

Instalação e Configuração do Ambiente Virtual

Em ambas as máquinas:

1. **Instalar o Python 3.13:** `sudo apt install python3.13`
 2. **Instalar o pacote para criar ambientes virtuais:** `sudo apt install python3.13-venv`
 3. **Criar um ambiente virtual na pasta corrente:** `python3 -m venv .venv`
 4. **Ativar o ambiente virtual:** `source .venv/bin/activate`
 5. **Instalar o pip (se necessário):** `sudo apt install python3-pip -y`
-

Instalação de Dependências nas Máquinas

Na máquina do atacante (Linux Mint):

Instale os pacotes necessários com o pip:

`pip install numpy`

`pip install pandas`

`pip install ping3`

`pip install scapy`

Na máquina do cliente (Ubuntu):

Instale os pacotes necessários com o pip:

`pip install numpy`

`pip install pandas`

`pip install pyshark`

`pip install flask`

`pip install joblib`

Execução dos Scripts

Na máquina do atacante:

1. **Obtenha privilégios de superusuário:** `sudo su`
2. **Ative o ambiente virtual:** `source .venv/bin/activate`
3. **Para enviar pacotes benignos:** `python3 benign_package.py`

4. **Para enviar pacotes malignos:** `python3 malign_package.py`
5. **Para enviar pacotes aleatórios:** `python3 random_package.py`

Na máquina do cliente:

1. **Ative o ambiente virtual:** `source .venv/bin/activate`
 2. **Para receber e processar pacotes benignos ou malignos:** `python3 package_receiver_controlled.py`
 3. **Para receber e processar pacotes aleatórios:** `python3 package_receiver_random.py`
 4. **API de predição para pacotes benignos e malignos:** `python3 rest_controlled.py`
 5. **API de predição para pacotes aleatórios:** `python3 rest_random.py`
-

Funcionamento dos Algoritmos

Na máquina do atacante:

- **benign_package.py:** Este script seleciona dados do conjunto de tráfego benigno, cria um pacote e envia para a máquina de destino (cliente).
- **malign_package.py:** Este script seleciona dados do conjunto de tráfego maligno, cria um pacote e envia para a máquina de destino (cliente).
- **random_package.py:** Este script cria pacotes aleatórios e os envia para a máquina de destino (cliente).

Na máquina do cliente:

- **package_receiver_controlled.py:** Este script recebe pacotes benignos ou malignos na interface de rede definida, envia os dados para a API `rest_controlled`, que utiliza o modelo de IA (SVM) para prever a natureza do tráfego. Se a previsão indicar que é um ataque, o endereço IP do remetente é bloqueado através do iptables.
- **package_receiver_random.py:** Este script recebe pacotes aleatórios na interface de rede definida, envia os dados para a API `rest_random`, que utiliza o modelo de IA (Random Forest) para prever a natureza do tráfego. Se a previsão indicar que é um ataque, o endereço IP do remetente é bloqueado através do iptables.
- **rest_controlled.py:** Este script é responsável por receber os dados extraídos de pacotes benignos ou malignos e utiliza o algoritmo **SVM** (Support Vector Machine) para fazer a previsão do tráfego, retornando se é um tráfego benigno ou um ataque.

- **rest_random.py**: Este script é responsável por receber os dados extraídos de pacotes aleatórios e utiliza o algoritmo **Random Forest** para fazer a previsão do tráfego, retornando se é um tráfego benigno ou um ataque.