Programação Distribuida Com redes usando Linux e Python

A yellow and blue snake and a white duck

Description automatically generated

Aluno: Ian Costa dos Santos

Turma: 24E2\_4

1. **Considere o encapsulamento de pacote e que você irá enviar a string “abcde” na Internet (TCP/IP) por meio de um socket e descreva:**
   * **O conteúdo do pacote enviado ao chegar na camada física de seu hospedeiro com todos os cabeçalhos e indicando o que é o segmento, o datagrama e o quadro;**
   * **O conteúdo do mesmo pacote assumindo o modelo de referência ISO/OSI.**

**R:**

Modelo TCP/IP:

* Ao enviar dados pela Internet usando TCP/IP, os dados são encapsulados em diferentes camadas:
* Camada de Aplicação: A string "abcde" é enviada pela aplicação. O TCP/IP encapsula esses dados em um segmento TCP.
* Camada de Transporte: O segmento TCP é então encapsulado em um datagrama IP.
* Camada de Rede: O datagrama IP é encapsulado em um quadro de dados.
* Camada Física: O quadro de dados é então transmitido fisicamente pelo meio de transmissão, como fios ou ondas de rádio.

A screenshot of a computer

Description automatically generated

Modelo OSI:

* Camada de Aplicação: Os dados são enviados pela aplicação.
* Camada de Apresentação: Os dados são formatados para transmissão, se necessário.
* Camada de Sessão: Estabelece, gerencia e encerra conexões entre aplicativos.
* Camada de Transporte: Os dados são encapsulados em segmentos.
* Camada de Rede: Os segmentos são encapsulados em datagramas.
* Camada de Enlace de Dados: Os datagramas são encapsulados em quadros.
* Camada Física: Os quadros são transmitidos fisicamente.

A screenshot of a computer

Description automatically generated

Assim, ao enviar a string "abcde" pela Internet, ela é encapsulada em diferentes camadas do modelo TCP/IP e do modelo OSI, garantindo que os dados sejam transmitidos de forma confiável e eficiente.

1. **Suponha um cenário que você perceba que a conexão do seu hospedeiro com a Internet foi perdida. A perda da conexão ocorreu devido a uma falha no ISP, porém, sem saber dessa informação, você realiza uma verificação do seu hospedeiro para garantir que o problema não ocorreu com ele. Um dos passos da verificação é checar a configuração da tabela de rotas.**
   * **Execute o comando para visualizar o IPv4 e IPv6 e indique o endereço IPv4 e máscara de subrede.**
   * **Execute o comando para visualizar a tabela de rotas e indique qual rota deveria estar faltando, ou deveria estar mal configurada, para que a conexão com a Internet fosse perdida.**
   * **A partir da máscara de subrede, indique a rota correspondente à sub rede.**

**R:**

1. Endereço IPv4: 10.0.2.15

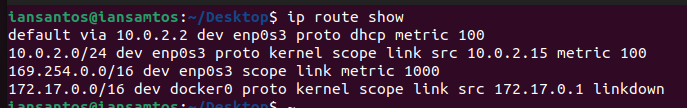
Endereço IPV6: fe80::7ae1:b728:8b79:5aa6/64

Máscara de sub-rede: /24 (corresponde a 255.255.255.0)

A screenshot of a computer

Description automatically generated

**b)** Para que a conexão com a Internet seja perdida, a rota que deveria estar faltando ou mal configurada é a rota padrão (default route). Esta rota direciona o tráfego que não pertence à rede local para o gateway, que depois direciona para a Internet.



c) A rota correspondente à subrede 10.0.2.0/24 é: “10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100”. Isso indica que a subrede 10.0.2.0/24 está acessível através do dispositivo de rede enp0s3 com endereço IP de origem 10.0.2.15.

1. **Forneça os endereços IP dos seguintes nomes de hospedeiros (nomes de rede), utilizado pelos menos dois comandos Linux diferentes: www.google.com, www.yahoo.com, localhost.**
   * **Remova todas as rotas da tabela de roteamento do seu hospedeiro e mostre que a conexão para o Gateway Default e o domínio www.google.com foi perdida.**
   * **A seguir, adicione uma rota para sub rede local e mostre que a conexão com o Gateway Default foi reestabelecida, porém a conexão para www.google.com ainda está perdida.**
   * **Por fim, escolha uma sub rede arbitrária que contenha o endereço IP de www.google.com, adicione uma rota para esta sub rede e mostre que a conexão com www.google.com foi reestabelecida.**
2. **Considere que você criou um novo serviço de transferência de arquivos chamado Transferência de Arquivo Texto (TAT), o qual executa na porta 12345 do seu hospedeiro. Como tal hospedeiro é utilizado com muita frequência, associe o número de porta 12345 ao nome tat. Para testar sua atribuição de porta, execute o comando lsof -i:tat e mostre que o comando retorna sem nenhum erro.**
3. **Descreva o funcionamento do protocolo DHCP e responda a seguinte pergunta:**
   * **Como um novo hospedeiro em uma LAN consegue obter endereço IP e máscara de subrede mesmo não conhecendo o endereço de nenhum outro nó e não conhecendo a subrede?**
4. **Descreva a diferença entre um hospedeiro linux configurado como roteador e um hospedeiro linux não configurado como roteador.**
5. **Assuma que você necessita logar em um hospedeiro que não possui interface gráfica. Tal hospedeiro necessita baixar a página inicial do Google em espanhol (es) e as únicas ferramentas disponíveis para realizar esta tarefa são o telnet e o curl. Para simular tal situação, realize o login remoto no localhost com o usuário alice e baixe com curl o conteúdo da URL www.google.com para o arquivo google\_es.html.**
6. **Execute as seguintes tarefas com o usuário alice e executando comandos ssh remotamente em dois terminais. No primeiro terminal, execute um servidor TCP com netcat no localhost na porta 12345. No segundo terminal, exiba a conexão do servidor TCP utilizando as ferramentas netstat e lsof.**
7. **Considere que você utilizou o comando netstat e verificou que o número de pacotes chegando no seu hospedeiro está muito maior do que o normal. Pela quantidade de dados, você suspeita que o tráfego pode ser HTTP ou uma conexão remota com telnet ou ssh. Descreva um comando tcpdump que seja capaz de capturar tais tipos de tráfego.**
8. **Você desenvolveu um cliente UDP na porta 12345 e deseja testá-lo rapidamente para corrigir alguns erros. Execute um servidor UDP com netcat que seja capaz de se comunicar com o cliente UDP.**
9. **Considere que sua empresa possui um servidor que envia arquivos-texto automaticamente para um outro hospedeiro localizado em uma rede externa. O arquivo schedule.txt contém a agenda de envio, onde cada linha possui os seguintes campos separados por vírgula: nome do arquivo a ser enviado sem espaços em branco, data de envio no formato DD-MM-AAAA, e horário de envio no formato mm:ss. Para que um arquivo-texto seja enviado, basta que ele seja armazenado na pasta send\_files, no diretório home do usuário bob, e que contenha uma entrada em schedule.txt.**
   * **Implemente uma aplicação Cliente-Servidor para transferência de arquivo-texto em que o cliente transfere um arquivo-texto para o Servidor, incluindo na mensagem a data e horário. Ao receber o arquivo, o Servidor salva-o no diretório send\_files e cria uma entrada em schedule.txt para que o arquivo possa ser transferido para a rede externa.**
   * **A aplicação deve utilizar apenas a biblioteca socket do Python e o código do servidor e do cliente devem ser implementados em servidor\_v0.txt e cliente\_v0.txt, respectivamente.**
   * **O arquivo armazenado na pasta send\_files deve possuir exatamente o mesmo conteúdo do arquivo enviado pelo cliente.  
     Obs.: Você não deve implementar o servidor de envio para rede externa, apenas a aplicação Cliente-Servidor.**
10. **A partir do exercício anterior, implemente um serviço de autenticação na aplicação Cliente-Servidor nos arquivos servidor\_v1.py e cliente\_v1.py. O servidor possui um arquivo chamado users.txt em que cada linha contém o nome do usuário e a senha separados por vírgula. Antes de enviar o arquivo, o cliente deve enviar suas credenciais (login e senha) para o servidor autenticá-lo. O servidor deve procurar as credenciais do cliente em users.txt e enviar uma mensagem autorizando ou negando acesso ao cliente.  
    Obs.: A única biblioteca permitida é a socket do Python.**
11. **Considere que sua empresa possui 10 servidores que armazenam na pasta internet\_data arquivos HTML contendo dados tabulados obtidos de fontes na Internet. Para um usuário visualizar tais arquivos, ele baixa os arquivos de algum dos 10 servidores e os abre no navegador. Para otimizar tal processo, você foi encarregado de implementar um servidor http no arquivo check\_servers.py rodando na porta 12345 e deixá-lo executando na pasta internet\_data. Dessa forma, quando alguém quer acessar tal dado basta digitar no navegador endereco\_ip\_servidor:12345/nome\_arquivo.html. Implemente o servidor HTTP utilizando apenas a biblioteca socket do python. Teste o Servidor HTTP com o navegador utilizando o arquivo test.html cujo conteúdo é descrito abaixo:  
    <html><body>  
    <h1>Dado Teste</h1>  
    <table>  
    <tr><td>titulo1</td><td>titulo2</td><td>titulo3</td></tr>  
    <tr><td>dado11</td><td>dado12</td><td>dado13</td></tr>  
    </table>  
    </body></html>**
12. **Implemente no arquivo https.py uma função chamada get\_https\_page que recebe como entrada uma url e retorna o documento HTML correspondente à url. Seu script Python deve utilizar apenas as bibliotecas socket e ssl e deve baixar o documento com o protocolo HTTPS na porta 443.**
13. **Considere que você gerencia um cluster de servidores e com frequência necessita verificar se eles estão ativos na rede e realizando login remoto com ssh. Para automatizar este processo, você resolve implementar um script python que utiliza a biblioteca scapy para verificar se todos os hospedeiros em uma lista de endereços IP estão ativos e se a porta 22 (ssh) está aberta.**
    * **Implemente tal script e teste-o utilizando uma lista contendo o IP local e o IP do Gateway Padrão. Seu script deve ser implementado no arquivo check\_hosts.py e deve utilizar apenas a biblioteca scapy.**
    * **Descreva qual a principal diferença da implementação do seu script com scapy para uma implementação com a biblioteca pcapy-ng.**
14. **Escreva um programa Python no arquivo check\_download.py que verifica se usuários da LAN estão fazendo download de conteúdo Web por meio do protocolo HTTP. Seu script deve capturar apenas os pacotes que chegam de algum Servidor Web HTTP para os hospedeiros da LAN. Ao capturar um pacote, você deve imprimir o endereço IP do servidor. Seu script deve utilizar apenas a biblioteca scapy.**
15. **Você suspeita que seu Gateway está com a Tabela ARP falsificada. Descreva como você pode identificar tal Falsificação ARP utilizando um script Python com scapy.**
16. **Com o objetivo de possuir um módulo em Python que forneça endereços IPv4 e nomes de servidores de e-mail para uma dada URL. Implemente o script resolver.py o qual possui as funções url2ip e url2email, as quais recebem uma URL como entrada e retornam o endereço IPv4 e os nomes de servidores de email (MX), respectivamente. A função url2ip deve ser implementada com a biblioteca dnspython e a função url2email deve ser implementada com a biblioteca dnsrecon.**
17. **Implemente no arquivo fuzzer.py um script de pentest que testa todas as URLs previsíveis para o domínio de teste testphp.vulnweb.com e com caminhos contidos no arquivo deste link “<https://github.com/fuzzdb-project/fuzzdb/blob/master/discovery/predictable-filepaths/login-file-locations/Logins.txt>”.**
18. **Implemente no arquivo asyncnmap.py um script de varredura de portas assíncrono utilizando a biblioteca nmap do Python. Você deve executar o script para realizar a varredura nas seguintes URLs: www.example.com e testphp.vulnweb.com.**
19. **Considere que você suspeita que seu hospedeiro pode ter sido acessado indevidamente por outro usuário e decide checar se existem conexões abertas por código malicioso ou backdoors. Execute o comando nmap para realizar tal verificação.**