

Explorando SNMP y Técnicas de Enumeración en Seguridad Informática

¿Qué es SNMP y por qué es importante?

Imagina que tu red de internet en casa fuera una ciudad llena de dispositivos como computadoras, impresoras y cámaras de seguridad, todos trabajando juntos para que todo funcione de maravilla. Ahora, imagina que hay un sistema que te permite monitorear todo lo que hacen y, si es necesario, enviar órdenes para cambiar su comportamiento. Ese sistema existe y se llama **SNMP** (Protocolo Simple de Gestión de Redes). Aunque no lo veas, es una pieza clave para que las redes modernas funcionen correctamente.

Pero, ¿qué pasaría si alguien no autorizado lograra acceder a este sistema? Podría controlar tus dispositivos, ver información privada e incluso interrumpir tu conexión. Vamos a aprender cómo funciona este protocolo, cómo lo usan las redes, y por qué debemos protegerlo.

¿Cómo funciona SNMP?

Para entender SNMP, piensa en un maestro (el administrador de red) y muchos alumnos (los dispositivos conectados). El maestro les hace preguntas como:

- ¿Qué tareas estás haciendo?
- ¿Cuánto espacio tienes disponible?
- ¿Tienes algún problema?

Los alumnos responden, y el maestro puede incluso pedirles que cambien su comportamiento, como cerrar una puerta (desactivar un puerto) o limpiar su escritorio (liberar memoria). Este intercambio de información ocurre a través de un lenguaje llamado **SNMP**.

Aquí tienes un resumen visual de cómo funciona:

1. **Gestor (manager):** Es el cerebro que monitorea y controla todo.
2. **Agentes:** Son los ojos y oídos del gestor en cada dispositivo de la red.
3. **MIB (Base de Información de Gestión):** Es como un diccionario donde cada dispositivo almacena su información importante.

Los datos que maneja SNMP

Los dispositivos gestionados envían datos al gestor sobre cosas como:

- Usuarios conectados.

- programas instalados.
- puertos abiertos (como puertas de entrada al dispositivo).

Estos datos se organizan en una tabla jerárquica (el MIB), y el administrador puede usarlos para entender qué está pasando en la red.

SNMP y la Seguridad

Por defecto, SNMP utiliza los puertos **161 y 162**. Pero aquí viene el problema: no siempre está configurado para ser seguro. Dependiendo de la versión que use una red, los riesgos varían. Vamos a ver las versiones:

SNMPv1 (la versión más antigua y débil)

- **Cadenas de comunidad en texto plano:** Es como si la contraseña para acceder a los dispositivos fuera "1234".
- Estas cadenas tienen nombres predefinidos como "public" (para leer datos) y "private" (para cambiar configuraciones).
- Cualquier persona que capture el tráfico de la red puede verlas y usarlas.

Ejemplo: Un hacker puede recopilar información sobre qué dispositivos hay en tu red y modificar configuraciones importantes, como bloquear el acceso a internet. ¡Imagina el caos si esto pasa en una empresa o banco!

SNMPv2 (mejor, pero no perfecto)

Esta versión mejoró el rendimiento y la cantidad de datos que se pueden gestionar, pero la seguridad sigue siendo limitada. Por eso, no se adoptó de forma masiva.

SNMPv3 (el más seguro)

Aquí las cosas mejoran mucho:

- Las contraseñas y datos están cifrados, lo que significa que nadie puede leerlos si intercepta el tráfico.
- Incluye autenticación para asegurar que solo los usuarios autorizados puedan acceder.

A pesar de sus ventajas, SNMPv3 no siempre está habilitado, y muchas redes todavía usan versiones antiguas por desconocimiento o falta de actualización.

¿Por qué deberías preocuparte?

Un atacante que explote SNMP puede:

1. **Espiar:** Ver qué dispositivos están conectados, qué hacen y qué información manejan.
2. **Controlar:** Cambiar configuraciones, bloquear servicios o incluso apagar dispositivos.

3. Interrumpir: Si el SNMP está mal configurado, podría paralizar toda la red.

Cómo proteger tu red

1. Actualiza a SNMPv3: Si tienes dispositivos antiguos, revisa si puedes actualizar el protocolo.

2. Usa contraseñas seguras: Cambia las cadenas de comunidad predeterminadas como "public" y "private".

3. Filtra los accesos: Configura tu red para que solo dispositivos específicos puedan comunicarse con SNMP.

4. Monitorea el tráfico: Si detectas actividad sospechosa en los puertos 161 o 162, investiga inmediatamente.

Explorando VyOS: Tu Primer Laboratorio de Redes

Este software es como un todo terreno para configurar redes: puede ser un enrutador, un firewall o incluso manejar conexiones VPN. Si tienes curiosidad por entender cómo se gestiona una red de manera profesional, ¡este es el lugar para empezar!

¿Qué es VyOS y por qué es importante?

VyOS es un sistema operativo basado en Debian GNU/Linux, pero diseñado específicamente para redes. Es como un administrador de tráfico en una ciudad: organiza, dirige y protege los datos que circulan por tu red. A diferencia de otros sistemas comerciales (que suelen ser costosos), VyOS es gratuito y de código abierto, lo que significa que puedes personalizarlo a tu gusto.

Características principales de VyOS:

- **Control total desde la línea de comandos (CLI):** Si te gusta ser detallista, esto te encantará.
- **Soporte para protocolos avanzados de enrutamiento:** Incluye BGP y OSPF, usados en redes empresariales.
- **Gestión de VPN:** Compatible con IPSec, OpenVPN y WireGuard.
- **Firewall avanzado:** Puedes establecer reglas para bloquear o permitir tráfico.
- **Control de tráfico (QoS):** Ideal para priorizar qué datos son más importantes en tu red.

Se utiliza tanto en empresas como en entornos de aprendizaje, ya que es ideal para experimentar en redes virtuales.

Preparando el laboratorio

Primero, necesitamos instalar VyOS en una máquina virtual. Aquí están los pasos básicos:

Descarga la ISO oficial:

Ve al siguiente enlace:

1. [VyOS Nightly Builds](#)

Consulta la documentación:

Si tienes dudas, aquí puedes encontrar guías rápidas:

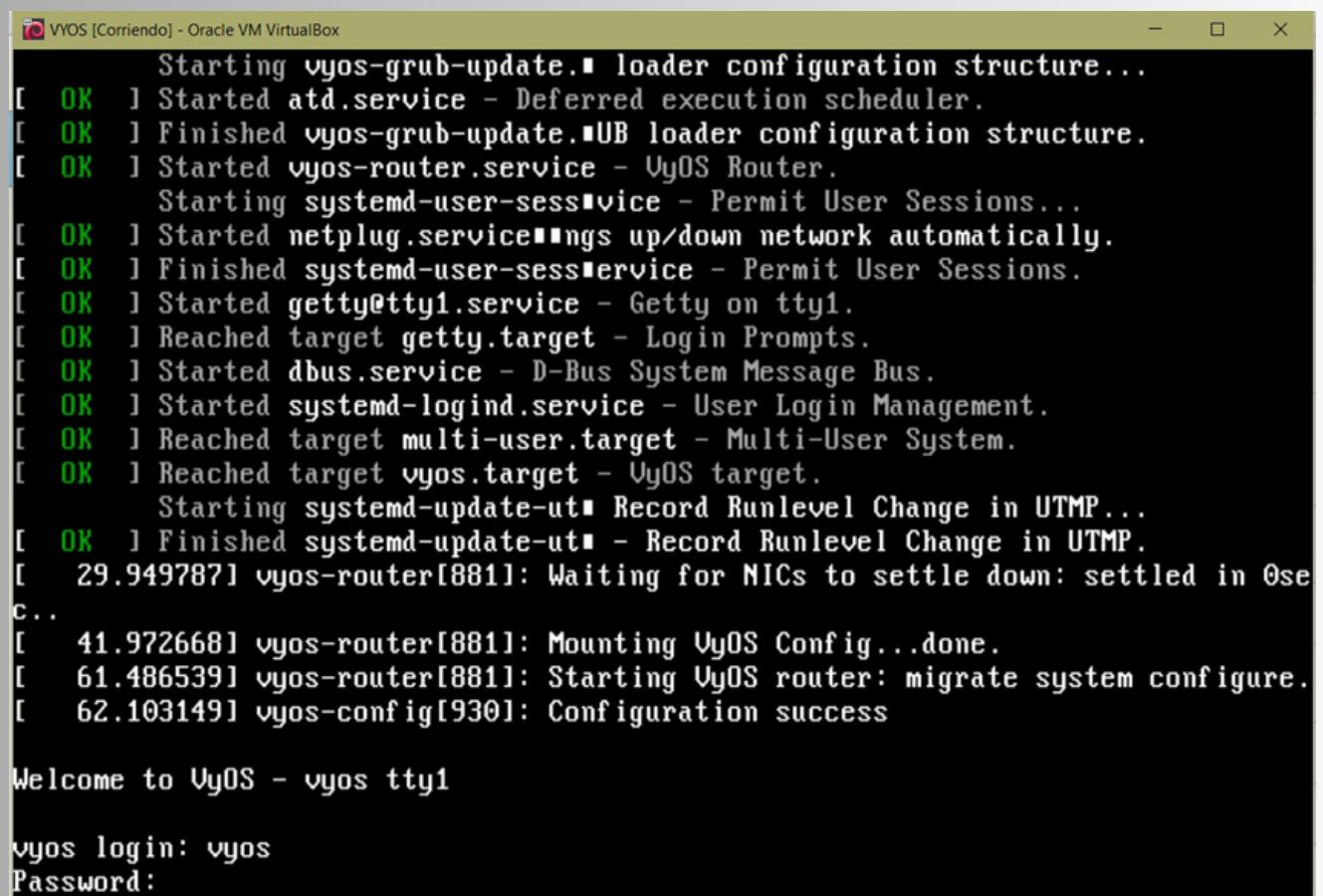
2. [Documentación oficial de VyOS](#)

3. Credenciales por defecto:

• Usuario: **vyos**

• Contraseña: **vyos**

El primer inicio de **Vyos** se verá como se muestra a continuación:

A screenshot of a terminal window titled "VyOS [Corriendo] - Oracle VM VirtualBox". The terminal shows the boot sequence of a VyOS virtual machine. It starts with "Starting vyos-grub-update. loader configuration structure...", followed by "Started atd.service - Deferred execution scheduler.", "Finished vyos-grub-update. UB loader configuration structure.", "Started vyos-router.service - VyOS Router.", "Starting systemd-user-session.service - Permit User Sessions...", "Started netplug.service - Brings up/down network automatically.", "Finished systemd-user-session.service - Permit User Sessions.", "Started getty@tty1.service - Getty on tty1.", "Reached target getty.target - Login Prompts.", "Started dbus.service - D-Bus System Message Bus.", "Started systemd-logind.service - User Login Management.", "Reached target multi-user.target - Multi-User System.", "Reached target vyos.target - VyOS target.", "Starting systemd-update-utmp - Record Runlevel Change in UTMP...", "Finished systemd-update-utmp - Record Runlevel Change in UTMP.", "29.949787] vyos-router[881]: Waiting for NICs to settle down: settled in 0 seconds.", "41.972668] vyos-router[881]: Mounting VyOS Config...done.", "61.486539] vyos-router[881]: Starting VyOS router: migrate system configure.", "62.103149] vyos-config[930]: Configuration success". The prompt "Welcome to VyOS - vyos tty1" is shown, followed by "vyos login: vyos" and "Password:".

```
VyOS [Corriendo] - Oracle VM VirtualBox
Starting vyos-grub-update. loader configuration structure...
[ OK ] Started atd.service - Deferred execution scheduler.
[ OK ] Finished vyos-grub-update. UB loader configuration structure.
[ OK ] Started vyos-router.service - VyOS Router.
Starting systemd-user-session.service - Permit User Sessions...
[ OK ] Started netplug.service - Brings up/down network automatically.
[ OK ] Finished systemd-user-session.service - Permit User Sessions.
[ OK ] Started getty@tty1.service - Getty on tty1.
[ OK ] Reached target getty.target - Login Prompts.
[ OK ] Started dbus.service - D-Bus System Message Bus.
[ OK ] Started systemd-logind.service - User Login Management.
[ OK ] Reached target multi-user.target - Multi-User System.
[ OK ] Reached target vyos.target - VyOS target.
Starting systemd-update-utmp - Record Runlevel Change in UTMP...
[ OK ] Finished systemd-update-utmp - Record Runlevel Change in UTMP.
[ 29.949787] vyos-router[881]: Waiting for NICs to settle down: settled in 0 seconds.
[ 41.972668] vyos-router[881]: Mounting VyOS Config...done.
[ 61.486539] vyos-router[881]: Starting VyOS router: migrate system configure.
[ 62.103149] vyos-config[930]: Configuration success

Welcome to VyOS - vyos tty1

vyos login: vyos
Password:
```

Después procederemos a la instalación:

install image

You can change this banner using "set system login banner post-login" command.

VyOS is a free software distribution that includes multiple components, you can check individual component licenses under /usr/share/doc/*/copyright

```
---  
WARNING: This VyOS system is not a stable long-term support version and  
         is not intended for production use.
```

```
vyos@vyos:~$ install image
```

Welcome to VyOS installation!

This command will install VyOS to your permanent storage.

Would you like to continue? [y/N] _

yes

```
---  
WARNING: This VyOS system is not a stable long-term support version and  
         is not intended for production use.
```

```
vyos@vyos:~$ install image
```

Welcome to VyOS installation!

This command will install VyOS to your permanent storage.

Would you like to continue? [y/N] y

What would you like to name this image? (Default: 1.5-rolling-202412160007)default

default

Please enter a password for the "vyos" user:

Please confirm password for the "vyos" user:

What console should be used by default? (K: KVM, S: Serial)? (Default: K) _

Creamos una password, en mi caso he vuelto a poner "vyos" (No recomendable fuera de un lab)

Y pulsamos **k** si queremos que se instale la consola por defecto.

```
DISKS Found  
The following disks were found:  
Drive: /dev/sda (20.0 GB)  
Which one should be used for installation? (Default: /dev/sda)  
Installation will delete all data on the drive. Continue? [y/N] y_
```

Continuamos pulsando **yes**

```
Would you like to use all the free space on the drive? [Y/n] y
```

Creating partition table...

The following config files are available for boot:

1: /opt/vyatta/etc/config/config.boot

2: /opt/vyatta/etc/config.boot.default

Which file would you like as boot config? (Default: 1)

De nuevo "Sí a todo" y por último pulsamos **1**

Comenzará la instalación de la imagen y nos pedirá reiniciar. Pulsamos **reboot** y tras el reinicio ya tendremos lista la imagen, ahora, a configurarla.

Iniciamos con el Usuario: vyos Password: vyos

```
configure
```

```
set interface ethernet eth0 address 192.168.100.145/24
```

```
-----  
eth0      192.168.100.1/24  08:00:27:b1:d3:8e  default  1500  u/u  
lo        127.0.0.1/8         00:00:00:00:00:00  default  65536  u/u  
          ::1/128  
vyos@vyos:~$ configure  
[edit]  
vyos@vyos# set interface ethernet eth0 address 192.168.100.145/24_
```

```
commit
```

```
save
```

```
exit
```

```
reboot
```

De nuevo `show interfaces` para comprobar los cambios:

```
---  
WARNING: This VyOS system is not a stable long-term support version and  
         is not intended for production use.  
vyos@vyos:~$ show interfaces  
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down  
Interface  IP Address      MAC                VRF          MTU   S/L   Desc  
ption  
-----  
eth0      192.168.100.1/24  08:00:27:b1:d3:8e  default      1500  u/u  
          192.168.100.145/24  
lo        127.0.0.1/8      00:00:00:00:00:00  default      65536  u/u  
          ::1/128  
vyos@vyos:~$
```

Configurando el servicio SNMP en VyOS

¡Hola a todos! Hoy vamos a aprender a configurar el servicio **SNMP** en VyOS. Esto nos permitirá monitorear nuestra red y administrar dispositivos como routers o switches desde una ubicación central. Imaginen que estamos configurando una cámara que vigila el comportamiento de nuestra red. Así de importante es SNMP.

Paso 1: Entrar al modo de configuración

Lo primero que necesitamos hacer es ingresar al modo de configuración de VyOS. Para ello, escribimos:

```
configure
```

Esto nos permitirá realizar cambios en el sistema. ¡Vamos al siguiente paso!

Paso 2: Configurar el servicio SNMP

Para que SNMP funcione, necesitamos establecer una **clave de comunidad (community string)**. Esta clave actúa como una contraseña que permite que los administradores obtengan información o incluso modifiquen la configuración del dispositivo.

```
set service snmp community [clave-de-comunidad] authorization [nivel-de-autorización]
```

- **[clave-de-comunidad]**: Aquí definimos un nombre que sirve como contraseña, por ejemplo, `iloveyou`.

- **[nivel-de-autorización]**:

- `ro`: Solo lectura (permite ver información, pero no modificar nada).

- `rw`: Lectura y escritura (permite ver y también realizar cambios).

Ejemplo práctico:

Supongamos que queremos configurar una clave de comunidad básica:

```
set service snmp community iloveyou authorization ro
```

Esto crea una clave llamada `iloveyou` con permiso de **solo lectura**. Si queremos permitir cambios, usamos:

```
set service snmp community iloveyou authorization rw
```

Nota sobre la clave de comunidad

Elegí `iloveyou` como clave de comunidad porque es un ejemplo clásico de malas prácticas. Es una de las primeras claves que aparecen en los diccionarios de fuerza bruta como "rockyou.txt". **Nunca usen claves fáciles o predecibles en entornos reales.**

Por ejemplo, en una red real, podrían usar algo como:

```
redSNMP2025_secure!
```

¿Por qué? Porque es más difícil de adivinar y mejora la seguridad de nuestra configuración.

Paso 3: Guardar y aplicar los cambios

Después de configurar la clave, siempre debemos guardar y aplicar los cambios. Para ello, escribimos:

```
commit
```

```
save
```

```
exit
```

Y listo, nuestro servicio SNMP ya estará funcionando.

Ejemplo adicional: Crear un usuario en VyOS

Ahora, les enseñaré cómo agregar un nuevo usuario al sistema. Esto puede ser útil si queremos que otra persona tenga acceso con su propia cuenta.

Ingresar al modo de configuración:

Como antes, empezamos con:

1. `configure`

Agregar un usuario con contraseña:

2. Usamos el siguiente comando:

```
set system login user [nombre-de-usuario] authentication plaintext-password  
[contraseña]
```

Ejemplo práctico:

Supongamos que queremos crear un usuario llamado `sergio` con la contraseña `sergiomc` :

```
set system login user sergio authentication plaintext-password sergiomc
```

1. Aplicar y guardar los cambios:

Una vez definido el usuario, guardamos y aplicamos:

```
commit save exit
```

1. Reiniciar el sistema:

Finalmente, reiniciamos VyOS para asegurarnos de que los cambios tomen efecto:

```
reboot
```

Enumeración SNMP con Kali Linux

Utilizaremos herramientas integradas de **Kali Linux** para investigar y extraer información de un sistema en red, en este caso, nuestro servidor **VyOS** configurado previamente.

SNMP (Protocolo Simple de Administración de Red) es un estándar para monitorear y administrar dispositivos en redes. Usaremos dos cadenas de comunidad: una de solo lectura (**ro**) para visualizar datos y otra de lectura-escritura (**rw**) para modificar información. Vamos paso a paso.

Parte 1: Escaneo con Nmap

Objetivo

Verificar si el servicio SNMP está disponible y ejecutándose en el sistema destino.

Escaneo de puertos UDP

Usaremos **nmap**, una herramienta de escaneo de red. El comando es:

```
nmap -sU -p161,162 192.168.100.145
```

```
(root@kali)-[/home/yorha2b]
# nmap -sU -p161,162 192.168.100.145
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-11 17:37 CET
Nmap scan report for 192.168.100.145
Host is up (0.0028s latency).

PORT      STATE  SERVICE
161/udp    open   snmp
162/udp    closed snmptrap
MAC Address: 08:00:27:B1:D3:8E (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
```

Explicación del comando:

- **-sU**: Escaneo de puertos UDP, necesario porque SNMP usa este protocolo.
- **-p161,162**: Especificamos los puertos de SNMP. El puerto **161** es para consultas y el **162** para notificaciones.

Resultado esperado:

Nmap debería mostrar que el puerto **161** está abierto en el dispositivo VyOS.

¿Por qué no usamos TCP? SNMP funciona exclusivamente sobre UDP por ser más ligero y rápido. Sin embargo, carece de confirmación de entrega, lo que lo hace menos seguro.

Escaneo de rango de red

Si queremos encontrar dispositivos en toda una red, usamos:

```
nmap -sU --top-ports 200 -Pn 192.168.100.1/24
```

```
Nmap scan report for 192.168.100.145
Host is up (0.00070s latency).
Not shown: 194 closed udp ports (port-unreach)
PORT      STATE  SERVICE
123/udp    open   ntp
161/udp    open   snmp
443/udp    open|filtered https
6001/udp   open|filtered X11:1
49182/udp  open|filtered unknown
49184/udp  open|filtered unknown
MAC Address: 08:00:27:B1:D3:8E (Oracle VirtualBox virtual NIC)
```

Explicación del comando:

- **--top-ports 200**: Escanea los 200 puertos más utilizados.
- **-Pn**: Ignora la comprobación de si los hosts están activos (útil si el firewall bloquea los pings).
- **192.168.100.1/24**: Especificamos un rango de IP para buscar otros dispositivos activos.

Reflexión: Esto simula una fase de reconocimiento en un entorno real. Una buena práctica es identificar posibles víctimas antes de enfocarnos en una.

Parte 2: Exploración con Snmpwalk

Objetivo

Extraer información del dispositivo utilizando la cadena de comunidad.

Paso 1: Comando básico

```
snmpwalk -v1 -c iloveyou 192.168.100.145
```

Explicación del comando:

- **-v1**: Usamos SNMP versión 1.
- **-c iloveyou**: Especificamos la cadena de comunidad configurada.
- **192.168.100.145**: IP del objetivo.

```

(root@kali)-[/home/yorha2b]
# snmpwalk -v1 -c iloveyou 192.168.100.145
Created directory: /var/lib/snmp/cert_indexes
iso.3.6.1.2.1.1.1.0 = STRING: "VyOS 1.5-rolling-202412160007"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.44641
iso.3.6.1.2.1.1.3.0 = Timeticks: (115269) 0:19:12.69
iso.3.6.1.2.1.1.4.0 = STRING: "root"
iso.3.6.1.2.1.1.5.0 = STRING: "vyos"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown"
iso.3.6.1.2.1.1.7.0 = INTEGER: 14
iso.3.6.1.2.1.1.8.0 = Timeticks: (8) 0:00:00.08
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (7) 0:00:00.07
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (8) 0:00:00.08
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (8) 0:00:00.08
iso.3.6.1.2.1.2.1.0 = INTEGER: 3

```

Muestra información estructurada en forma de árbol, como datos del sistema, interfaces de red, etc.

Parte 3: Modificar datos con Snmpset

Objetivo

Demostrar que una cadena de comunidad con permisos de **lectura y escritura (rw)** permite modificar configuraciones del sistema.

Paso 1: Cambiar el nombre del host

Ejecutamos el siguiente comando:

```
snmpset -v1 -c iloveyou 192.168.100.145 iso.3.6.1.2.1.1.5.0 s Hacked
```

```

(root@kali)-[/home/yorha2b]
# snmpset -v1 -c iloveyou 192.168.100.145 iso.3.6.1.2.1.1.5.0 s Hacked
iso.3.6.1.2.1.1.5.0 = STRING: "Hacked"

(root@kali)-[/home/yorha2b]
#

```

Explicación del comando:

- **-v1** Usamos SNMP versión 1.
- **-c iloveyou** Usamos la cadena de comunidad con permisos de escritura.
- **iso.3.6.1.2.1.1.5.0**: El **OID** que apunta al nombre del host.
- **s**: Especifica que el valor será una cadena de texto.
- **Hacked**: El nuevo nombre del host.

El nombre del host debería cambiar a **Hacked**. Para verificarlo, ejecutamos de nuevo:

```
snmpwalk -v1 -c iloveyou 192.168.100.145
```

```
(root@kali)-[/home/yorha2b]
# snmpwalk -v1 -c iloveyou 192.168.100.145
iso.3.6.1.2.1.1.1.0 = STRING: "VyOS 1.5-rolling-202412160007"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.44641
iso.3.6.1.2.1.1.3.0 = Timeticks: (167858) 0:27:58.58
iso.3.6.1.2.1.1.4.0 = STRING: "root"
iso.3.6.1.2.1.1.5.0 = STRING: "Hacked"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown"
iso.3.6.1.2.1.1.7.0 = INTEGER: 14
iso.3.6.1.2.1.1.8.0 = Timeticks: (8) 0:00:00.08
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
```

Parte 4: Extraer información específica

Objetivo

Usar un **OID** para extraer información más detallada y específica del sistema.

```
snmpwalk -v1 -c iloveyou 192.168.100.145 1.3.6.1.2.1.25.4.2.1.2
```

Explicación del comando:

-

1.3.6.1.2.1.25.4.2.1.2: Este **OID** consulta una sección específica de la MIB relacionada con los procesos en ejecución.


```

(root@kali)-[/home/yorha2b]
# snmpwalk -v1 -c iloveyou 192.168.100.145 1.3.6.1.2.1.25.4.2.1.2
iso.3.6.1.2.1.25.4.2.1.2.1 = STRING: "systemd"
iso.3.6.1.2.1.25.4.2.1.2.2 = STRING: "kthreadd"
iso.3.6.1.2.1.25.4.2.1.2.3 = STRING: "pool_workqueue_release"
iso.3.6.1.2.1.25.4.2.1.2.4 = STRING: "kworker/R-rcu_g"
iso.3.6.1.2.1.25.4.2.1.2.5 = STRING: "kworker/R-rcu_p"
iso.3.6.1.2.1.25.4.2.1.2.6 = STRING: "kworker/R-slab_"
iso.3.6.1.2.1.25.4.2.1.2.7 = STRING: "kworker/R-netns"
iso.3.6.1.2.1.25.4.2.1.2.9 = STRING: "kworker/0:0H-events_highpri"
iso.3.6.1.2.1.25.4.2.1.2.12 = STRING: "kworker/R-mm_pe"
iso.3.6.1.2.1.25.4.2.1.2.13 = STRING: "rcu_tasks_trace_kthread"
iso.3.6.1.2.1.25.4.2.1.2.14 = STRING: "ksoftirqd/0"
iso.3.6.1.2.1.25.4.2.1.2.15 = STRING: "rcu_sched"
iso.3.6.1.2.1.25.4.2.1.2.16 = STRING: "migration/0"
iso.3.6.1.2.1.25.4.2.1.2.17 = STRING: "idle_inject/0"
iso.3.6.1.2.1.25.4.2.1.2.18 = STRING: "cpuhp/0"
iso.3.6.1.2.1.25.4.2.1.2.19 = STRING: "cpuhp/1"
iso.3.6.1.2.1.25.4.2.1.2.20 = STRING: "idle_inject/1"
iso.3.6.1.2.1.25.4.2.1.2.21 = STRING: "migration/1"
iso.3.6.1.2.1.25.4.2.1.2.22 = STRING: "ksoftirqd/1"
iso.3.6.1.2.1.25.4.2.1.2.24 = STRING: "kworker/1:0H-events_highpri"
iso.3.6.1.2.1.25.4.2.1.2.25 = STRING: "cpuhp/2"
iso.3.6.1.2.1.25.4.2.1.2.26 = STRING: "idle_inject/2"
iso.3.6.1.2.1.25.4.2.1.2.27 = STRING: "migration/2"
iso.3.6.1.2.1.25.4.2.1.2.28 = STRING: "ksoftirqd/2"
iso.3.6.1.2.1.25.4.2.1.2.29 = STRING: "kworker/2:0-cgroup_destroy"
iso.3.6.1.2.1.25.4.2.1.2.30 = STRING: "kworker/2:0H-events_highpri"
iso.3.6.1.2.1.25.4.2.1.2.31 = STRING: "cpuhp/3"
iso.3.6.1.2.1.25.4.2.1.2.32 = STRING: "idle_inject/3"
iso.3.6.1.2.1.25.4.2.1.2.33 = STRING: "migration/3"
iso.3.6.1.2.1.25.4.2.1.2.34 = STRING: "ksoftirqd/3"
iso.3.6.1.2.1.25.4.2.1.2.36 = STRING: "kworker/3:0H-events_highpri"
iso.3.6.1.2.1.25.4.2.1.2.37 = STRING: "kworker/u9:0-events_unbound"
iso.3.6.1.2.1.25.4.2.1.2.38 = STRING: "kworker/u10:0-events_unbound"
iso.3.6.1.2.1.25.4.2.1.2.39 = STRING: "kworker/u11:0-events_unbound"
iso.3.6.1.2.1.25.4.2.1.2.40 = STRING: "kworker/u12:0-events_unbound"
iso.3.6.1.2.1.25.4.2.1.2.41 = STRING: "kdevtmpfs"
iso.3.6.1.2.1.25.4.2.1.2.42 = STRING: "kworker/R-inet_"
iso.3.6.1.2.1.25.4.2.1.2.44 = STRING: "kauditd"
iso.3.6.1.2.1.25.4.2.1.2.45 = STRING: "khungtaskd"
iso.3.6.1.2.1.25.4.2.1.2.46 = STRING: "oom_reaper"
iso.3.6.1.2.1.25.4.2.1.2.47 = STRING: "kworker/R-write"
iso.3.6.1.2.1.25.4.2.1.2.48 = STRING: "kcompactd0"
iso.3.6.1.2.1.25.4.2.1.2.49 = STRING: "ksmd"
iso.3.6.1.2.1.25.4.2.1.2.50 = STRING: "khugepaged"
iso.3.6.1.2.1.25.4.2.1.2.51 = STRING: "kworker/R-kinte"
iso.3.6.1.2.1.25.4.2.1.2.52 = STRING: "kworker/R-kbloc"
iso.3.6.1.2.1.25.4.2.1.2.54 = STRING: "kworker/R-md"

```

Obtenemos una lista de procesos en ejecución en el sistema objetivo.

Las OIDs actúan como "direcciones" que indican qué datos queremos obtener.

Familiarizarse con ellas es clave para aprovechar SNMP al máximo.

¿Qué son las OIDs en SNMP?

Las **OIDs** (Object Identifiers o Identificadores de Objetos) son identificadores jerárquicos que se utilizan en **SNMP** (Protocolo Simple de Administración de Red) para identificar de forma única los objetos o variables que se pueden monitorear o gestionar en un dispositivo de red. En términos simples, una OID es como una "dirección" que apunta a un dato específico en un dispositivo administrado por SNMP, como un router, un switch, o una impresora.

Estructura de una OID

Una OID es una secuencia de números separados por puntos, organizada en un árbol jerárquico definido por la **ISO**. Cada nivel en la jerarquía representa una categoría o un nodo, y cada número identifica una rama o un objeto dentro de esa categoría.

Por ejemplo:

1.3.6.1.2.1.1.1

Desglose de este ejemplo:

1. : Representa el nivel raíz del árbol (ISO).
2. : Identifica el estándar desarrollado por la Organización Internacional de Normalización (ISO) y el Instituto de Ingeniería Eléctrica y Electrónica (ISO/ITU-T).
3. : Indica que pertenece al Departamento de Internet (IETF).
4. : Define los documentos estándar de Internet (Management).
5. 1: Específico del MIB (Management Information Base), donde se almacenan los datos gestionados.
6. 1.1: Apunta a un objeto específico, en este caso, el **nombre del sistema** (System Description).

Este árbol puede parecer abstracto, pero en la práctica se traduce en datos concretos como el estado del dispositivo, su CPU, uso de memoria, etc.

Ejemplos prácticos

1. Ejemplo simple: Consultar el nombre del dispositivo

OID: 1.3.6.1.2.1.1.5.0

Esta OID se utiliza para obtener el **nombre del host** de un dispositivo SNMP.

Resultado de una consulta SNMP:

HOST-NAME: "Router-Central"

1. Uso de CPU en un router

OID: 1.3.6.1.4.1.9.2.1.57.0

Este identificador podría corresponder al porcentaje de uso de la CPU en un dispositivo de Cisco.

Resultado:

CPU Utilization: 23%

1. Estado de la interfaz de red

OID: 1.3.6.1.2.1.2.2.1.8.x

Aquí, **x** es el índice de la interfaz.

- Si el valor es **1** La interfaz está activa (UP).
- Si el valor es **2** La interfaz está inactiva (DOWN).

Resultado para **1.3.6.1.2.1.2.2.1.8.3**:

Interface 3: UP

1. Consumo de memoria de un dispositivo

OID: **1.3.6.1.4.1.2021.4.6.0**

Esta OID podría corresponder al uso total de memoria en un dispositivo Linux.

Resultado:

Total Memory Used: 512 MB

En definitiva

- Las **OIDs** son como "direcciones" en un árbol de datos SNMP que te indican dónde buscar información específica en un dispositivo.
- Están organizadas jerárquicamente, con números separados por puntos.
- Cada OID corresponde a una métrica o valor medible (como nombre de host, estado de interfaz, uso de CPU, etc.).

Los comandos básicos como **snmpget** o **snmpwalk** te permiten explorar y consultar estas OIDs en dispositivos SNMP.

Enumeración SNMP con Herramientas Avanzadas

En esta práctica vamos a explorar herramientas adicionales que nos permiten realizar enumeración SNMP de una forma más eficiente. Además de **snmpwalk**, existen otras herramientas poderosas como **SNMP-check**, **Braa**, y **Metasploit**. Aprenderemos a usar estas herramientas para realizar auditorías de seguridad o pruebas de penetración en redes.

SNMP-Check

Objetivo

Aprender a usar la herramienta **SNMP-check** para enumerar dispositivos SNMP de manera legible y estructurada.

Comando básico:

```
snmp-check 192.168.100.145 -p 161 -c iloveyou
```

```
(root@kali)~/home/yorha2b
# snmp-check 192.168.100.145 -p 161 -c iloveyou
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.100.145:161 using SNMPv1 and community 'iloveyou'

[*] System information:

Host IP address      : 192.168.100.145
Hostname             : Hacked
Description           : VyOS 1.5-rolling-202412160007
Contact              : root
Location              : Unknown
Uptime snmp           : 00:35:11.49
Uptime system        : 00:34:12.98
System date          : 2025-1-11 17:10:26.0

[*] Network information:

IP forwarding enabled : yes
Default TTL           : 64
TCP segments received : 2
TCP segments sent     : 2
TCP segments retrans  : 0
Input datagrams       : 27721
Delivered datagrams   : 27705
Output datagrams      : 25981

[*] Network interfaces:

Interface            : [ up ] lo
Id                   : 1
Mac Address          : :::::
Type                 : softwareLoopback
Speed                : 10 Mbps
MTU                  : 65536
In octets             : 12532
Out octets           : 12532

Interface            : [ up ] eth0
Id                   : 2
Mac Address          : 08:00:27:b1:d3:8e
Type                 : ethernet-csmacd
Speed                : 1000 Mbps
```

Explicación de los parámetros:

- **-p 161** Este parámetro especifica el puerto en el que estamos buscando el servicio SNMP. En este caso, **161** es el puerto predeterminado para SNMP.
- **-c iloveyou** Aquí especificamos la **cadena de comunidad** que estamos utilizando para acceder al dispositivo. En este caso, es **iloveyou**, que corresponde a la cadena de comunidad con permisos de solo lectura.

¿Por qué usar SNMP-check?

SNMP-check es útil para realizar enumeración de SNMP de forma más amigable para los humanos. A diferencia de **snmpwalk**, que proporciona la salida en formato de árbol, **SNMP-check** organiza la información de manera más legible, lo que facilita la interpretación de los resultados.

Nota importante: Recuerda que las cadenas de comunidad como **iloveyou** no son seguras. En escenarios reales, es recomendable usar cadenas más fuertes.

Braa - Escáner Masivo de SNMP

Objetivo

Explorar la herramienta **Braa**, que permite realizar consultas SNMP de manera masiva, ideal para escanear múltiples dispositivos al mismo tiempo.

Comando básico:

```
braa iloveyou@192.168.100.145:.1.3.6.*
```

Explicación del comando:

- **iloveyou** Es la **cadena de comunidad** que se utilizará para acceder al dispositivo.
- **192.168.100.145** Es la IP del servidor SNMP al que estamos accediendo.
- **:.1.3.6.*** Es OID (identificador de objeto) que vamos a consultar. El **OID .1.3.6.*** cubre una amplia gama de objetos de administración SNMP, lo que permite consultar una gran cantidad de información.

¿Qué hace Braa?

Braa es ideal para realizar escaneos SNMP a gran escala, ya que es capaz de consultar múltiples dispositivos SNMP de forma simultánea y rápida. Esto es útil cuando tenemos que escanear una red entera o un gran número de dispositivos sin consumir muchos recursos.

Diferencia clave con snmpwalk: A diferencia de **snmpwalk**, que requiere que consultes dispositivos uno por uno, **Braa** te permite hacer consultas simultáneas a muchos dispositivos a la vez, lo que mejora la eficiencia en entornos grandes.

Parte 3: Enumeración SNMP con Metasploit

Objetivo

Aprender a usar el módulo de de dispositivos SNMP. **Metasploit** llamado **snmp_enum** para enumerar información

Comandos básicos en Metasploit:

1. Abrir Metasploit:

```
msfconsole
```

1. Usar el módulo snmp_enum:

```
use auxiliary/scanner/snmp/snmp_enum
```

1. Ver opciones del módulo:

```
show options
```

1. Configurar la IP del objetivo:

```
set RHOSTS 192.168.100.145
```

1. Configurar la cadena de comunidad:

```
set community iloveyou
```

1. Ejecutar el escaneo:

```
run
```

Explicación:

- **use auxiliary/scanner/snmp/snmp_enum**: Usamos el módulo **snmp_enum** de Metasploit, diseñado específicamente para enumerar información SNMP.
- **set RHOSTS 192.168.100.145**: Establecemos la IP del dispositivo de destino.
- **set community iloveyou**: Configuramos la cadena de comunidad que estamos usando.
- **run**: Ejecutamos el escaneo.


```

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/snmp/snmp_enum
msf6 auxiliary(scanner/snmp/snmp_enum) > set RHOSTS 192.168.100.145
RHOSTS => 192.168.100.145
msf6 auxiliary(scanner/snmp/snmp_enum) > set community iloveyou
community => iloveyou
msf6 auxiliary(scanner/snmp/snmp_enum) > run

[+] 192.168.100.145, Connected.

[*] System information:

Host IP           : 192.168.100.145
Hostname          : Hacked
Description       : VyOS 1.5-rolling-202412160007
Contact           : root
Location          : Unknown
Uptime snmp       : 00:49:39.86
Uptime system     : 00:48:41.35
System date       : 2025-1-11 17:24:54.0

[*] Network information:

IP forwarding enabled : yes
Default TTL           : 64
TCP segments received : 2
TCP segments sent     : 2
TCP segments retrans  : 0
Input datagrams       : 30138
Delivered datagrams   : 29269
Output datagrams      : 27203

[*] Network interfaces:

Interface          : [ up ] lo
Id                 : 1
Mac Address        : :::::
Type               : softwareLoopback
Speed              : 10 Mbps
MTU                : 65536
In octets          : 12532
Out octets         : 12532

Interface          : [ up ] eth0
Id                 : 2
Mac Address        : 08:00:27:b1:d3:8e
Type               : ethernet-csmacd
Speed              : 1000 Mbps
MTU                : 1500
In octets          : 3078471

```

El módulo de **Metasploit** devuelve los mismos resultados que obtenemos con herramientas como **snmpwalk** o **snmp-check**, pero con la ventaja de estar integrados en una plataforma de pruebas de penetración como **Metasploit**, lo que permite una integración más fluida con otras herramientas.

Reflexión Final: Explorando SNMP y Técnicas de Enumeración en Seguridad Informática

Hemos recorrido un camino completo desde los fundamentos de **SNMP (Simple Network Management Protocol)** hasta la práctica de enumeración utilizando herramientas avanzadas de seguridad. Este protocolo, diseñado para la gestión y monitoreo de dispositivos en redes, se ha convertido en un área clave en pruebas de penetración y auditorías de seguridad debido a las vulnerabilidades asociadas a versiones anteriores como **SNMPv1** y **SNMPv2c**.

Lo visto:

1. Configuración y gestión de SNMP:

- Configuramos SNMP en un dispositivo **Vyos**.
- Aprendimos la diferencia entre cadenas de comunidad de solo lectura (**ro**) y lectura-escritura (**rw**).
- Reflexionamos sobre la importancia de elegir cadenas seguras y evitar las predeterminadas como **public** o **private**.

1. Pruebas iniciales con Kali Linux:

- Usamos herramientas básicas como **nmap** y **snmpwalk** para identificar servicios activos y extraer información de dispositivos.
- Exploramos cómo las cadenas de comunidad débiles pueden ser utilizadas para acceder a datos críticos.

1. Enumeración avanzada:

- Con **snmp-check**, obtuvimos información en un formato más organizado y legible.
- Usamos **Braa** para realizar consultas masivas, optimizando el tiempo en redes grandes.
- Con **Metasploit**, integramos la enumeración SNMP en un marco avanzado para pruebas de penetración.

1. Manipulación de datos con SNMP:

- Aprendimos cómo cambiar configuraciones de un dispositivo utilizando herramientas como **snmpset**, demostrando la importancia de proteger dispositivos SNMP de accesos no autorizados.

A tener en cuenta:

- **Seguridad básica es clave:** Aunque SNMP es una herramienta de gestión útil, las versiones más antiguas son inherentemente inseguras, ya que las cadenas de comunidad se transmiten en texto plano. Esto nos lleva a subrayar la importancia de usar **SNMPv3** en implementaciones modernas.
- **Comprender las herramientas:** Cada herramienta que utilizamos (como **nmap**, **snmpwalk**, **snmp-check**, o **Metasploit**) tiene sus fortalezas. Conocerlas nos permite elegir la más adecuada según el contexto.
- **El atacante y el administrador:** Al aprender cómo un atacante podría explotar las vulnerabilidades de SNMP, también adquirimos habilidades para proteger mejor nuestras redes como administradores.