Autor: Roosvelt David Colaço

Blumenau 10/05/2023

Caderno de testes para validação de vlans em switches gerenciáveis

Sumário

Objetivo do caderno de testes

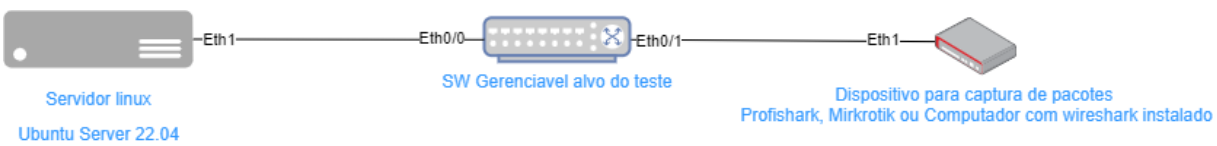
Este teste automatizado tem como objetivo verificar se um determinado switch está recebendo e encaminhando tags de vlans corretamente

Requisitos de Hardware e Software

- Sistema Operacional Ubuntu 22.04 ou LTS mais recente
- Usuário com permissão root no sudoers
- Interpretador Python3 instalado
- Gerenciador de pacotes pip
- Bibliotecas Scapy e tqdm instaladas
- Um Switch de rede compatível com vlans
- Dois computadores, ou um computador com 2 interfaces de rede, ou um computador e um dispositivo com capacidade de captura de pacotes (Profishark, Mikrotik packet sniffer)
- Software de captura de pacotes como o Wireshark

Colocando a mão na massa e preparando o teste.

- Instale o Sistema operacional Ubuntu em uma máquina física ou virtual, garante que a máquina tenha acesso direto a interface de rede que estará conectada com o Switch.
- Abra o Terminal e atualize o sistema com *sudo apt update && apt upgrade -y*
- Instale o Python3 e o gerenciador de pacotes pip com *sudo apt install python3 pip*
- Instale as bibliotecas com *sudo pip3 install scapy* e *sudo pip3 install tqdm*
- Vá para o diretório do linux /opt com *cd /opt*
- Baixe o programa de teste com o seguinte comando:
  - *wget https://raw.githubusercontent.com/roosveltdavid/Testeautomatizadovlan/main/testeswitch.py*
  - digite *sudo chmod +x testeswitch.py* para dar permissão de execução ao arquivo
- Conecte os dispositivos seguindo a lógica abaixo



- Configure as interfaces de rede (switch, computadores e máquina de captura de pacotes) às interfaces de rede da máquina.
- Verifique a interface criada pelo sistema operacional no servidor gerador de tráfego
    - Utilize o comando *ip -br a s*
    - Caso a interface pretendida esteja down mesmo com o cabo conectado, digite *ip link set eth1 up* (o nome eth1 pode mudar conforme o hardware)
    - Colete o mac da interface com o comando *ip link show*
    - Salve a informação do nome da interface pois ela será necessária na execução do código
- Certifique-se de que a máquina de captura de pacotes está configurada para capturar o tráfego na interface correta.
    - No caso do mikrotik iremos criar as vlans para serem testadas de forma massiva. Para isso usaremos o comando abaixo

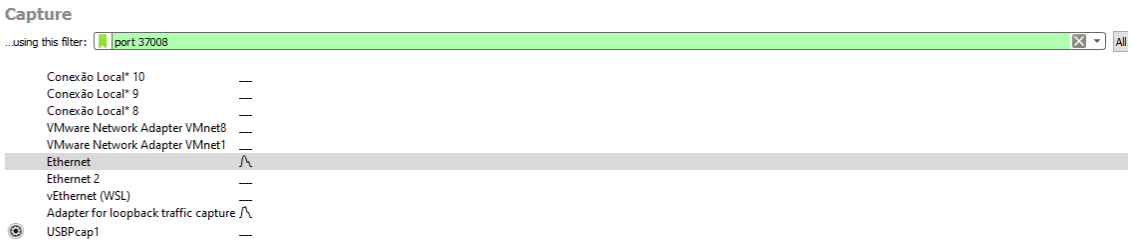
        :for x from=10 to=256 do={interface vlan add name=("vlan$x") vlan-id=$x interface=ether1}

    - Para configurar a captura usaremos a funcionalidade “packet sniffer incluída por padrão no RouterOS

        /tool sniffer

        set file-name=capturavlans.pcapng filter-interface=ether1 streaming-enabled=yes streaming-server=192.168.0.2

A configuração acima salvará um arquivo .pcapng no mikrotik e também pode fazer o streaming do fluxo na porta 37008 que poderá ser lido por outro computador escutando o tráfego nesta porta



Executando o teste

O teste foi pensado para ter uma interface fácil e rápida para inserção de dados conforme a característica de cada ambiente de teste

Em nosso cenário de testes usaremos os seguintes dados:

- Vlans de teste: 10-256
- Interface de rede utilizada no servidor: ens4
- Quantidade de pacotes a serem enviados: 100 (por vlan)
- Mac de Origem: 00:11:22:33:44:55
- Mac de Destino: 50:00:00:02:00:00

- IP de Origem: 192.168.100.1
- Ip de Destino: 192.168.100.2
- Porta: 9444

IPs e portas são apenas para gerar um payload útil dentro do pacote para melhorar a eficiência do teste.

Antes de rodar o teste seguir o seguinte checklist abaixo:

- Checar se todos os equipamentos estão ligados
- Checar se todas as portas envolvidas estão linkadas
- Ativar a captura de pacotes, ativar o recebimento do stream no wireshark
- Validar se a coleta está funcionando

Com o Checklist executado, iniciaremos o teste. O programa foi desenvolvido pensando em ter uma interface de input dos dados para facilitar o uso em múltiplos ambientes de testes.

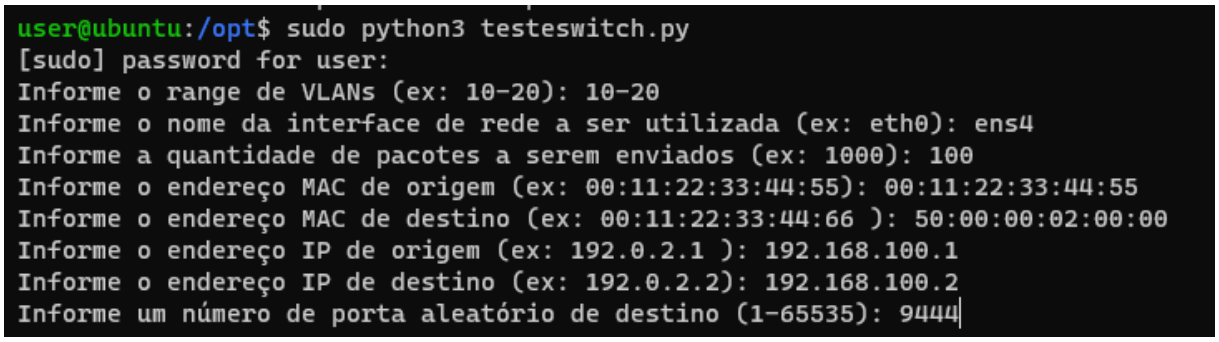
Para chamar o teste basta utilizar os seguintes comandos:

cd /opt

sudo python3 testeswitch.py

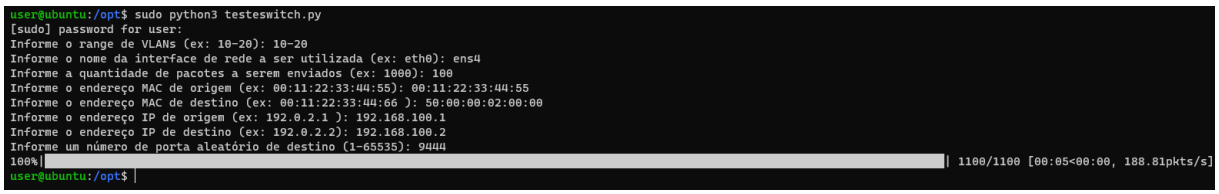
imediatamente o sistema pedirá a senha de elevação de privilégios (é necessário para que o script consiga acessar a placa de rede em modo privilegiado e promíscuo)

em seguida irá dar prompt a prompt as informações a serem preenchidas conforme mencionamos acima.

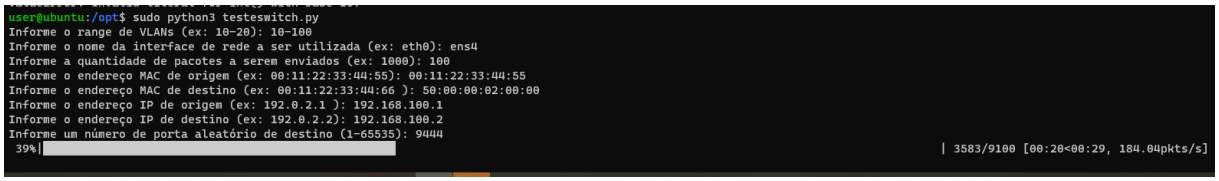


Após digitar a porta de destino e teclar enter o teste irá iniciar.

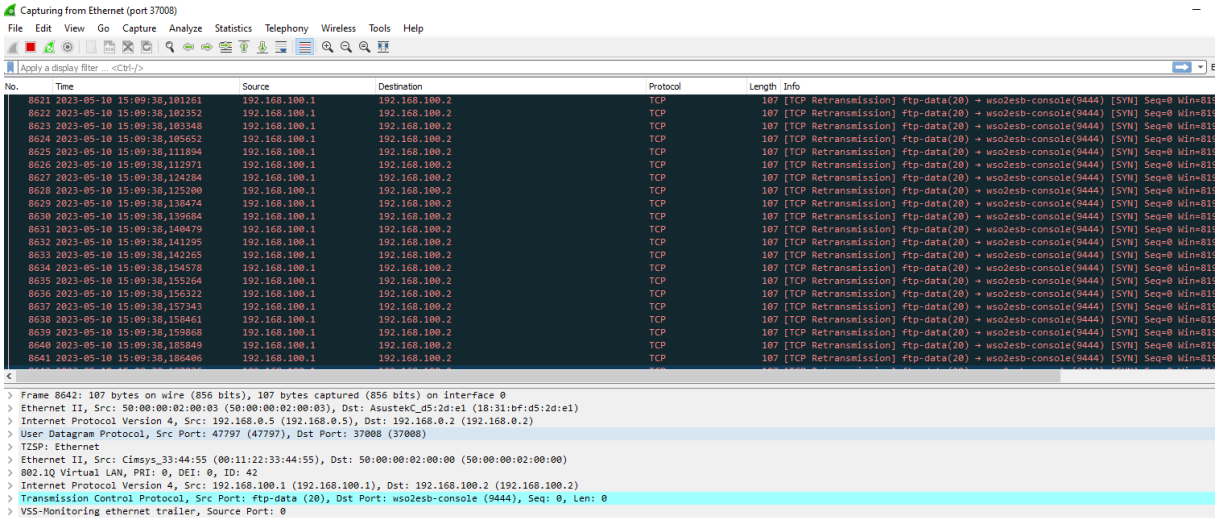
Uma barra de progresso irá mostrar o andamento do teste.



Enquanto o teste executa



Podemos acompanhar a captura de pacotes no wireshark



Após a conclusão do teste podemos analisar dentro do wireshark os pacotes recebidos

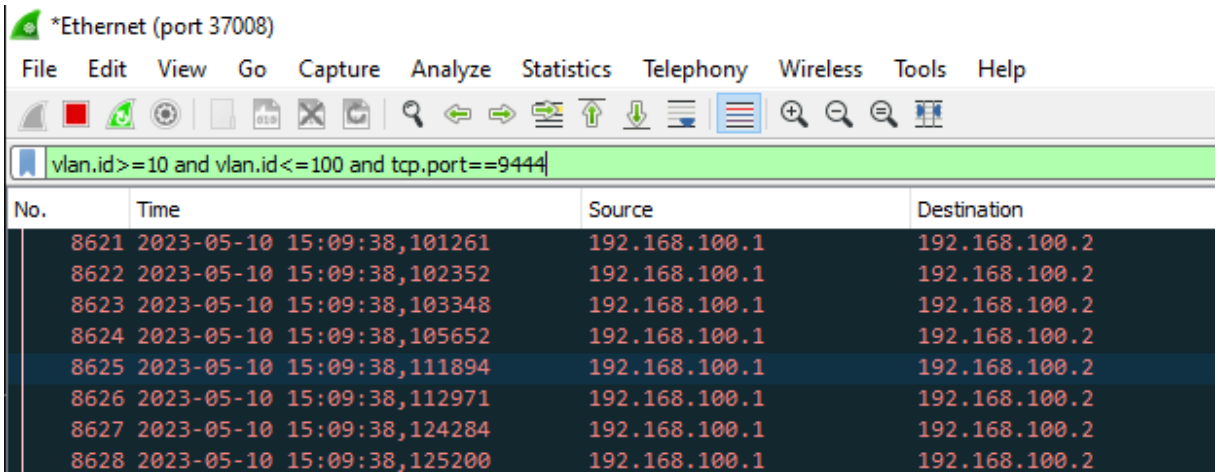
Resultados dos testes

Para analisar os resultados podemos utilizar o filtro *vlan.id* de acordo com as vlans utilizadas.

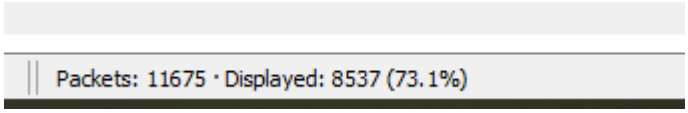
No teste acima fizemos o teste com os ranges de vlan de 10 a 100, para isso usaremos o filtro abaixo:

vlan.id>=10 and vlan.id<=100

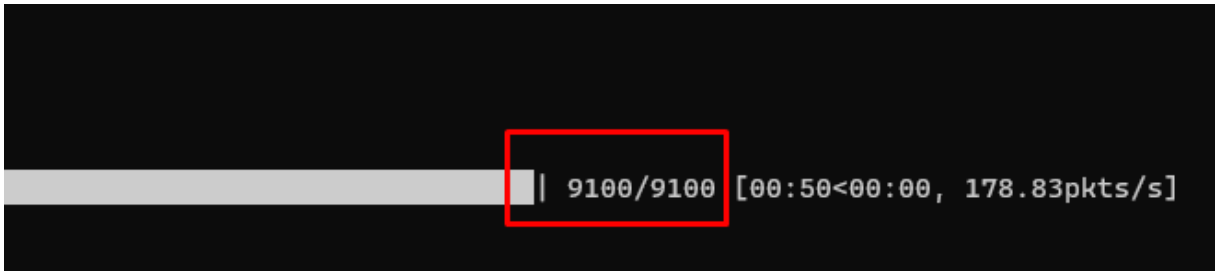
Para reduzir o lixo de informações como arp-requests e outros pacotes, vamos adicionar a flag *tcp.port==9444* para filtrar especificamente o tráfego gerado pela ferramenta.



Para averiguar o sucesso do teste precisamos ver a quantidade de pacotes mostrados com o filtro



Este número deve estar próximo do mostrado no console da ferramenta de testes.



Se houver uma discrepância muito grande, é recomendado revisar as configurações do equipamento e testar ranges de vlans menores até confirmar que os pacotes enviados estão sendo corretamente recebidos em todas as vlans.

Desta forma temos uma bastante rápida para averiguações em massa do funcionamento de um switch com suporte a vlans.