Prosody

Actualizar lista de paquetes

```
root@prosody:/home/usoftware# apt update && apt upgrade -y
Obj:1 http://deb.debian.org/debian bookworm InRelease
Des:2 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:3 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [195 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [290 kB]
Descargados 588 kB en 1s (548 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@prosody:/home/usoftware#
```

Instalar prosody y librerías recomendadas (lua-event)

apt install -y prosody lua-event lua-sec

```
Seleccionando el paquete libevent-2.1-7:amd64 previamente no seleccionado.
Preparando para desempaquetar .../06-libevent-2.1-7_2.1.12-stable-8_amd64.deb ...
Desempaquetando libevent-2.1-7:amd64 (2.1.12-stable-8) ...
Seleccionando el paquete libunbound8:amd64 previamente no seleccionado.
Preparando para desempaquetar .../07-libunbound8_1.17.1-2+deb12u3_amd64.deb ...
Desempaquetando libunbound8:amd64 (1.17.1-2+deb12u3) ...
Seleccionando el paquete lua-bit32:amd64 previamente no seleccionado.
Preparando para desempaquetar .../08-lua-bit32_5.3.0-4_amd64.deb ...
Desempaquetando lua-bit32:amd64 (5.3.0-4) ...
Seleccionando el paquete lua-event:amd64 previamente no seleccionado.
Preparando para desempaquetar .../09-lua-event_0.4.6-2+b1_amd64.deb ...
Desempaquetando lua-event:amd64 (0.4.6-2+b1) ...
Seleccionando el paquete lua-posix:amd64 previamente no seleccionado.
Preparando para desempaquetar .../10-lua-posix_33.4.0-3+b1_amd64.deb ...
Desempaquetando lua-posix:amd64 (33.4.0-3+b1) ...
Seleccionando el paquete lua-readline:amd64 previamente no seleccionado.
Preparando para desempaquetar .../11-lua-readline_3.2-2_amd64.deb ...
Desempaquetando lua-readline:amd64 (3.2-2) ...
Seleccionando el paquete lua-unbound:amd64 previamente no seleccionado.
Preparando para desempaquetar .../12-lua-unbound_1.0.0-2_amd64.deb ...
Desempaquetando lua-unbound:amd64 (1.0.0-2) ...
Configurando lua5.4 (5.4.4-3+deb12u1) ...
update-alternatives: utilizando /usr/bin/lua5.4 para proveer /usr/bin/lua (lua-interpreter) en modo automático
update-alternatives: utilizando /usr/bin/luac5.4 para proveer /usr/bin/luac (lua-compiler) en modo automático
Configurando lua-expat:amd64 (1.5.1-3) ...
Configurando lua-filesystem:amd64 (1.8.0-3) ...
Configurando lua-bit32:amd64 (5.3.0-4) ...
Configurando ssl-cert (1.1.2) ...
Configurando libevent-2.1-7:amd64 (2.1.12-stable-8) ...
Configurando lua-bitop:amd64 (1.0.2-7) ...
Configurando lua-posix:amd64 (33.4.0-3+b1) ...
Configurando lua-readline:amd64 (3.2-2) ...
Configurando prosody (0.12.3-1) ...
grep: /etc/prosody/prosody.cfg.lua: No existe el fichero o el directorio
Configurando lua-event:amd64 (0.4.6-2+b1) ...
Configurando libunbound8:amd64 (1.17.1-2+deb12u3) ...
Configurando lua-unbound:amd64 (1.0.0-2) ...
Procesando disparadores para man-db (2.11.2-2) ...
Procesando disparadores para libc-bin (2.36-9+deb12u13) ...
root@prosody:/home/usoftware#
```

Configuración del dominio virtual host

nano /etc/prosody/prosody.cfg.lua

```
---------------- ZONA SENTINEL NEXUS --------------------
VirtualHost "sentinelnexus.local"
        authentication = "internal_plain"
        c2s_require_encryption = false
        s2s_require_encryption = false
        allow_unencrypted_plain_auth = true_
```

Crear los agentes

Agente monitor

 # Crear al Agente Monitor

prosodyctl register monitor sentinelnexus.local sentinel123

 # Crear al Agente Cerebro

prosodyctl register cerebro sentinelnexus.local sentinel123

Para ver los agentes creados

prosodyctl register monitor sentinelnexus.local sentinel123

```
root@prosody:/home/usoftware# ls /var/lib/prosody/sentinelnexus%2elocal/accounts/
cerebro.dat  monitor.dat
root@prosody:/home/usoftware#
```

Para instalar spade necesitamos una versión más vieja de Python, la 3.11 ya que en la 3.13 no tiene compatibilidad con la librería, es por eso por lo que todos vamos a utilizar el agente spade y en entorno virtual con Python 3.11

Vamos a estandarizar el proyecto con Python 3.11.9 para no tener problemas de compilación

Descargar Python 3.11.9 y agregar el path
luego cambiar el nombre de la carpeta del venv a venv_viejo (ir a la ruta donde tienes el proyecto)

Crear el nuevo entorno con Python 3.11

py -3.11 -m venv venv

(Si no te funciona py, prueba con la ruta completa: C:\Python311\python.exe -m venv venv).

Activar e instalar

.\venv\Scripts\activate

pip install -r requirements.txt

Instalar spade

pip install spade

Despues de instalar spade ahora queda decirle a Windows donde queda el servidor DEBIAN

Para eso **con ip a** buscamos la ip del servidor en nuestra maquina virtual

```
root@prosody:/home/usoftware# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:5d:c1:3d brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.100.100.41/24 brd 10.100.100.255 scope global ens18
       valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe5d:c13d/64 scope link
       valid_lft forever preferred_lft forever
```

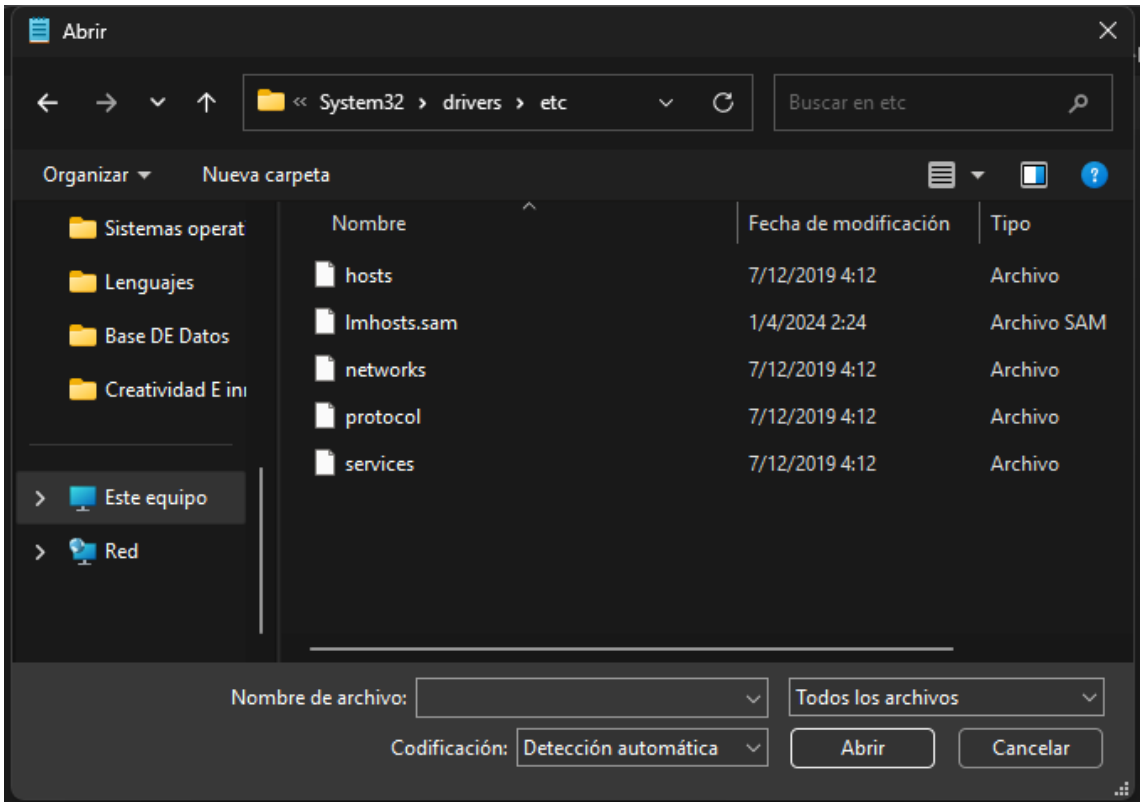
Configurar el archivo host en Windows

Ip del servidor: 10.100.100.41

En Windows, abrir el block de notas como administrador y buscar la siguiente ruta

C:\Windows\System32\drivers\etc y cambiar a todos los archivos para ver los documentos host

Abrir el archivo host



Ir al final del archivo y agregar la siguiente línea

```
10.100.100.41    sentinelnexus.local
```

Guardar el archivo y cerrar

Luego hacer una prueba de conexión ping

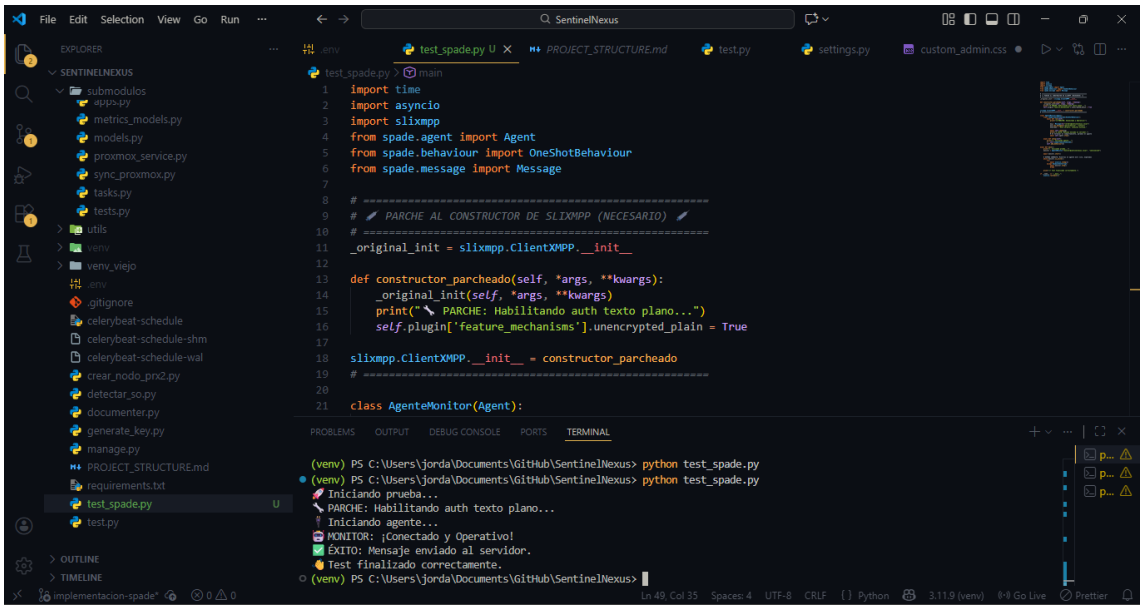
Desde la terminal de vscode o de Windows

```
C:\Windows\System32>ping sentinelnexus.local

Haciendo ping a sentinelnexus.local [10.100.100.41] con 32 bytes de datos:
Respuesta desde 10.100.100.41: bytes=32 tiempo=5ms TTL=61
Respuesta desde 10.100.100.41: bytes=32 tiempo=4ms TTL=61
Respuesta desde 10.100.100.41: bytes=32 tiempo=4ms TTL=61
Respuesta desde 10.100.100.41: bytes=32 tiempo=6ms TTL=61

Estadísticas de ping para 10.100.100.41:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 4ms, Máximo = 6ms, Media = 4ms
```

Paso final hacer un script de prueba

Verificar que la librería SPADE en Windows pueda iniciar sesión en el servidor DEBIAN

Scritpt de prueba + conexión exitosa



Para establecer una conexión en tiempo real entre el sistema sentinel nexus y los servidores se implemento una arquitectura basada en el estándar XMPP, utilizando el server prosody

Las acciones realizadas fueron:

1. **Configuración del Servidor (Prosody):** Se deshabilitó explícitamente el módulo de encriptación TLS (modules_disabled = { "tls" }) para el dominio local sentinelnexus.local, permitiendo conexiones directas al puerto 5222 sin la sobrecarga de negociación de certificados.
2. **Adaptación del Cliente (SPADE/Python):** Se implementó un "Monkey Patch" (parche en tiempo de ejecución) en la librería slixmpp. Este parche intercepta el

constructor del cliente XMPP para habilitar la bandera unencrypted_plain = True. Esto autoriza al agente a autenticarse mediante el mecanismo SASL PLAIN sin requerir un túnel encriptado previo.

En cuanto a la seguridad

1. Aislamiento de Red (Seguridad Perimetral) Esta configuración **NO** es insegura porque el tráfico de datos viaja exclusivamente a través de una **Red de Área Local (LAN)** privada y aislada (10.100.100.41).

- A diferencia de internet, donde los datos pasan por routers públicos, aquí los paquetes viajan directamente del servidor al cliente a través de un switch virtual controlado.
- Para que alguien intercepte la contraseña, tendría que haber hackeado ya la red física de la universidad o tener acceso administrativo al hypervisor Proxmox. Si el atacante ya está ahí, tener SSL no serviría de nada.

2. Latencia y Rendimiento (Eficiencia) En sistemas de monitoreo en tiempo real (como SentinelNexus), la velocidad es crítica.

- La encriptación SSL/TLS añade una capa de procesamiento extra (handshake) cada vez que un agente se conecta.
- Al eliminar la encriptación innecesaria en una red segura, reducimos la latencia y el consumo de CPU tanto en el servidor como en los agentes, haciendo que el monitoreo sea más ligero y rápido.

3. **Entorno de Confianza Controlado** El uso de certificados SSL es vital para verificar la identidad del servidor ("¿Es este servidor quien dice ser?"). En nuestro caso, nosotros somos dueños tanto del servidor (Debian) como del cliente (Windows/Django). No necesitamos un certificado para "confiar" en nosotros mismos; la confianza está implícita en la infraestructura que controlamos.