Projeto Final - DDOS

Universidade de Brasília
Departamento de Ciência da Computação
Teleinformática e Redes 2
Prof. Jacir Luiz Bordim
Edgar Fabiano de Souza Filho - 14/0019201
Ingrid Santana Lopes - 14/0083065
Marcos Paulo Cayres Rosa - 14/0027131
Paulo Victor Gonçalves Farias - 13/0144754
Rennê Ruan Alves Oliveira - 14/0030930

***Resumo*—Este relatório tem como objetivo explicar dois tipos de ataques DDOS e como eles podem ser implementados em um ambiente de teste. O primeiro ataque é o SYN Flood, que está relacionado com o envio de uma grande quantidade de pacotes TCP SYN para iniciar uma conexão em um servidor. Esse volume de dados tem como intenção exaurir a capacidade do servidor da vítima e causar a negação de serviço. O segundo ataque testado é o amplificação de DNS, este se refere ao uso de servidores DNS para o envio de várias requisições para um IP de uma vítima, ele também gera um grande volume de dados que deve exaurir a capacidade do servidor, também causando uma negação de serviço. Os dois ataques foram implementados em uma WLAN (*Wireless Local Area Network*) para testes. Estes ataques foram feitos de modo sincronizado e distribuído. Obteu-se êxito no experimento e seus resultados serão comentados.**

I. SYN Flood

Quando uma conexão é estabelecida entre dois hosts (servidor e cliente) usando o protocolo TCP (*Transfer Control Protocol*), quer dizer que ocorreu um procedimento, *three-way handshake*, com sucesso. Esse procedimento ocorre da seguinte forma: primeiramente um cliente envia uma requisição para o servidor. Essa requisição não deve conter nenhuma informação da camada de aplicação, mas ele deve mudar uma flag no cabeçalho do segmento chamada SYN para 1. Por isso esse segmento se chama de segmento TCP SYN.

Quando o servidor recebe essa nova requisição ele aloca buffers TCP e outras variáveis para a conexão, com isso ele pode enviar um segmento para o cliente indicando que ele pode iniciar a conexão. Esse segmento é especial pois ele também não deve conter nenhuma informação da camada de aplicação. O cabeçalho dele também deve conter algumas informações especiais. A flag SYN deve ser setada como 1. O campo do ACK também deve ser setado. Esse segmento é chamado de SYNACK.

O terceiro passo é a parte em que o cliente recebe um segmento SYNACK do servidor. Esse cliente também deve alocar buffers e variáveis para essa conexão. Com isso ele estabelece a conexão com o servidor depois de ter enviado mais um segmento, esse último apenas diz para o servidor que ele recebeu o segmento SYNACK.

O ataque **Syn Flood** é um ataque DoS (*Denial of Service*) ou seja, ele busca impedir que usuários tenham acesso a um servidor, ou seja, impede o serviço. Esta estratégia em particular busca explorar essa característica do protocolo TCP. Ao invés de enviar apenas uma requisição de conexão, ele envia várias requisições com intenção de exaurir a capacidade do servidor. Para fazer isso ele nunca responde ao SYNACK do servidor e continua enviando requisições. O servidor espera um minuto ou mais para fechar essas conexões onde ele não recebeu o ACK final. Um usuário de fora que tente se conectar não terá sucesso, já que o servidor não tem capacidade de alocar recursos para essa nova conexão.

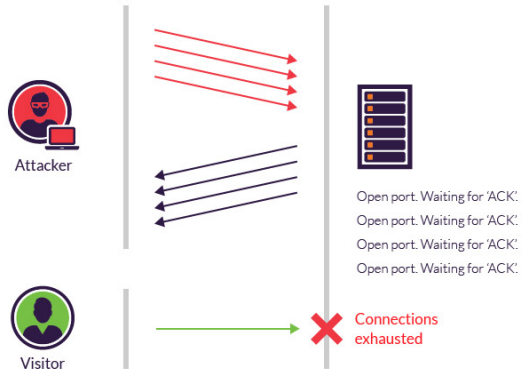

Figura 1: Ataque SYN FLOOD.

II. Amplificação de DNS

Quando um usuário deseja se conectar com um site na Internet, ele deve usar o protocolo DNS (*Domain Name System*), o DNS é responsável por resolver nomes em endereços IP. A partir de um nome, ele deve encontrar o IP de destino na rede. Para isso existem diversos servidores DNS espalhados pelo mundo, com o propósito de resolver esses nomes. É importante notar que existem diferentes tipos de servidores DNS e uma hierarquia existente. Um servidor DNS não contém todos os endereços IP do mundo. O usuário pode realizar requisições iterativas e recursivas para um servidor DNS. As requisições iterativas permitem que o usuário seja redirecionado para outro servidor DNS que possa ter a referência daquele nome que ele está buscando. Com as requisições recursivas o usuário força o

servidor a fazer a busca desse nome entre os outros servidores DNS e apenas aguarda a resposta.

O ataque de amplificação DNS busca explorar a capacidade desses servidores DNS para atacar uma vítima com um determinado IP e causar a negação de serviço. Quando um usuário legítimo deseja saber o endereço IP de um nome na Internet, ele deve perguntar ao servidor DNS. Ele recebe uma resposta do servidor DNS sobre o IP do endereço requisitado. O ataque de amplificação de DNS tem como objetivo criar requisições para o servidor DNS com o IP da vítima, de forma que ela receba todas as respostas do servidor DNS. O atacante pode usar diferentes máquinas para realizar o ataque de forma sincronizada e distribuída, de modo que o servidor DNS cria uma grande quantidade de respostas DNS para a vítima. O servidor da vítima não consegue lidar com o número de dados que está chegando e para de funcionar. Isso gera uma negação de serviço para todos os novos clientes que tentem se conectar com o servidor da vítima.

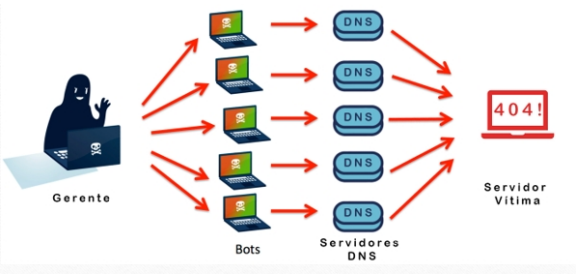

Figura 2: Ataque amplificação de DNS.

III. WLAN, Servidor Apache e Servidor DNS

O primeiro passo para a simulação desses ataques foi criar uma rede Wireless LAN(*Local Area Network*). Esta rede foi criada com um roteador comum, que obedece os padrões IEEE 802.11. Por questões legais, é importante criar um ambiente privado para realizar esses testes, pois ataques DDOS podem causar danos reais. No caso do ataque de amplificação de DNS, não poderemos usar os servidores DNS reais para criar esse ataque. Por esse motivo foi necessário criar um servidor DNS adicional. Com a rede local também foi possível permitir que os diferentes servidores e clientes usados nesse experimento se conectem à rede livremente.

O segundo passo foi criar um servidor Apache, esse servidor representará o servidor da vítima que será atacada. Servidores Apache são usados em larga escala, sendo o servidor web livre mais usado no mundo todo. Ele pode ser instalado com o seguinte comando no terminal Linux Debian:

```
sudo apt-get install apache2
```

Os arquivos de configuração no Linux estão no diretório /etc/apache. Após instalado e iniciado, o servidor apache funciona no endereço de Loopback (127.0.0.1), logo foi necessário realizar algumas modificações em seus arquivos para que ele fique dispoível para os diferentes usuários da rede. O apache apresenta a seguinte página quando iniciado:

Para que ele possa servir em um endereço diferente, é necessário editar uma série de arquivos e especificar a porta

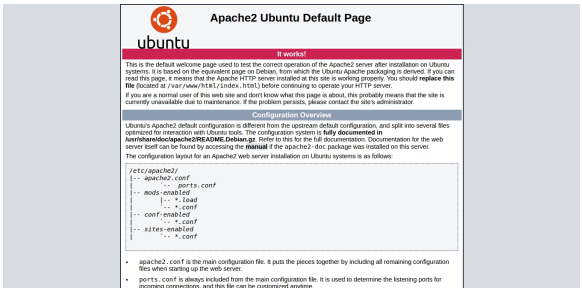

Figura 3: Servidor Apache

que eles devem ouvir, alguns exemplos podem ser vistos nas figuras a seguir.

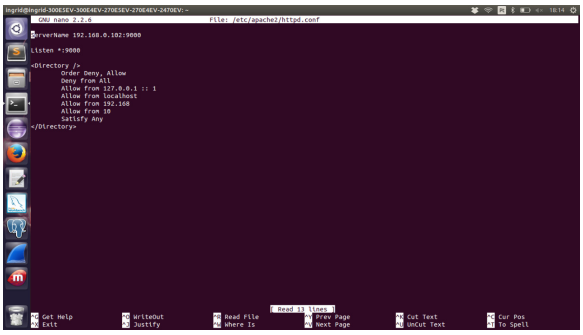

Figura 4: Configuração do arquivo HTTPD

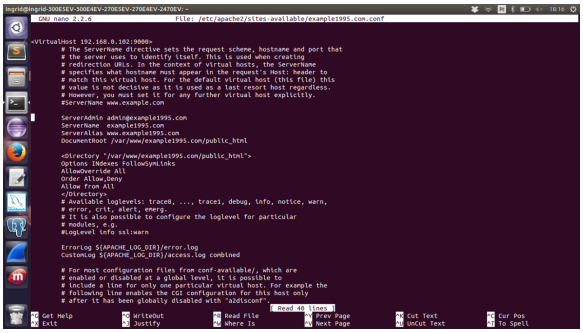

Figura 5: Configuração do arquivo CONF

Com isso já é possível realizar o ataque de Syn Flood. Para realizar o ataque de amplificação de DNS é necessário criar um servidor DNS que possa responder corretamente *querries* específicas para caracterizar a amplificação. O servidor DNS precisava ser local, funcionar de maneira recursiva e poder retornar todos os subdomínios de um domínio. Para a criação do mesmo foi utilizado o BIND9, o BIND precisa ser instalado e configurado para que o IP da máquina seja o que o servidor DNS deverá responder, além disso, utilizando as configurações do BIND foi permitido a recursão para todas as *querries*.

Após a configuração do DNS é preciso criar zonas de encaminhamento para que possam ser realizadas as buscas, cada zona de encaminhamento irá ter um arquivo de configuração, em nosso projeto foi utilizado apenas uma zona de encaminhamento entitulada *tr2.com*. Presente no arquivo de

```
options {
    directory "/var/cache/bind";

    recursion yes;
    allow-recursion { any;};
    listen-on {192.168.0.10;};
    allow-transfer {none;};
    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

     forwarders {
        8.8.8.8;
        8.8.4.4;
     };

    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //========================================================================
    dnssec-validation auto;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

Figura 6: Configuração DNS

configuração da zona de encaminhamento dita anteriormente temos informações acerca dos subdomínios presente na zona, serão gravados diversos *records* de tipo A, com a presença do nome e do IP de cada subdomínio. É preciso formar um pacote específico para realizar a *query* que o servidor DNS consiga responder, com isso utilizamos Raw Sockets para montar as *queries* específicas para *ns1.tr2.com*.

```
$ORIGIN tr2.com.
$TTL    86400
@       IN  SOA ns1.tr2.com. admin.tr2.com. (
                      3     ; Serial
                 604800     ; Refresh
                  86400     ; Retry
                2419200     ; Expire
                  86400 )   ; Negative Cache TTL
;
@       IN  NS  ns1.tr2.com.
```

Figura 7: Configuração do domínio DNS utilizado

IV. Distribuição e Sincronização dos Ataques

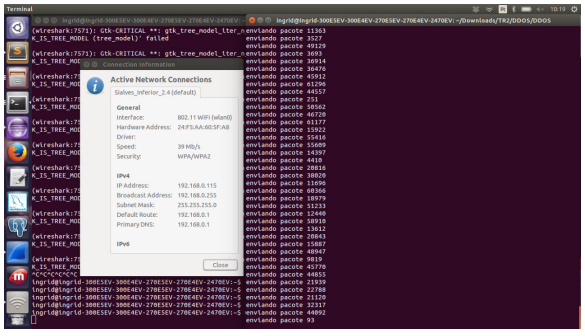

Figura 8: Bot realizando um ataque DOS

A distribuição foi feita a partir de um programa chamado 'bot.py'. Este programa representa um processo zumbi que está sendo executado em uma máquina infectada. Sendo assim, pode-se obter diferentes dispositivos que estão a disposição de um gerente que comanda esses programas. O gerente é uma outra máquina que é responsável por sincronizar o ataque e manter controle dos bots conectados. Toda vez que um desses bots é iniciado, o gerente sabe que ele está ativo e pode ser usado para realizar o ataque. O processo de distribuição e sincronização é feito da mesma forma para os dois ataques. O que muda é o código do bot, um ataque SYN Flood deve usar requisições TCP enquanto um ataque de amplificação DNS deve fazer requisições sobre o DNS, logo foi ncessário criar programas diferentes.

V. Resultados Obtidos

A. Syn Flood

O código distribuído para o SYN Flood é o 'botsyn.py'. Ele representa um programa malicioso que deve ser capaz de enviar pacotes TCP SYN. Para o SYN Flood, os bots devem ser capazes de usar RAW SOCKETS para enviar pacotes TCP SYN. Esses pacotes indicam ao servidor que o cliente deseja iniciar a conexão. Os pacotes TCP/IP tem a estrutura abaixo. Pode-se perceber que ele esta enviando diferentes pacotes com diferentes endereços de IP, como ele nunca irá receber os SYNACKs desses endereços, ele consegue exaurir a capacidade do servidor com esse grande número de pacotes SYN. A única flag setada é o SYN = 1, o que quer dizer para o servidor que ele quer iniciar uma nova conexão.

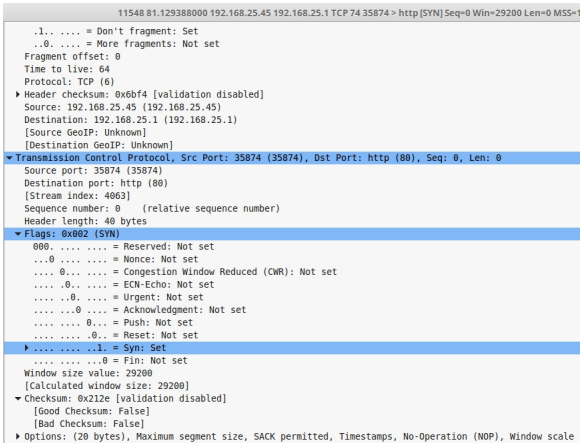

Figura 9: Estrutura do pacote para Syn Flood

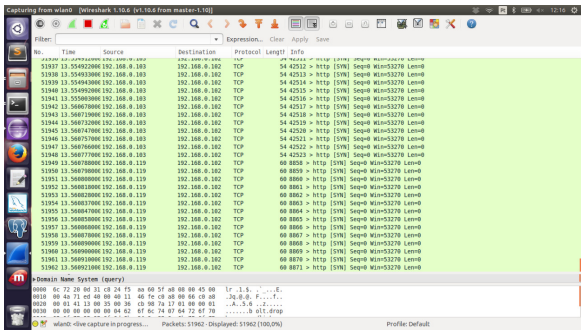

Figura 10: Captura dos pacotes TCP SYN

O IP da vítima é o 192.168.0.102 como indicado na figura, pela captura dos pacotes pelo Wireshark foi possível visualizar

os diferentes pacotes chegando. Esses pacotes estão sendo enviados pelos vários bots comandados pelo gerente. Esses bots estão executando um programa malicioso que representa um processo zumbi em uma máquina infectada. O servidor Apache criado, parou de funcionar em alguns momento após o ataque. Foi possível também perceber que as requisições para o endereço IP aonde ele estava hospedado levavam bastante tempo antes de conseguir se conectar.

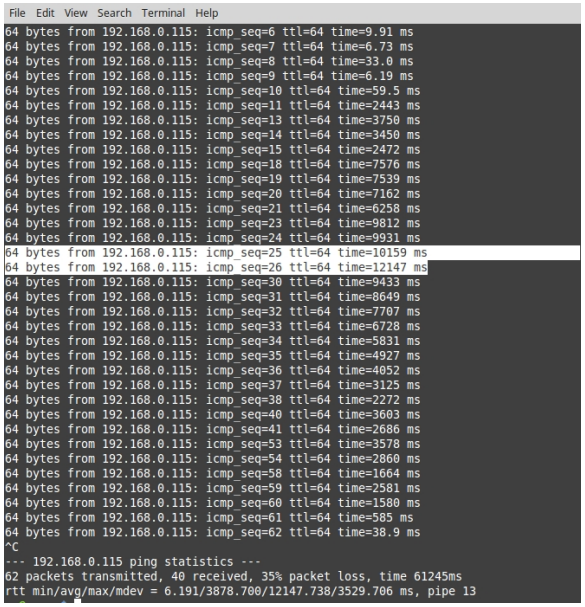

Figura 11: Ping no servidor atacado

Foi utilizado também o ping para garantir que o ataque estava causando danos reais ao servidor, sendo possível notar que em alguns casos houve a demora de até mais de 10 segundos. E ainda foi possível perceber uma perca de 35 porcento dos pacotes.

B. Amplificação de DNS

Depois que o ataque SYN Flood, foi feito com êxito foi necessário testar o ataque de amplificação de DNS. Como indicado na seção III, o servidor DNS foi criado utilizando uma ferramenta chamada bind9, e com isso foi possível receber querys e enviar respostas sobre os (*Resource Records*). Este ataque foi feito da mesma maneira que o anterior, mas ao invés dos bots enviarem as requisições para o servidor da vítima, eles devem enviar todas as requisições para o IP do servidor DNS, que neste experimento foi o 192.168.0.106.

As requisições DNS enviadas usam o protocolo da camada de transporte UDP, assim como em um DNS real. Elas são enviadas utilizando a porta 53 também referente ao DNS. Após o gerente indicar que os bots deviam começar o ataque eles começam a enviar um grande número de requisições para o servidor DNS criado. O servidor DNS por sua vez, vê esses pacotes e os envia para o endereço de destino, que representa o IP da vítima. O ataque teve sucesso em parte, pois os pacotes chegaram como pode ser visto na captura de pacotes a seguir:

O servidor Apache não parou de funcionar como no ataque SYN Flood. Isso ocorre pois ele sempre ignora todos os

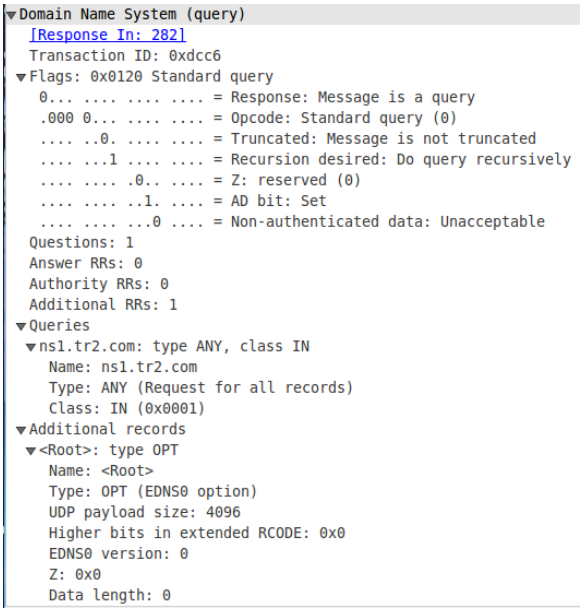

Figura 12: Estrutura de query DNS

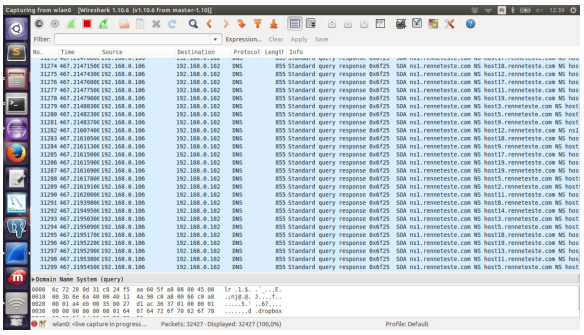

Figura 13: Captura dos pacotes DNS

pacotes UDP. Isso é uma característica do próprio servidor, ele só aceita pacotes TCP. Por isso não foi possível checar se o servidor da vítima pararia de funcionar com esse ataque.

Uma nova tentativa foi feita utilizando um outro servidor, mas agora outro servidor DNS. Este servidor DNS em python recebe simples requisições e mostra em um terminal o que ele recebeu. d

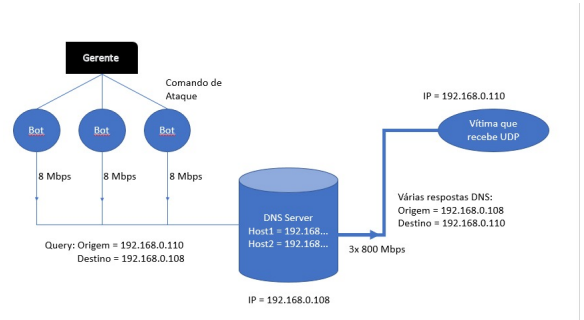

Figura 14: Topologia da Rede

A figura 14 mostra como funciona a topologia geral da rede. Primeiramente existe um Gerente que comanda diferentes bots, esses bots enviam requisições para um servidor DNS com uma taxa de 8Mpbs assim que o Gerente informa que eles devem começar o ataque. Ao invés de mandar o ataque diretamente para um servidor, o objetivo aqui é mandar *querys* com o endereço de origem modificado. Por isso o IP de Origem deve ser o da vítima.

Ao receber esse grande número de *querys*, o servidor DNS irá enviar essas informações para aquele endereço de IP de origem, mesmo não sendo ele quem tenha feito aquele ataque. Com apenas três bots na rede, a vítima vai receber 2400Mbps. Como o servidor da vítima é também um servidor DNS e ele funciona utilizando o protocolo UDP, as requisições não são ignoradas. Isso pode ser visto na figura abaixo, onde um usuário faz uma requisição de um endereço exemplo.com com o comando dig.

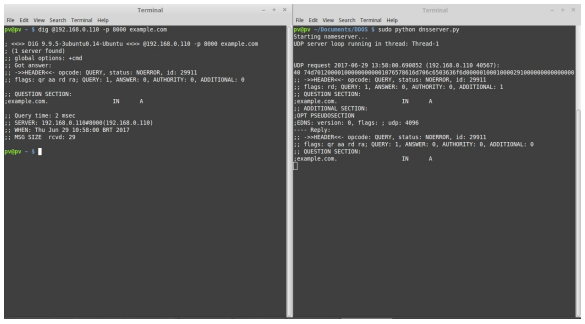

Figura 15: Servidor DNS Vítima

Depois disso foi feito o teste do ataque de amplificação de DNS. Assim que o gerente indicou o início do ataque os bots enviaram vários pacotes como mostrado na figura 8. Com esse grande número de pacotes sendo enviados, o servidor DNS em python (a vítima), parou de receber as requisições feitas utilizando o dig, como pode também ser visto abaixo. Sendo assim o ataque teve sucesso, pois negou o serviço de um servidor.

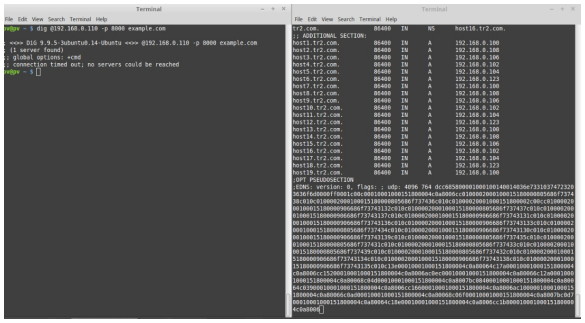

Figura 16: Servidor DNS Vítima, com o dig falhando

VI. Conclusão

Com o fim deste projeto, foi possível aprender o que são ataques DDOS, em especial o ataque de SYN Flood e o de amplificação de DNS. É importante notar que o SYN Flood. Foi possível analisar como esses ataques podem ser feitos efetivamente e como eles podem funcionar de forma distribuída e sincronizada, o que foi notado após realizar o ataque em um ambiente real. O projeto agregou também conteúdos sobre servidores DNS, servidores Apache, redes WLAN e os protocolos da camada de transporte TCP e UDP.

Referências

[1] Config. Servidor Apache
https://www.digitalocean.com/community/tutorials/como-configurar-apache-virtual-hosts-no-ubuntu-16-04-pt

[2] Config. Servidor DNS com Bind9
https://www.digitalocean.com/community/tutorials/como-configurar-apache-virtual-hosts-no-ubuntu-16-04-pt

[3] Ataque SYN Flood com raw sockets
http://binarytides.com/tcp-syn-flood-dos-attack-with-hping/

[4] Criar um servidor DNS em Python
https://www.youtube.com/watch?v=HdrPWGZ3NRo

[5] Bibiloteca dnslib
https://pypi.python.org/pypi/dnslib