Universidade de Brasília Faculdade UnB Gama

Laboratório sobre DNS, SMTP + Cliente de e-mail e WWW

Fundamentos de Redes de Computadores - Turma 1

Prof. : Fernando William Cruz

Engenharia de Software

Alunos

Ana Carolina Rodrigues Leite - 19/0101792

Ricardo de Castro Loureiro - 20/0043111

1. Introdução

A proposta deste laboratório é construir uma intranet com os principais serviços de uma intranet TCP/IP.

DNS (Domain Name System) é um sistema fundamental da Internet que traduz os human-readable domain names (nomes de domínio) em endereços IP numéricos, permitindo que os dispositivos conectados à rede se comuniquem e encontrem recursos online de forma eficiente.

SMTP (Simple Mail Transfer Protocol) é um protocolo utilizado para o envio de e-mails entre servidores e clientes de e-mail. Ele garante a entrega confiável e apropriada das mensagens, permitindo que usuários de todo o mundo se comuniquem por meio de correio eletrônico.

HTTP (Hypertext Transfer Protocol) é o protocolo base da World Wide Web. Ele permite a transferência de dados entre um cliente (geralmente um navegador) e um servidor web, possibilitando o acesso a páginas da web, imagens, vídeos e outros recursos online de maneira rápida e organizada.

Esses três elementos, DNS, SMTP e HTTP, são essenciais para o funcionamento da Internet moderna, possibilitando a navegação na web, o envio de e-mails e a localização de servidores e recursos online de forma transparente para os usuários.

2. Perguntas do Lab de DNS

1. Qual é o retorno do comando dnsdomainname? O que significa?

   R: O comando dnsdomainname retorna o nome do domínio DNS do sistema. Significa que é o nome do domínio pelo qual o sistema é conhecido na rede. No entanto, em alguns sistemas operacionais, como em algumas distribuições Linux, esse comando pode estar desativado ou não retornar nada, pois depende da configuração específica do sistema.

2. O que é o nome localhost? E o endereço 127.0.0.1 dado a ele no arquivo /etc/hosts? Por que deve sempre existir este endereço e nome em sistemas UNIX/Linux?

   R: O nome "localhost" é uma convenção utilizada para referenciar o

próprio computador ou dispositivo local. O endereço IP 127.0.0.1 é o endereço loopback do dispositivo, ou seja, sempre se refere ao próprio dispositivo. Essa configuração é essencial em sistemas UNIX/Linux porque permite que serviços e aplicativos internos se comuniquem consigo mesmos sem depender de uma conexão de rede externa, tornando o funcionamento local do sistema mais independente e confiável.

3. O que é o FQDN?

   R: O FQDN (Fully Qualified Domain Name) é o nome completo de um host na Internet, incluindo o nome do host e o nome de domínio. Por exemplo, um FQDN seria "host.example.com". Ele fornece o caminho completo para localizar o host na hierarquia do sistema de nomes de domínio.

4. Podemos ter 2 servidores DNS na mesma rede? Qual é a configuração mais adequada para esta situação?

   R: Sim, é possível ter dois servidores DNS na mesma rede. Na configuração mais adequada para essa situação, ambos os servidores DNS devem ser configurados como autoritativos para a mesma zona de domínio, mas cada servidor deve ter registros DNS diferentes. Isso ajuda a equilibrar a carga de consultas DNS e aumentar a disponibilidade da resolução de nomes.

5. O que é DNS reverso? Como isso foi implementado no lab?

   R: O DNS reverso é uma técnica que mapeia endereços IP para nomes de domínio. Isso é útil para obter o nome do domínio associado a um determinado endereço IP. No laboratório, o DNS reverso pode ser implementado criando uma zona especial chamada "in-addr.arpa" (para IPv4) ou "ip6.arpa" (para IPv6) no servidor DNS e adicionando registros PTR para mapear endereços IP para nomes de domínio.

6. O que é a entrada MX inserida no domínio? Pode haver mais de uma?

   R: A entrada MX (Mail Exchange) em um domínio especifica os servidores de e-mail responsáveis por receber e encaminhar e-mails

para esse domínio. Pode haver mais de uma entrada MX com diferentes prioridades. Quando um servidor de e-mail tenta entregar uma mensagem para o domínio, ele consulta os registros MX para determinar para qual servidor enviar o e-mail.

7. O que é resposta autoritativa dada por um servidor DNS? Explique.

   R: Uma resposta autoritativa dada por um servidor DNS é aquela em que o servidor possui autoridade direta sobre a zona de domínio consultada. Significa que o servidor é o detentor das informações sobre o domínio e fornece uma resposta confiável e oficial. Em contraste, uma resposta não autoritativa é quando o servidor consulta outro servidor DNS para obter a resposta.

8. O que é um servidor caching-only.

   R: Um servidor caching-only é um servidor DNS configurado para responder apenas a consultas de cache. Ele não é autoritativo para nenhuma zona de domínio, mas armazena em cache as respostas de consultas previamente feitas. Isso permite que os clientes obtenham respostas mais rápidas para consultas frequentes e reduz a carga nos servidores DNS autoritativos. É uma configuração comum para melhorar a eficiência e a velocidade do sistema de resolução de nomes.

3. Perguntas do Lab de SMTP

1. Descrever as principais modificações realizadas nos arquivos /etc/postfix/main.cf e /etc/postfix/master.cf e os efeitos de cada modificação. Anexar os arquivos no relatório.
2. Qual a diferença entre o armazenamento de e-mails no formato mbox e maildir? Instale o maildir e descreva quais os procedimentos de instalação de um e outro caso.

   R: A diferença entre o armazenamento de e-mails no formato mbox e maildir reside na organização dos dados. O formato mbox utiliza um único arquivo para cada caixa de correio, com todos os e-mails concatenados, o que pode tornar a leitura e escrita lenta para muitas

mensagens. Por outro lado, o maildir usa diretórios individuais para cada e-mail, o que possibilita acesso mais eficiente e evita problemas de travamento comuns no formato mbox. Embora o maildir seja mais robusto em situações de leitura/gravação simultânea, ele pode consumir mais inodes, o que pode ser uma preocupação em sistemas com limitações nessa estrutura.

3. Para que servem os esquemas de autenticação SASL/TLS. Caso tenha instalado, descreva qual foi a solução adotada(Dovecot ou outro) e explique os passos de instalação.

   R: Os esquemas de autenticação SASL (Simple Authentication and Security Layer) e TLS (Transport Layer Security) são utilizados para aumentar a segurança na comunicação e autenticação em servidores de e-mail. Vamos explicar brevemente para que servem:

   Autenticação SASL:

   - O SASL é um framework que permite autenticação e segurança para diversos protocolos, incluindo SMTP (para envio de e-mails) e IMAP/POP3 (para acesso aos e-mails).
   - Com o SASL, os clientes de e-mail precisam fornecer credenciais (nome de usuário e senha) para se autenticarem no servidor antes de enviar ou acessar os e-mails.
   - Isso ajuda a evitar o uso indevido do servidor por spammers e aumenta a segurança na entrega de e-mails.

   TLS (Transport Layer Security):

   - O TLS é um protocolo de criptografia que fornece segurança na comunicação entre o cliente e o servidor.
   - Quando o TLS é ativado, a comunicação entre o cliente de e-mail e o servidor ocorre de forma criptografada, protegendo as informações, como senhas e conteúdo de e-mails, de possíveis interceptações por terceiros.

4. Qual é a diferença entre os formatos RFC822 e MIME types definidos para e-mails? Explicá-los.

   R: O formato RFC822 é uma especificação mais antiga que define a estrutura básica dos e-mails, mas não possui suporte nativo para anexos ou conteúdo não textual. Por outro lado, o MIME é uma extensão do formato RFC822 que permite a inclusão de anexos e suporta diversos tipos de conteúdo através dos MIME types, tornando o envio e recebimento de e-mails mais versátil e adequado para lidar com diferentes tipos de dados e mídias.
5. Faça uma conexão telnet no servidor SMTP e envie um e-mail para qualquer usuário cadastrado utilizando os comandos do protocolo (HELO, MAIL FROM, RCPT TO, DATA, QUIT, etc.) Anotar os resultados dessa experiência.
6. Faça uma conexão telnet no servidor POP e receba e-mails utilizando os comandos deste protocolo (USER, PASS, LIST, RETR e QUIT). Anotar os resultados dessa experiência.

4. Perguntas do Lab de WWW

1. Promova alterações no servidor para que o servidor consiga publicar páginas de uso geral e páginas para cada usuário (via uso de pasta pessoal - public_html). Relatar aqui todas as modificações feitas para viabilizar este serviço.
2. Promova alterações no servidor para que o mesmo faça controle de restrições de acesso a determinados diretórios com uso de diretivas do próprio servidor (inclui uso do htaccess). Por exemplo, o diretório public_html de cada usuário só deve ser acessado para escrita (Read/Write) pelo próprio dono e pode ser lido (Read Only) pelos demais.
3. Promova alterações no servidor Web para que o mesmo comporte domínios virtuais. Relacione as modificações realizadas aqui. Nesse caso, crie um domínio virtual para cada aluno do grupo e coloque uma cópia do relatório individual como conteúdo desses domínios virtuais. Obs: Provavelmente será necessário inserir novas diretivas no DNS. Exemplo: www.alunos.com.br dá acesso ao domínio real (página principal) do grupo. Essa página deve conter as informações gerais do trabalho e links para os domínios virtuais de cada um dos alunos do grupo. Exemplo: O link www.zedasilva.com.br é um domínio virtual

relativo ao Zé da Silva que faz parte do grupo. Nesse domínio deverá estar o relatório individual deste aluno.

4. O que é Server Side Includes (SSI)? Promova alguma alteração no servidor mostrando a funcionalidade desse mecanismo.

   R: Server Side Includes (SSI) é uma tecnologia que permite incluir conteúdo dinâmico em páginas web, diretamente no lado do servidor, antes que a página seja enviada ao cliente (navegador). Ele é geralmente usado para inserir informações repetitivas, como cabeçalhos, rodapés, menus, ou para realizar cálculos ou exibir dados dinâmicos.

5. Quais são os mecanismos implementados por um servidor Web para conseguir atender tantas conexões simultaneamente? Se houver diretivas de configuração, relacioná-las aqui.

   R: Um servidor web é projetado para lidar com várias conexões simultâneas e atender a solicitações de clientes de forma eficiente. Isso é alcançado através de diferentes mecanismos:

   1. Modelo de processo/forking: Cria um novo processo para cada conexão, o que pode ser custoso em recursos, mas simples de implementar.

   2. Modelo de thread: Cria um novo thread para cada conexão, compartilhando o espaço de memória, tornando-o mais eficiente em recursos do que o modelo de processo.

   3. Modelo de eventos assíncronos: Usa eventos assíncronos e loops de eventos para monitorar atividades de entrada/saída e atender várias conexões usando um único thread, economizando recursos.

   4. Pool de processos ou threads: Mantém um número fixo de processos ou threads disponíveis para reutilização, reduzindo o custo de criação de novos e otimizando o uso de recursos.

   5. Limitação de recursos por conexão: Configurar limites para recursos por conexão para evitar o uso excessivo de recursos por uma única conexão.

Esses mecanismos permitem que os servidores web atendam um grande número de clientes simultaneamente de forma eficiente e responsiva.

6. Explique qual é a utilidade e como funciona um proxy server no caso do protocolo HTTP.

   R:

   Um proxy server é um intermediário entre o cliente (navegador) e o servidor web no protocolo HTTP. Ele melhora o desempenho ao armazenar em cache respostas frequentes, protege a privacidade ocultando o endereço IP do cliente e oferece controle de acesso e segurança filtrando e inspecionando o tráfego. O proxy recebe as solicitações, verifica a cache, aplica políticas, envia a solicitação ao servidor de destino e entrega a resposta ao cliente, agindo como um intermediário eficiente e seguro no processo de comunicação HTTP.

   Como funciona um proxy server no protocolo HTTP:

   1. Quando um cliente (navegador) faz uma solicitação HTTP para acessar um recurso na web, o pedido é encaminhado primeiro para o proxy server em vez de ser enviado diretamente ao servidor web de destino.
   2. O proxy server examina a solicitação e, dependendo da sua configuração, pode tomar várias ações, como verificar a cache para ver se a resposta já está armazenada, aplicar políticas de acesso, verificar a segurança da solicitação ou simplesmente encaminhar a solicitação ao servidor web de destino.
   3. Se a resposta estiver em cache, o proxy server entrega-a diretamente ao cliente, economizando tempo e largura de banda.
   4. Caso contrário, o proxy server envia a solicitação ao servidor web de destino em nome do cliente e aguarda a resposta do servidor.
   5. Quando o servidor web responde, o proxy server pode armazenar a resposta em cache para uso futuro e, em seguida, entrega-la ao cliente.

7. Qual é a relação entre o protocolo MIME e o serviço WEB?

   R: O protocolo MIME (Multipurpose Internet Mail Extensions) permite que o serviço web transmita e processe diferentes tipos de conteúdo além do texto

simples. Inicialmente usado para transferência de anexos binários em e-mails, ele foi estendido para o serviço web. Ao enviar uma resposta HTTP, o cabeçalho "Content-Type" é definido com base no protocolo MIME, permitindo ao cliente processar corretamente a resposta. Essa relação possibilita que o serviço web ofereça uma experiência de usuário mais rica, suportando conteúdos como imagens, áudio, vídeo, CSS e JavaScript, além de texto. O protocolo MIME é fundamental para a versatilidade dos sites modernos.

8. Faça uma conexão telnet no servidor WEB e utilize os comandos do protocolo (GET, HEAD, etc.) para visualizar páginas, cabeçalhos, etc. Anotar os resultados dessa experiência.
9. Crie e publique uma página HTML principal contendo: (i) dados gerais sobre o grupo, (ii) informações do grupo (faixa de IP utilizada, a descrição do nome de domínio usado, nome/alias/IP das máquinas servidoras e faixa de IPs dinâmicos para as estações de trabalho e outras informações relevantes); (iii) Respostas das questões dos roteiros e (iv) Link para os domínios virtuais de cada aluno (provavelmente mapeado no public_html do aluno), onde deve estar o relatório individual do aluno descrito em formato HTML.

5. Conclusão

5.1 - Resultado do projeto

Eu, Ana Carolina, pude aprender muito mais sobre o DNS e SMTP, entendo o que cada serviço faz e como configurar cada um. Obtive mais conhecimento teórico do que prático, pois em todos os laboratórios eu não obtive sucesso.

Fiquei focada na parte do laboratório DNS, tentando fazer com que o ping entre qualquer domínio funcionasse, mas não deu certo. Fiz todos os passos até que ao chegar na parte de enviar email eu percebi que o domínio não funcionava. Testei novamente através de vídeos do youtube e fóruns da internet, sobre outras formas de configurar o dns, mas nenhuma delas deu certo.

O ponto de maior dificuldade foi poder fazer a configuração do dns, que mesmo fazendo muitas pesquisas ele não funcionou. Ainda assim achei interessante e somente através das pesquisas pude entender como as configurações do DNS e SMTP funcionavam. Considero minha participação muito ativa na busca por tentar configurar o dns e fazer o envio de email por smtp, me dando nota 7 pela experiência de poder desbravar os vários comandos e entender o funcionamento dos mesmos.

[ALTERAR

Eu, Ricardo, fui capaz de integrar os conhecimentos teóricos da aula com a prática desenvolvida no laboratório. Compreendi profundamente o conceito do protocolo SSL/TLS, bem como a importância dos certificados digitais.

Inicialmente, encontrei algumas dificuldades para gerar os certificados e associá-los ao servidor do Apache2. Contudo, esse desafio ajudou-me a entender melhor como funciona um sistema de criptografia e segurança.

Apesar de ter encontrado diferenças no sistema Mac, consegui efetivamente realizar o laboratório no ambiente Debian Ubuntu. A interpretação de cada comando do OpenSSL demandou esforço, e a configuração do servidor Apache2 apresentou alguns problemas. Essas dificuldades eram esperadas, dado que eu nunca havia trabalhado com essas tecnologias antes. Sendo necessário pesquisar para entender como utilizar.

No entanto, ao final, consegui gerar meu próprio certificado para responder às solicitações HTTPS e criar uma página com o certificado digital. Portanto, considero minha participação bastante ativa e, avaliando meu desempenho, acredito merecer

uma nota 9.

Portanto, podemos dizer que conseguimos entender muito mais sobre o fato de como funciona a criptografia e como ela pode auxiliar no processo de segurança na própria internet. Tivemos dificuldade em pode executar as chaves de criptografia e configurar o apache, mas conseguimos reter conhecimento teórico sobre esse laborátorio.