



Relatório de Aula Prática – Redes de Computadores

Título: Criptografia

Aluno: Vitor Mayorca Camargo

Data: 27/02/2025

1. INTRODUÇÃO

A internet surgiu na década de 1960 como ARPANET, inicialmente usada para comunicação militar e acadêmica nos EUA (LEINER, *et al.* 2009). Em 1989, a criação da World Wide Web democratizou o acesso e compartilhamento de informações por meio de navegadores web, permitindo que pessoas comuns usassem a internet de forma fácil e rápida (MONTEIRO, 2001).

As informações transmitidas na internet seguem protocolos rigorosos, como o TCP e o IP, que surgiram da necessidade de estabelecer regras e padrões para unificar as diferentes redes e máquinas que formam a internet (CORONA, 2004). O protocolo TCP (Transmission Control Protocol) foi projetado para fornecer uma comunicação confiável entre dois computadores, reduzindo ou aumentando a taxa de transmissão de dados de forma dinâmica, com o fim de evitar perda de dados. (CAMPISTA, *et al.* 2010).

Já o protocolo IPv4 (Internet Protocol) define a base para a transação, tráfego e reconhecimento de dados em uma rede. Ele é responsável por definir o "Endereço IP", que é um número único dado a cada máquina ou "host" na rede. Esse endereço é composto por quatro números (de 0 a 255) separados por pontos (de 000.000.000.000 a 255.255.255.255), e permite que uma máquina seja identificada e se comunique com outras na rede (CORONA, *et al.* 2004). Os autores também explicam que o protocolo TCP/IP divide os dados em pequenas porções para transferência na Internet. O protocolo IP organiza o envio e recebimento desses pacotes, enquanto o protocolo TCP se encarrega de dividir, ordenar e garantir que o fluxo de dados ocorra corretamente.

Duas versões do protocolo IP são usadas como padrão na internet. O IPv4 especifica as capacidades e protocolos básicos que toda máquina deve seguir, e usa um endereço de 32 bits (TANENBAUM, 2003). Já o IPv6, mais avançado, possui uma maior segurança e melhor comunicação entre as redes, fornecendo um endereço de 128 bits (CHANDRA, KATHING, KUMAR. 2013).

Com a popularização da internet, se tornou cada vez mais necessário garantir a segurança dos dados transmitidos na internet, contra ataques, manipulações, e acessos não autorizados. Uma das técnicas mais comuns de proteção de dados digitais é com o uso de *firewalls*. Kurose e Ross (2021) explicam que um *firewall* é um sistema de hardware e/ou software responsável por proteger a rede interna de uma organização, controlando todo o tráfego entre a internet e os recursos da rede. Os mesmos autores também explicam que um *firewall* tem três objetivos: garantir que todo o tráfego passe por ele, permitir apenas o tráfego autorizado conforme a política de segurança, e ser resistente a invasões para evitar vulnerabilidades.

Os Firewalls também podem ser divididos em três grupos: filtros de pacotes tradicionais, filtros de estado e gateways de aplicação. Os filtros de pacotes tradicionais são aplicados no roteador de borda da rede, e analisam os datagramas individualmente, e decidem se o bloqueiam ou o liberam, por meio de diferentes políticas de filtragem. Filtros de pacotes com controle de estado rastreiam conexões TCP antes de tomar decisões de filtragem. Eles analisam pacotes

individualmente, e monitoram a criação e encerramento dessas conexões TCP, sempre verificando se um pacote faz parte de uma conexão legítima (KUROSE; ROSS, 2021).

No Ubuntu, o UFW (Uncomplicated Firewall) é a ferramenta padrão de configuração de *firewall*. Ele permite criar um firewall baseado em host para IPv4 e IPv6 de forma amigável. Por padrão, o UFW está desativado. Para usuários que preferem uma interface gráfica, existe o Gufw. Ao ativar o UFW, ele aplica regras padrão que bloqueiam conexões de entrada, exceto algumas exceções para facilitar o uso doméstico (UBUNTU, 2025).

A atividade prática realizada no dia 27/02/2025 teve como objetivo testar as funcionalidades do UFW seguindo as instruções do manual disponibilizado no website oficial da Ubuntu. Depois, a eficácia do *firewall* foi verificada com o uso do sniffer de rede *Wireshark*.

Os testes foram realizados numa máquina virtual, executando num notebook da marca DELL, modelo Vostro 3520, com um CPU Intel Core i7-1255U 1.70 GHz, 16Gb de memória RAM DDR4, placa de vídeo integrada Intel Iris Xe Graphics, placa de vídeo dedicada NVIDIA GeForce MX550, adaptador de rede modelo Intel(R) Wi-Fi 6 AX201 160MHz, unidade de disco SSD NVMe ADATA 512Gb (Figura 2).

```

bodhi@bodhi:~$ neofetch

      .
      *  -/+00SSSS00+/- .
      **                               S+:
      -+S  ****                      S+-
      .OS  *****                      SO.
      /S   *****                      S\
      +S   *****                      S+
      /S   *****                      S\
      .S   *****                      S.
      +S   *****                      S+
      O   *****                      O
      O   *****                      O
      +S   *****                      S+
      .S   *****                      S.
      \S   *****                      S/
      +S   *****                      S+
      +S   *****                      S+


      \S   *****      ****      S/

      .O                               ***
      -+S                               ***
      :+S                               S+:  **
      . -/+00SSSS00+/- .              **

bodhi@bodhi
-----
OS: Bodhi 7.0 x86_64
Host: VirtualBox 1.2
Kernel: 5.15.0-78-generic
Uptime: 56 mins
Packages: 1208 (dpkg)
Shell: bash 5.1.16
Resolution: 800x600
DE: Enlightenment
WM: Moksha
WM Theme: MokshaGreen
Theme: MokshaGreen [GTK2/3]
Icons: Icons-Moksha-Green [GTK2/3]
Terminal: terminology
Terminal Font: terminus-18-bold
CPU: 12th Gen Intel i7-1255U (1) @

GPU: 00:02.0 VMware SVGA II Adapter

Memory: 1038MiB / 1964MiB



```

Figura 1: Ambiente virtual após a execução do comando *neofetch*.

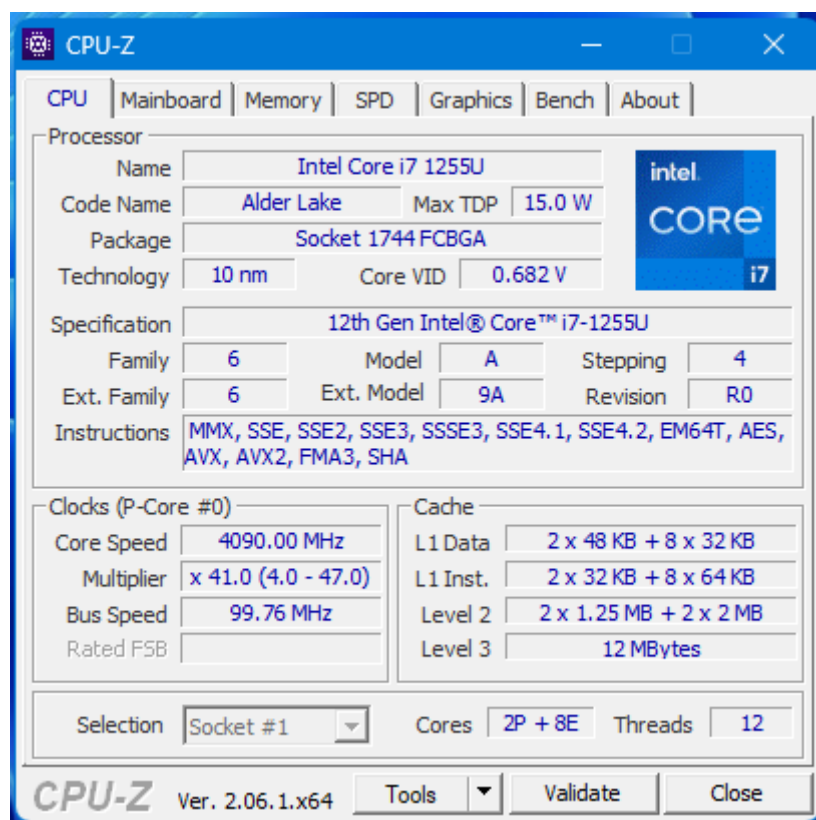


Figura 2: Especificações do sistema segundo o software *CPU-Z*.

4. METODOLOGIA

4.1 Criando o Firewall com UFW:

O firewall foi ativado e configurado seguindo as orientações presentes na documentação oficial da Ubuntu (UBUNTU, 2025).

4.2 Testando o Firewall com Wireshark:

Para testar o firewall, foi criada uma regra simples: um bloqueio na porta padrão do HTTP e HTTPS (portas 80 e 443). Para isso, usaram-se os comandos mostrados na figura 3. Depois, foi iniciada uma captura no *wireshark*, e foram feitos alguns acessos a diferentes websites pelo navegador e pelo comando *curl*.

```
bodhi@bodhi:~$ sudo ufw deny out 443/tcp
Rule added
Rule added (v6)
bodhi@bodhi:~$ sudo ufw deny out 80/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
bodhi@bodhi:~$ sudo ufw deny 80/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
bodhi@bodhi:~$ sudo ufw deny 443/tcp
Rule added
Rule added (v6)
```

Figura 3: Regras usadas no *firewall*.

5. RESULTADOS E DISCUSSÃO

5.1 Criando o Firewall com UFM:

O *firewall* foi criado com sucesso, seguindo o tutorial da Ubuntu. A figura 4 mostra as regras criadas após o final do tutorial.

5.1 Testando o Firewall com Wireshark:

Após criar um novo *firewall* que bloqueasse conexões HTTP, foram usados os comandos “curl google.com” e “curl example.com”. Também tentou-se acessar os mesmos websites pelo navegador. Percebe-se que os acessos foram bloqueados com sucesso (figuras 4 e 5). Ao avaliar a captura feita pelo wireshark (figura 6), foram adicionados os filtros “tcp.port == 80” e “tcp.port == 443”. Como esperado, o *firewall* bloqueou todo o tráfego nessas portas, então o wireshark não conseguiu capturar nenhum pacote sendo comutado nessas portas.

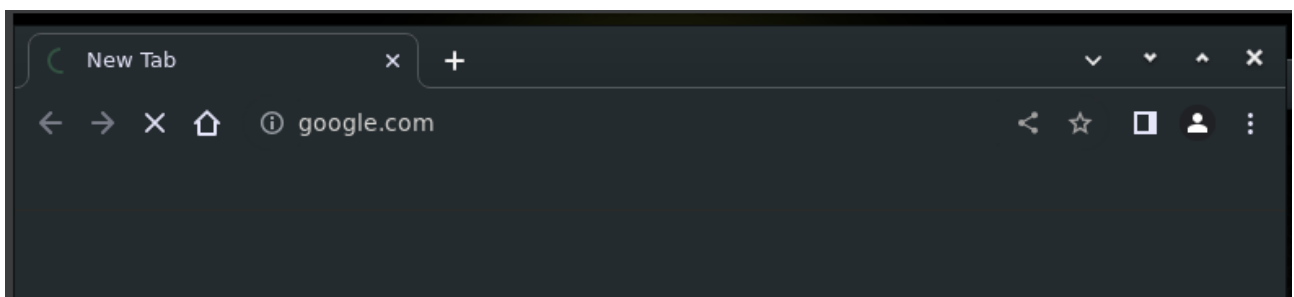


Figura 4: Tentativa de acesso a google.com com o *firewall* ativado.

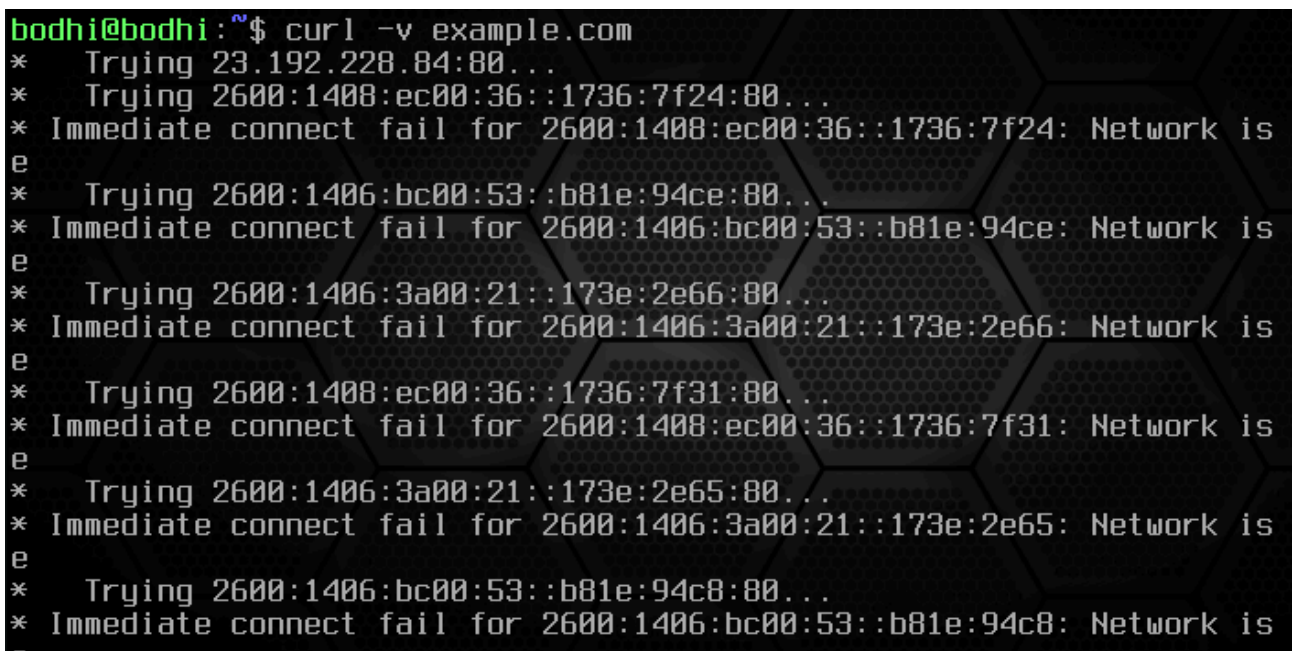


Figura 5: Tentativa de acesso a example.com com o *firewall* ativado.

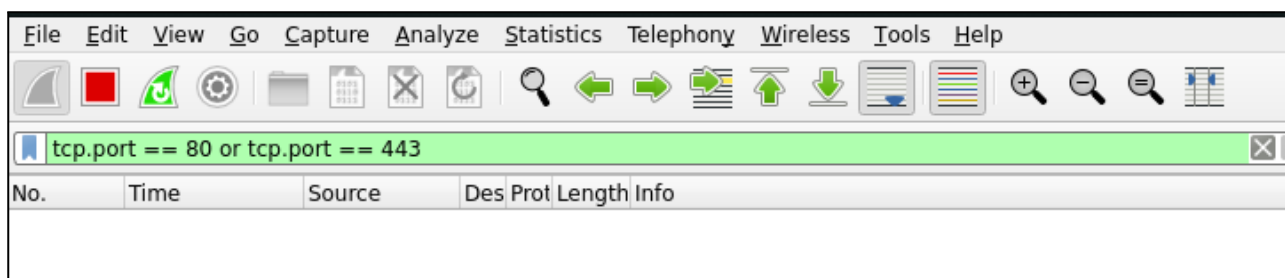


Figura 6: Captura do wireshark.

6. CONCLUSÕES

A aula prática proporcionou uma boa compreensão sobre a parte prática da configuração e funcionamento do *Uncomplicated Firewall* (UFW), e ajudou a entender mais como que um *firewall* decide quais tipos de pacotes devem ser bloqueados. Como esperado, o sniffer de rede wireshark não conseguiu capturar nenhum pacote transmitido pelas portas bloqueadas pelo firewall.

Conclui-se, portanto, que a prática foi bem-sucedida em atingir seus objetivos, consolidando o entendimento sobre firewalls, e mostrando que o UFW de fato é pouco complicado, como o próprio nome já diz.

BIBLIOGRAFIA

CAMPISTA, Miguel Elias M. et al. Interconexão de Redes na Internet do Futuro: Desafios e Soluções. **Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC**, v. 2010, p. 47-101, 2010.

CHANDRA, Deka Ganesh; KATHING, Margaret; KUMAR, Das Prashanta. A comparative study on IPv4 and IPv6. In: **2013 International Conference on Communication Systems and Network Technologies**. IEEE, 2013. p. 286-289.

CORONA, Adrián Estrada. *et al.* **Protocolos TCP/IP de internet**. 2004.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet: Uma abordagem top-down**. Trad. 8 ed. Sao Paulo: Francisco Araújo da Costa, 2021.

LEINER, Barry M. et al. **A brief history of the Internet**. ACM SIGCOMM computer communication review, v. 39, n. 5, p. 22-31, 2009.

MONTEIRO, Luís. **A internet como meio de comunicação: possibilidades e limitações**. In: Congresso Brasileiro de Comunicação. sn, 2001.

TANENBAUM, Andrew S. **Redes de Computadores**, 7ª Edição, Editora Campus, Rio de Janeiro – RJ, 2003.

UBUNTU. UFW. Ubuntu Community Help Wiki. Disponível em: <<https://help.ubuntu.com/community/UFW>>. Acesso em 4 de março de 2025.