores / auditores / piratas informáticos una herramienta avanzada para explorar sus redes. Nmap está disponible para descarga gratuita y también viene con el código fuente completo que se puede modificar y redistribuir según los términos de la licencia.

Bien documentado: se ha realizado un esfuerzo significativo en páginas de manual completas y actualizadas, documentos técnicos, tutoriales e incluso un libro completo. Encuéntrelos en varios idiomas aquí.

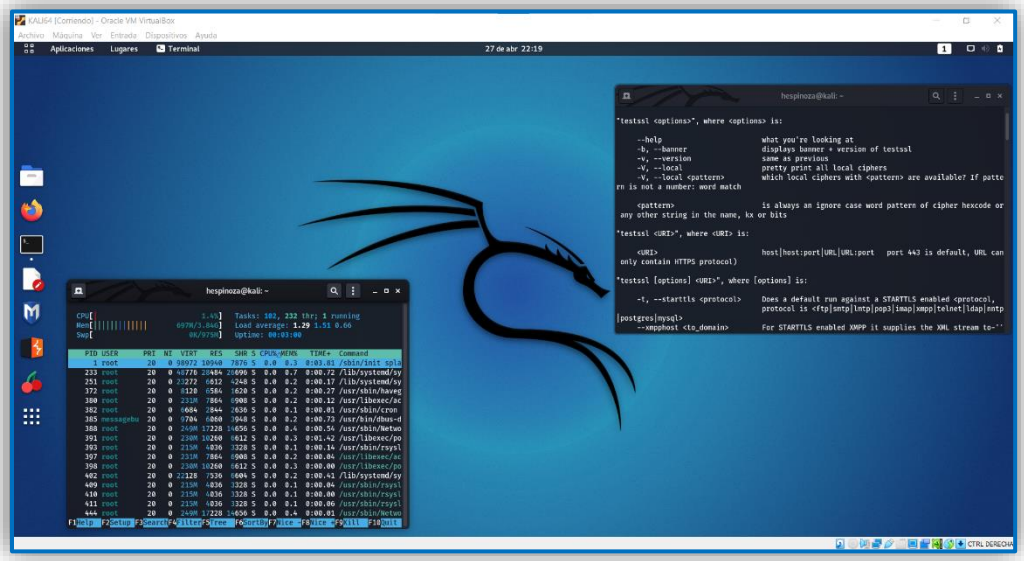
Compatible: si bien Nmap no tiene garantía, está bien respaldado por una vibrante comunidad de desarrolladores y usuarios. La mayor parte de esta interacción ocurre en las listas de correo de Nmap. La mayoría de los informes de errores y las preguntas deben enviarse a la lista nmap-dev, pero solo después de leer las pautas.

Fuente: http://nmap.org/

III. Paso a Paso

PASO 1: Un entorno adecuado para trabajar.

Primero debe contar con una distribución de Kali[2] Linux funcionando ya sea en una máquina física o en una máquina virtual[3][4].



PASO 2: Instalar el comando.

Una vez que se cuenta con este sistema operativo de manera funcional podemos instalar el comando "mmap"; en general este ya viene preinstalado en la distribución KALI, pero si no fuere así puede instalarlo con el siguiente comando:

```
sudo apt-get install nmap
```

PASO3: Verificar su instalación.

Una vez que se instalado podemos explorar las múltiples opciones que ofrece para su ejecución:

```
└─# nmap -h
Nmap 7.91 ( https://nmap.org )
```

[2] https://www.kali.org/downloads/

[3] https://my.vmware.com/en/web/vmware/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0

[4] https://www.virtualbox.org/wiki/Downloads

```
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
```

```
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
        directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
        <Lua scripts> is a comma-separated list of script-files or
        script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
     probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
  --min-rate <number>: Send packets no slower than <number> per second
  --max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
  -S <IP_Address>: Spoof source address
  -e <iface>: Use specified interface
  -g/--source-port <portnum>: Use given port number
  --proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
  --data <hex string>: Append a custom payload to sent packets
  --data-string <string>: Append a custom ASCII string to sent packets
  --data-length <num>: Append random data to sent packets
  --ip-options <options>: Send packets with specified ip options
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
```

```
  -v: Increase verbosity level (use -vv or more for greater effect)
  -d: Increase debugging level (use -dd or more for greater effect)
  --reason: Display the reason a port is in a particular state
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Nmap.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
```

Paso 4: Ponerlo en marcha para verificar nuestra infraestructura.

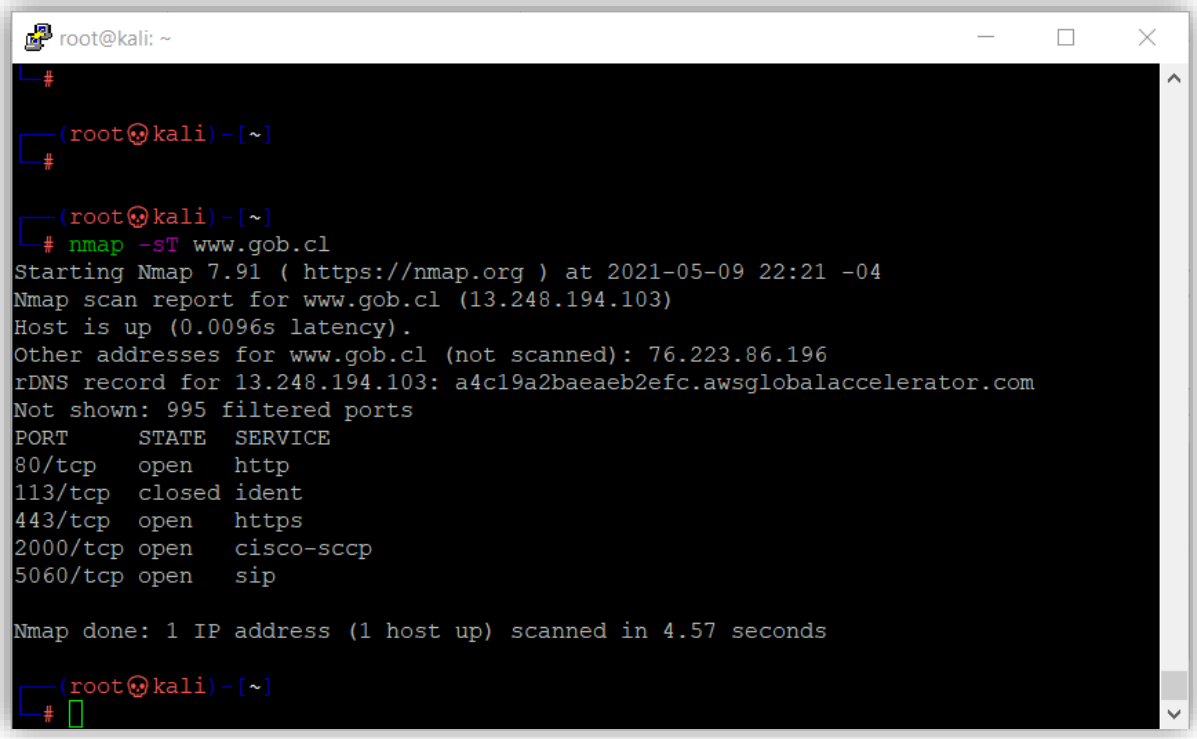
Como se ve un fragmento de reporte en una consola KALI después de la ejecución más simple:

```
EJEMPLOS:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
```

Vista Parcia de un ejemplo:

Ejecución del comando:

#nmap –sT www.gob.cl



El resultado de este comando puede ser usado como evidencia de verificación para indicar que se han subsanado los problemas reportados por CSIRT.

Estudie las múltiples opciones que tiene el comando para obtener resultados específicos o redirigir la salida de este hacia otros formatos de archivo, para su inclusión en informes posteriores.

Algunos ejemplos de uso complementarios [reemplace los sitios de ejemplo por el suyo] son:

- Usando NMAP para encontrar vulnerabilidades.
  - nmap -sV --script=vulscan/vulscan.nse www.example.com

Para el siguiente ejemplo instale previamente el script siguiente:

#cd /usr/share/nmap/scripts/
#git clone https://github.com/vulnersCom/nmap-vulners.git

- nmap --script nmap-vulners -sV www.example.com

En caso de cualquier inquietud no dudes en consultarnos a soc-csirt@interior.gob.cl.