GUIA DE ESTUDO PARA

CERTIFICAÇÃO LINUX LPIC-1
EXAME 202 Versão 4.0

Liberado sob a GFDL por
Emerson Henrique Kfuri Pereira <emersonkfuri@gmail.com>

Revisão 20151008

Introdução

Sobre o LPI

O Linux Professional Institute foi formalmente constituído como uma organização sem fins lucrativos, em New Brunswick, Canadá, em 25 de Outubro de 1999. Possui escritórios operacionais em Toronto, Canadá e Sacramento, EUA - com filiais em todo o mundo. O LPI reúne uma comunidade ativa e empenhada de empresas, profissionais de TI, organizações de treinamento e voluntários para alcançar os programas do LPI. Ele é reconhecido mundialmente como a principal organização incentivando e ajudando na utilização profissional de Linux, Open Source e Software Livre.
As certificações padrão de indústria do LPI são entregues em milhares de locais em todo o mundo, em vários idiomas e com o apoio dos empregadores, fornecedores e formadores.

- Missão
    - Fornecer um framework global, liderança de indústria e serviços para melhorar, desenvolver e promover novas carreiras profissionais em tecnologias Linux e Open Source.
- Visão
    - Ser reconhecido como a organização nº 1 que fornece padrões de liderança global, direção e habilidade para aqueles que buscam uma carreira em Linux e tecnologia Open Source.

Endereço da página da organização: www.lpi.org

Sobre a Certificação LPIC-2

LPIC-2: Certificação Profissional de Rede Linux

LPIC-2 (Linux Professional Institute Certification 2) é a segunda certificação no programa de certificação profissional multinível do LPI. O LPIC-2 irá validar sua habilidade em administrar redes mistas de tamanho pequeno a médio. Você deve ter uma certificação LPIC-1 ativa para receber a certificação LPIC-2, mas os exames da LPIC-1 e LPIC-2 podem ser realizadas em qualquer ordem.

Para passar na LPIC-2, você deve ser capaz de:

- administrar um sítio de tamanho pequeno a médio
- planejar, implementar, manter, manter consistência, proteger, e solucionar problemas de uma pequena rede mista (MS, Linux) incluindo um:
    - servidor LAN (Samba, NFS, DNS, DHCP, gerenciamento de cliente)
    - Gateway Internet (firewall, VPN, SSH, cache/proxy web, correio)
    - Servidor Internet (servidor web e proxy reverso, servidor FTP)
- supervisionar assistentes
- aconselhar gerenciamento em automação e compras

Os exames 201 e 202 são requeridos exames para a certificação LPI Nível 2. Eles cobrem as habilidades avançadas para o profissional Linux que são comuns através de todas distribuições Linux.

Sobre o Exame 202

- Versão dos Objetivos do Exame
    - Versão 4.0 (última grande atualização: 1 de Novembro de 2013, última pequena atualização de formatação: 4 de Dezembro de 2014)
- Exame Coberto
    - LPIC-2 (LPI-202); Exame 2 de 2 para obter a certificação LPIC-2 Profissional de Rede Linux
- Objetivos Refletidos no Exame Publicado
    - 21 de Novembro de 2013
- Pré-requisitos Requeridos
    - Passar com sucesso nos exames LPIC-1 101 e 102, bem como o LPI 201. Os dois exames LPIC-2 podem ser completados em qualquer ordem, mas os candidatos precisam passar em ambos LPI-201 e LPI-202 em ordem ordem para obter a certificação LPIC-2.
- Sobre os Pesos dos Objetivos
    - Cada objetivo é atribuído um valor de peso. Os pesos variam aproximadamente de 1 a 10 e indicam a importância relativa de cada objetivo. Objetivos com pesos mais elevados serão cobertos no exame com mais perguntas.

Sobre este material

- Licença
    - Este material está disponibilizado sob a licença GNU FDL. É permitido copiar, distribuir e/ou modificar esse documento.
- Conteúdo
    - Este material contém informações obtidas na Internet e páginas de manuais (man page), citadas nas referências de cada texto.
    - Seu conteúdo é voltado principalmente para as áreas de conhecimento chave cobertas pelos objetivos do exame.
- Objetivos

- O objetivo primário desse material é prover explicações, referências, exemplos e exercícios para a Certificação Profissional de Rede Linux (LPIC-2), exame 201.
- Ele deve ser usado como guia e não como fonte única dos estudos para o exame em questão.

- Pré-requisitos

  - Esse material assume que seus usuários já possuem:
    - experiência como administrador de um servidor Linux

- Exercícios práticos

  - Para os exercícios práticos, é assumida a instalação de uma máquina virtual usando a mídia de instalação “mínima” da distribuição CentOS 6 x86_64.
  - Os requisitos de hardware mínimos da máquina virtual são:
    - 2 processadores
    - 2 GB RAM
    - 1 VHD de 20 GB
    - 4 VHDs de 10 GB
    - 3 NICs
  - Os discos virtuais podem ser dinâmicos.
  - A máquina virtual deve estar em modo de virtualização completa (Full Virtualization) para que os exercícios práticos funcionem corretamente.
  - A interface principal de rede da VM deve estar em modo bridge com o virtualizador
  - As interfaces 2 e 3 de rede devem ser configuradas em uma mesma rede privada
  - Algumas áreas de conhecimento chave de alguns objetivos não possuem exercícios práticos devido ao uso da distribuição CentOS 6.
  - Os exercícios práticos não tem respostas definidas, porém, possuem exemplos que se aplicam.
  - Os exercícios práticos de alguns objetivos podem depender de exercícios de objetivos anteriores.
  - Para obter ajuda nos parâmetros dos comandos, instalar o pacote man (não vem instalado por padrão na instalação mínima).

- Simulados

  - “…” (reticências) no enunciado de questões dos simulados significam que as frases devem ser completadas.
  - O número de questões dos simulados é 5 vezes o peso dos objetivos.

- Notações

  - Em comandos, os símbolos < > indicam um valor de um argumento não opcional.
  - Em comandos, os símbolos [ ] indicam um valor de um argumento opcional.
  - Nos textos, os símbolos ( ) indicam referências à páginas de manuais.
  - Nos textos, os símbolos [ ] indicam referências à fontes externas.

Histórico

Revisão 20150901 - Emerson Pereira <emersonkfuri@gmail.com>
- Versão inicial contendo o objetivo 207.1

Revisão 20150902 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição dos objetivos 207.2 e 207.3

Revisão 20150904 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do objetivo 208.1

Revisão 20150910 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do objetivo 208.2

Revisão 20150911 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição dos objetivos 208.3 e 208.4

Revisão 20150915 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do objetivo 209.1

Revisão 20150916 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do objetivo 209.2

Revisão 20150918 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição dos objetivos 210.1, 210.2 e 210.3

Revisão 20150921 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do objetivo 210.4

Revisão 20150924 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do tópico 211

Revisão 20151007 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Adição do tópico 212

Revisão 20151008 - Emerson Pereira <emersonkfuri@gmail.com>
- Complementação de conteúdo
- Exercícios práticos para os objetivos 210.3 e 210.4

Sumário

Tópicos

Tópico 207: Servidor de Nome de Domínio

207.1 Configuração básica de um servidor DNS

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de configurar o BIND para funcionar como um servidor de DNS caching-only. Esse objetivo inclui a habilidade de gerenciar um servidor em execução e configurar log.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários do BIND 9.x
- Definindo a localização dos arquivos de zona do BIND no arquivos de configuração
- Recarregando arquivos de configuração e zonas modificadas
- Consciência do dnsmasq, djbdns e PowerDNS como alternativas de servidores de nome

Termos e utilitários:

- /etc/named.conf
- /var/named
- /usr/sbin/rndc
- kill
- host
- dig

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários do BIND 9.x

- DNS [1]
    - O Domain Name System (DNS) é um sistema de nomenclatura hierárquica distribuída para computadores, serviços ou qualquer recurso conectado à Internet ou de uma rede privada. Ele associa várias informações com nomes de domínio atribuídos a cada uma das entidades participantes. Mais importante ainda, ele traduz nomes de domínio, o que pode ser facilmente memorizados por seres humanos, para os endereços IP numéricos necessários, para o propósito de serviços de computador e dispositivos em todo o mundo. O Domain Name System é um componente essencial da funcionalidade da maioria dos serviços de Internet, porque é serviço de diretório principal do Internet.
    - O DNS distribui a responsabilidade de atribuir nomes de domínio e mapear esses nomes para endereços IP com a designação de servidores autorizados para cada domínio. Servidores de nomes oficiais (servidores autoritativos) são atribuídos para serem responsáveis pelos seus domínios suportados, e podem delegar autoridade sobre sub-domínios para outros servidores de nomes. Esse mecanismo fornece um serviço distribuído e tolerante a falhas e foi concebido para evitar a necessidade para uma única base de dados central.
    - O DNS também especifica a funcionalidade técnica do serviço de banco de dados que está em seu núcleo. Ele define o protocolo DNS, uma especificação detalhada das estruturas de dados e trocas de comunicação de dados utilizados no DNS, como parte do Internet

Protocol Suite. Historicamente, outros serviços de diretório anteriores ao DNS não foram escaláveis para diretórios grandes ou globais como foram originalmente baseados em arquivos de texto, de forma destacada o resolvedor HOSTS.TXT. O DNS tem sido largamente utilizado desde os anos 1980.

- A Internet mantém dois espaços de nomes principais, a hierarquia de nome de domínio e os espaços de endereços do Internet Protocol (IP). O Domain Name System mantém a hierarquia de nome de domínio e fornece serviços de tradução entre ele e os espaços de endereços. Servidores de nome de Internet e um protocolo de comunicação implementam o Domain Name System. Um servidor de nomes DNS é um servidor que armazena os registros de DNS para um nome de domínio; um servidor de nomes DNS responde com respostas a consultas em seu banco de dados.
- Os tipos mais comuns de registros armazenados no banco de dados DNS são aqueles que lidam com autoridade de uma zona de DNS (SOA), os endereços IP (A e AAAA), trocadores de correio SMTP (MX), servidores de nomes (NS), os ponteiros para consultas de DNS reverso (PTR), e apelidos de nomes de domínio (CNAME). Apesar de não ser a intenção de ser um banco de dados de uso geral, o DNS pode armazenar registros para outros tipos de dados tanto para pesquisas automáticas de máquinas, para coisas como registros DNSSEC, ou para consultas humanas como registros de pessoa responsável (PR). Como um banco de dados de uso geral, o DNS também tem sido visto no uso de combate ao e-mail não solicitado (spam) usando uma lista blackhole em tempo real armazenada em um banco de dados DNS. Quer se trate de nomes da Internet ou em usos de propósito geral, o banco de dados DNS é tradicionalmente armazenado em um arquivo de zona estruturado.
- Operação
  - Mecanismo de resolução de endereço
    - Resolvedores de nomes de domínios determinam os servidores de nome de domínio responsáveis pelo nome de domínio em questão por uma sequência de consultas começando com o rótulo de domínio mais à direita (de nível superior).
    - O processo implica:
      - Um host de rede é configurado com um cache inicial (chamado hints) dos endereços conhecidos dos servidores de nomes de raiz. Tal arquivo de dica é atualizado periodicamente por um administrador, de uma fonte confiável.
      - Uma consulta a um dos servidores raiz para encontrar o servidor autoritário para o domínio de nível superior.
      - A consulta para o servidor TLD obtido para o endereço de um servidor DNS autoritário para o domínio de segundo nível.
      - A repetição da etapa anterior para processar cada etiqueta com o nome de domínio em seqüência, até a etapa final que retorna o endereço IP do host procurado.
    - O mecanismo dessa forma simples colocaria uma grande carga de operação sobre os servidores raiz, com todas as pesquisas para um endereço inicial, consultando um deles. Sendo tão crítico como eles são para o funcionamento global do sistema, tal uso pesado criaria um gargalo intransponível para trilhões de consultas feitas a cada dia. Na prática, o cache é usado em servidores de DNS para ultrapassar esse problema, e, como resultado, os servidores de nomes de raiz, na verdade, estão envolvidos com muito pouco do tráfego total.

        - Servidor de nome recursivo e armazenamento em cache
            - Em teoria, servidores autorizados são suficientes para o funcionamento da Internet. No entanto, com apenas servidores de nomes autoritários operando, todas as consultas DNS devem começar com consultas recursivas na zona de raiz do Domain Name System e cada sistema de usuário teria que implementar software capaz de resolver operação recursiva.
            - Para melhorar a eficiência, reduzir o tráfego de DNS através da Internet, e aumentar o desempenho em aplicativos de usuário final, o Domain Name System suporta servidores de cache de DNS que armazenam os resultados da consulta DNS por um período de tempo determinado na sua configuração (time-to-live) do registro de nome de domínio em questão. Normalmente, esses servidores de cache DNS, também chamados de caches de DNS, também implementam o algoritmo recursivo necessário para resolver um determinado nome começando com a raiz do DNS até os servidores de nomes oficiais do domínio consultado. Com essa função implementada no servidor de nomes, os aplicativos de usuário ganham eficiência na concepção e funcionamento.
            - Como um exemplo, se um cliente quer saber o endereço para "www.example.com", ele vai enviar, para um servidor de nome de caching recursivo, um pedido DNS afirmando “Eu gostaria do endereço IPv4 para 'www.example.com”. O servidor de nomes recursivo, então, consulta servidores de nome de autoridade até que ele receba uma resposta a essa consulta (ou retorna um erro se não for possível obter uma resposta) - nesse caso 93.184.216.34.
            - A combinação de cache DNS e funções recursivas em um servidor de nomes não é obrigatório; as funções podem ser implementadas de forma independente em servidores para fins especiais.
            - Prestadores de serviços de Internet normalmente fornecem servidores de nomes recursivo e cache para seus clientes. Além disso, muitos roteadores de rede doméstica de implementam caches DNS e recursores para melhorar a eficiência na rede local.
- BIND [2]
    - O BIND ou named, é o software Domain Name System (DNS) mais utilizado na Internet. Em sistemas operacionais Unix-like é o padrão de fato.
    - O software foi originalmente concebido na Universidade da Califórnia em Berkeley (UCB), no início de 1980. O nome origina como um acrônimo de Berkeley Internet Name Domain, refletindo o uso do aplicativo dentro da UCB. O software é composto, mais proeminente, do componente de servidor DNS, chamado named, contraído de “name daemon”. Além disso, a suíte contém várias ferramentas de administração, e uma biblioteca de interface de resolvedor DNS. A última versão do BIND é o BIND 9, lançado pela primeira vez em 2000.
    - A partir de 2009, o Internet Software Consortium (ISC) desenvolveu um novo software, chamado inicialmente BIND10. Com o lançamento da versão 1.2.0 do projeto, foi rebatizado Bundy para encerrar o envolvimento do ISC no projeto.
    - História
        - Originalmente escrito por quatro estudantes de pós-graduação do Grupo de Pesquisa de Sistemas de Computação na Universidade da Califórnia, em Berkeley (UCB), o BIND foi lançado pela primeira vez com o Berkeley Software Distribution 4.3BSD. Paul Vixie começou a mantê-lo em 1988, enquanto trabalhava para a

Digital Equipment Corporation. A partir de 2012, a Internet Systems Consortium mantém, atualiza e escreve novas versões do BIND.

- O BIND foi escrito por Douglas Terry, Mark Painter, David Riggle e Songnian Zhou no início de 1980, na Universidade da Califórnia, em Berkeley, como resultado de uma subvenção DARPA. O BIND é acrônimo para Berkeley Internet Name Domain, a partir de um artigo técnico publicado em 1984.
- As versões do BIND até 4.8.3 foram mantidas pelo Grupo de Pesquisa de Sistemas de Computador (CSRG) na UC de Berkeley.
- Em meados da década de 1980, Paul Vixie do DEC assumiu o desenvolvimento do BIND, lançando as versões 4.9 e 4.9.1. Paul Vixie continuou a trabalhar no BIND após deixar a DEC. O BIND versão 4.9.2 foi patrocinado pela Vixie Enterprises. Vixie eventualmente fundou o ISC, que se tornou a entidade responsável pela BIND versões começando com 4.9.3.
- O BIND 8 foi lançado pela ISC em Maio de 1997.
- A versão 9 foi desenvolvida pela Nominum, Inc. sob um contrato de terceirização do ISC, e a primeira versão foi lançada 9 de Outubro de 2000. Ele foi escrito a partir do zero, em parte, para abordar as dificuldades arquitetônicas com auditoria dos código de base dos BIND anteriores, e também para suportar o DNSSEC (DNS Security Extensions). Outras características importantes do BIND 9 incluem: TSIG, nsupdate, IPv6, rndc (remote name daemon control), visões, suporte multiprocessador e uma arquitetura de portabilidade melhorada. O rndc usa um segredo compartilhado para fornecer criptografia para os terminais locais e remotos durante cada sessão. O desenvolvimento do BIND 9 ocorreu sob uma combinação de contratos comerciais e militares. A maior parte dos recursos do BIND 9 foram financiados por fornecedores UNIX que queriam garantir que o BIND ficasse competitivo com as ofertas de DNS da Microsoft; os recursos DNSSEC foram financiados pelos militares dos EUA, que consideraram tão importante a segurança do DNS. O BIND 9 foi lançado em Setembro de 2000.
- Em 2009, a ISC iniciou um esforço para desenvolver uma nova versão da suíte de software, chamado BIND10. Além do serviço de DNS, a suíte BIND10 também incluiu componentes de servidor DHCP IPv4 e IPv6. Em Abril de 2014, com o lançamento do BIND10 1.2.0 o ISC concluiu os seus trabalhos de desenvolvimento do projeto e renomeou o projeto para Bundy, movendo o repositório de código-fonte para o GitHub para um maior desenvolvimento de esforços públicos de fora. O Bundy é suportado por comunidade pelo o web site http://bundy-dns.de/. O ISC descontinuou o seu envolvimento no projeto, devido às medidas de corte de custos. O desenvolvimento dos componentes DHCP foram divididos no projeto Kea.

- Configuração
  - /etc/named.conf [3]
    - A configuração do BIND 9 é composta por declarações e comentários. Declarações terminam com um ponto e vírgula. Declarações e os comentários são os únicos elementos que podem aparecer sem colocar chaves. Muitas declarações contêm um bloco de sub-declarações, que também são terminadas com um ponto e vírgula.
    - As declarações a seguir são suportadas:
      - acl - define uma lista de correspondência de endereço IP nomeado, para controle de acesso e outros usos

- controls - declara canais de controle para serem utilizados pelo utilitário rndc
- include - inclui um arquivo
- key - especifica as informações de chaves para uso em autenticação e autorização usando TSIG
- logging - especifica o que o servidor registra, e para onde as mensagens de log são enviadas
- lwres - configura o named para atuar também como um daemon resolvedor de peso leve (lwresd).
- masters - define uma lista de mestres do named para inclusão nas cláusulas mestres de zona no stub e no escravo
- options - controla as opções de configuração global do servidor e define padrões para outras declarações
- server - define determinadas opções de configuração em uma base per-server
- statistics-channels - declara canais de comunicação para ter acesso as estatísticas do named
- trusted-keys - define chaves DNSSEC de confiança
- managed-keys - listas chaves DNSSEC para serem mantidas atualizadas usando manutenção de âncoras de confiança - RFC 5011
- view - define uma visão
- zone - define uma zona

- Utilitários
  - rndc(8) - utilitário de controle de servidor de nome
    - Exemplos:
      - rndc reload - recarrega arquivo de configuração e zonas
      - rndc reload <zona> [classe [visão]] - recarrega uma única zona
      - rndc refresh <zona> [classe [visão]] - agenda imediata manutenção de uma zona
      - rndc reconfig - recarrega arquivo de configuração e zonas novas apenas
      - rndc stats - escreve estatísticas do servidor para o arquivo de estatísticas
      - rndc stop - salva as atualizações pendentes para os arquivos mestres e para o servidor
      - rndc halt - para o servidor sem salvar as atualizações pendentes
      - rndc flush - descarrega todos os caches do servidor
      - rndc flush <visão> - descarrega os caches para uma visão
      - rndc status - exibe o estado do servidor
  - host(1) - utilitário de pesquisa DNS
    - Exemplos:
      - host <nome> - consulta o endereço IP associado ao nome
      - host <endereço IP> - consulta o nome associado ao endereço IP
      - host <nome> <servidor> - consulta o endereço IP associado ao nome usando o servidor especificado
      - host -t <tipo> <string> - consulta a string do tipo especificado no servidor
  - dig(1) - utilitário de pesquisa DNS

- Exemplos:
  - dig <nome> - consulta o endereço IP associado ao nome
  - dig <endereço IP> - consulta o nome associado ao endereço IP
  - dig @<servidor> <nome> - consulta o endereço IP associado ao nome usando o servidor especificado
  - dig <string> <tipo> - consulta a string do tipo especificado no servidor

Definindo a localização dos arquivos de zona do BIND no arquivos de configuração

- /etc/named.conf [3]
  - zone <nome da zona> [classe] {
    - type <tipo>;
    - file <arquivo>;
    - <outras opções>

Recarregando arquivos de configuração e zonas modificadas

- Através do script do serviço para SysVinit
  - Exemplo: /etc/init.d/named reload
- Através do sinal HUP para o processo
  - Exemplos:
    - kill -HUP <PID>
    - killall -HUP named
    - pkill -HUP named
- Através de subcomandos do rndc
  - Exemplos:
    - rndc reload
    - rndc reconfig
    - rndc refresh

Consciência do dnsmasq, djbdns e PowerDNS como alternativas de servidores de nome

- dnsmasq [4]
  - dnsmasq é um software livre, encaminhador de DNS e servidor DHCP para redes pequenas. É considerado ser facilmente configurado, com baixo uso de recursos de sistema.
  - Ele suporta Linux, BSDs, Android e OS X, e é incluído na maioria das distribuições Linux.
  - Funcionalidades
    - O servidor DHCP suporta concessões DHCP estáticas e dinâmicas, múltiplas redes e intervalos de endereços IPs. O servidor DHCP integra com o servidor DNS e aceita máquinas locais com endereços DHCP alocados aparecerem no DNS.
    - Ele faz cache de registros DNS, reduzindo a carga no fluxo de saída de servidores de nomes e melhorando a performance, e pode ser configurado para pegar automaticamente os endereços dos servidores de saída.
    - O dnsmasq suporta modernos padrões de Internet como IPv6 e DNSSEC, inicialização por rede com suporte para BOOTP, PXE e TFTP e também scripting LUA.
    - Alguns provedores de serviços de Internet reescrevem as respostas NXDOMAIN (domínio não existe) dos servidores de DNS. Isso força os navegadores web a procurar uma página sempre que um usuário tenta navegar para um domínio que

não existe. O dnsmasq pode filtrar esses registros NXDOMAIN falsos, previnindo esse comportamento potencialmente indesejado.

- djbdns [5]
  - O pacote de software djbdns é uma implementaçao DNS. Ele foi criado por Daniel J. Bernstein, em reposta a suas frustações com repetidas falhas de seguranças no amplamente utilizado software de DNS BIND. Como um desafio, Bernstein ofereceu um prêmio de U$ 1000 para a primeira pessoa a achar uma falha de segurança no djbdns, que foi premiado em Março de 2009 a Matthew Dempsky.
  - A partir de 2004, o componente tinydns do djbdns foi o segundo mais popular servidor DNS em termos de número de domínios para qual eram servidores autoritativos, e o terceiro mais popular em termos de número de hosts de DNS executando ele.
  - O djbdns nunca foi vulnerável a amplamente utilizada vulnerabilidade de envenenamento de cache reportado em Julho de 2008, mas descobriu-se que ele é vulnerável a um ataque relacionado.
  - O código fonte não foi gerenciado centralizadamente deste de seu lançamento em 2001, e foi lançado para domínio público em 2007. A partir de Março de 2009, existem uma série de bifurcações (“forks”), um dos quais é o dbndns (parte do Projeto Debian), e mais de uma dúzia de correções para modificar a versão lançada.
  - Desenho
    - No djbdns, diferentes funcionalidades e serviços são divididos em programas separados. Por exemplo, transferências de zonas, análise de arquivos de zona, caching e resolução recursiva são implementados como programas separados. O resultado dessas decisões de desenho é a redução do tamanho do código e complexidade do programa daemon que provê a função principal de responder requisições de pesquisa. Bernstein afirma que isso é o verdadeiro espírito do sistema operacional Unix, e faz a verificação de segurança muito simples
- PowerDNS [6]
  - O PowerDNS é um servidor DNS, escrito em C++ e licenciado sob o GPL. Ele é executado na maioria dos derivados do Unix. O PowerDNS apresenta um grande quantidade de diferentes backends, de simples arquivos de zonas estilo BIND à bancos de dados relacionais e algorítimos de balanceamento de carga e alta disponibilidade/failover. Um recursor DNS é provido como um programa separado.
  - O PowerDNS é um produto da empresa holandesa PowerDNS.com BV, com inúmeras contribuições da comunidade opensource. O principal autor é Bert Hubert.
  - Funcionalidades
    - O servidor autoritativo PowerDNS (pdns_server) consiste de um único núcleo, e diversos backends carregáveis dinamicamente que executam em multi-thread. O núcleo lida com todo o processamento de pacotes e inteligência de DNS, enquanto um ou mais backends entregam registros de DNS, usando métodos de armazenamento arbitrários.
    - As transferências de zona e notificações de atualização são suportadas, e os processos podem ser executados sem privilégios e enjaulados. Vários caches são mantidos para acelerar o processamento de consultas. Controle de tempo de execução está disponível através do comando pdns_control, que permite a restauração das zonas separadas, expurgos de cache, notificações de zona e despejo de estatísticas em formato Multi Router Traffic Grapher/rrdtool. Informações em tempo real também podem ser obtidas através do servidor web embutido opcional.

- Há muitos projetos independentes para criar interfaces de gerenciamento para o PowerDNS, incluindo PowerAdmin, PDNSOps, PowerDNS em Rails e JPower Admin, e até mesmo um módulo Drupal chamado PowerAdmin.

Termos e utilitários

- /etc/named.conf - (5) arquivo de configuração para o named
- /var/named/ - diretório de dados do BIND
- /usr/sbin/rndc - (8) utilitário de controle de servidor de nome
- kill - (1) termina um processo
- host - (1) utilitário de pesquisa DNS
- dig - (1) utilitário de pesquisa DNS

Referências

1. http://en.wikipedia.org/wiki/Domain_Name_System
2. http://en.wikipedia.org/wiki/BIND
3. https://kb.isc.org/article/AA-01031
4. http://en.wikipedia.org/wiki/Dnsmasq
5. http://en.wikipedia.org/wiki/Djbdns
6. http://en.wikipedia.org/wiki/PowerDNS

Exercícios práticos

1. Preparação para exercícios
   a. Instalar os pacotes bind e bind-utils (Ex.: yum install <pacote>)

   b. Parar o serviço iptables (Ex.: service <serviço> stop)

2. Arquivos de configuração, termos e utilitários do BIND 9.x
   a. Listar os arquivos do pacote bind (Ex.: rpm <opções>)

   b. Listar os arquivos do pacote bind-utils (Ex.: rpm <opções>)

   c. Exibir o conteúdo dos arquivos /etc/named.* (Ex.: cat <arquivo>)

3. Software BIND com configuração padrão
   a. Iniciar o serviço named (Ex.: service <serviço> start)

   b. Através do utilitário rndc, verificar o estado do servidor named (Ex.: rndc <opções>)

   c. Através do utilitário service, verificar o estado do servidor named (Ex.: service <serviço> status)

   d. Consultar o endereço www.example.com, usando o servidor de nomes local, através do utilitário dig (Ex.: dig <opções>)

e. Consultar o endereço www.example.com, usando o servidor de nomes local, através do utilitário host (Ex.: host <opções>)

f. Parar o serviço named (Ex.: service <serviço> stop)

4. Software BIND como caching only, sem configuração explícita
   a. Apagar todos os arquivos /etc/named* e o arquivo /etc/rndc.key (Ex.: rm <opções> <objeto>)

   b. Apagar o diretório /var/named (Ex.: rm <opções> <objeto>)

   c. Criar o arquivo vazio /etc/named.conf (Ex.: touch <arquivo>)

   d. Iniciar o serviço named (Ex.: service <serviço> start)

   e. Através do utilitário service, verificar o estado do servidor named (Ex.: service <serviço> status)

   f. Consultar o endereço www.example.com, usando o servidor de nomes local, através do utilitário dig (Ex.: dig <opções>)

   g. Consultar o endereço www.example.com, usando o servidor de nomes local, através do utilitário host (Ex.: host <opções>)

   h. Acompanhe o log geral do sistema de forma de forma contínua (Ex.: tail <opções> <arquivo>)

   i. A partir de um host externo da mesma rede, consultar o endereço www.google.com, usando o serviço DNS instalado, através do utilitário host (Ex.: host <opções>)

   j. Se for possível, a partir de um host externo, consultar o endereço www.example.com, usando o serviço DNS instalado (Ex.: host <endereço> <servidor>)

   k. Parar de acompanhar o log geral do sistema

   l. Através do utilitário rndc, parar o serviço named (Ex.: rndc <opções>)

5. Software BIND como caching only, com configuração manual
   a. Alterar os atributos do arquivo /etc/named.conf
      i. Usuário Dono: root (chown <usuário> <objeto>)

      ii. Grupo Dono: named (chgrp <grupo> <objeto>)

      iii. Permissão: 640 (chmod <permissão> <objeto>)

      iv. Contexto SELinux: system_u:object_r:named_conf_t:s0 (chcon <contexto> <objeto>)

   b. Alterar os atributos do diretório /var/named

i. Usuário Dono: root (chown <usuário> <objeto>)

ii. Grupo Dono: named (chgrp <grupo> <objeto>)

iii. Permissão: 750 (chmod <permissão> <objeto>)

iv. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

c. Criar os diretórios /var/named/{data,slaves} (Ex.: mkdir <diretório>)

d. Alterar os atributos dos diretórios /var/named/{data,slaves}
   i. Usuário Dono: named (chown <usuário> <objeto>)

   ii. Grupo Dono: named (chgrp <grupo> <objeto>)

   iii. Permissão: 770 (chmod <permissão> <objeto>)

   iv. Contexto SELinux: system_u:object_r:named_cache_t:s0 (chcon <contexto> <objeto>)

e. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
   i. Opções
      1. Diretório: /var/named
      2. Aceitar Consultas (allow-query) : <rede do endereço principal/máscara do endereço principal>
   ii. Zona
      1. Nome: “.”
      2. Classe: IN
      3. Tipo: hint
      4. Arquivo: named.ca

f. Copiar o arquivo /usr/share/doc/bind-<versão>/sample/var/named/named.ca para /var/named (Ex.: cp <origem> <destino>)

g. Alterar os atributos do arquivo /var/named/named.ca
   i. Usuário Dono: root (chown <usuário> <objeto>)

   ii. Grupo Dono: named (chgrp <grupo> <objeto>)

   iii. Permissão: 640 (chmod <permissão> <objeto>)

   iv. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

h. Exibir o conteúdo do arquivo /var/named/named.ca (Ex.: cat <arquivo>)

i. Remover o arquivo /etc/rndc.key (Ex.: rm <objeto>)

j. Através do comando rndc-confgen, configurar o canal de comunicação entre o named e o rndc (Ex.: rndc-confgen)

k. Iniciar o serviço named (Ex.: service <serviço> start)

l. Através do utilitário rndc, verificar o estado do servidor named (Ex.: rndc <opções>)

m. Consultar o endereço www.example.com, usando o servidor de nomes local, através do utilitário host (Ex.: host <opções>)

n. A partir de uma máquina externa, consultar o endereço www.example.com, usando o serviço DNS instalado (Ex.: host <endereço> <servidor>)

o. Iniciar o serviço iptables (Ex.: service <serviço> start)

Simulado

1. São funcionalidades do software BIND 9x:
   a. Suporte a multiprocessadores
   b. Suporte a IPv6
   c. Suporte a visões
   d. Suporte a atualização automática de registros

2. O software BIND 9x vem compilado por padrão com suporte a cache, não sendo necessária nenhuma intervenção para o mesmo poder ser usado por máquinas de outras redes como servidor de cache.
   a. V
   b. F

3. Um servidor DNS do tipo cache recursivo é aquele que faz consultas recursivas aos servidores raiz e armazena a informação recebida em um cache local para não ser necessário a repetição da consulta de forma constante.
   a. V
   b. F

4. É possível executar o software BIND 9.x com o arquivo de configuração completamente vazio.
   a. V
   b. F

5. As declarações …, …, …, …, … e … do arquivo de configuração do software BIND 9.x, se referem respectivamente a opções gerais, zonas, visões, canais de controle, registros de operações e listas de acessos.

6. Para se consultar quais são os servidores raiz do serviço de DNS através do comando host, os argumentos … são necessários.

7. Através do comando dig, para pesquisar os servidores MX do domínio example.com no servidor 8.8.8.8, os argumentos … são necessários.

8. Para definir o diretório de dados do software BIND 9.x, a opção … deve ser usada na declaração … de seu arquivo de configuração padrão ….

9. Para declarar a zona ".", da classe Internet, do tipo "hint", usando o arquivo "named.ca", dentro da visão "internal", as seguintes configurações dentro do arquivo /etc/named.conf são necessárias.
   ...

10. Para recarregar o serviço named, os seguintes comandos podem ser usados
    a. top
    b. kill
    c. rndc
    d. service

11. Através do comando rndc, para se recarregar a zona example.com, os argumentos … são necessários.

12. O argumento … do comando rndc, limpa todos os caches do servidor DNS, sem gerar interrupções no serviço.

13. O software dnsmasq implementa serviço DHCP.
    a. V
    b. F

14. O nome do componente servidor de nomes do software djbdns é tinydns.
    a. V
    b. F

15. O software PowerDNS não permite uso de banco de dados relacionais como backend.
    a. V
    b. F

207.2 Criar e manter zonas de DNS

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de criar um arquivo de zona para uma zona direta ou reversa e hints para servidores de nível de raiz. Esse objetivo inclui definindo valores apropriados para registros, adicionando hosts em zonas e adicionando zonas ao DNS. Os candidatos devem ser capazes de delegar zonas a outro servidor de DNS.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários do BIND 9
- Utilitários para requerer informação de um servidor DNS
- Esquema, conteúdo e localização de arquivos de zonas do BIND
- Vários métodos para adicionar um novo host nos arquivos de zona, incluindo zonas reversas

Termos e utilitários:

- /var/named/
- sintaxe do arquivo de zona
- formato de registro de recursos
- dig
- nslookup
- host

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários do BIND 9

- Configuração
  - /etc/named.conf [1]
- Utilitários
  - rndc(8) - utilitário de controle de servidor de nome
  - dig(1) - utilitário de pesquisa DNS
  - nslookup(1) - consulta servidores de nome de Internet interativamente
    - Exemplos:
      - nslookup - consulta servidores de nome de Internet interativamente
      - nslookup <nome> - consulta o nome no DNS
      - nslookup <nome> <servidor> - consulta o nome no servidor DNS especificado
  - host(1) - utilitário de pesquisa DNS

Utilitários para requerer informação de um servidor DNS

- dig
- nslookup
- host

Esquema, conteúdo e localização de arquivos de zonas do BIND

- Arquivos de Zonas [2]
  - Um arquivo de zona Domain Name System (DNS) é um arquivo de texto que descreve uma zona DNS. Uma zona DNS é um subconjunto, muitas vezes um único domínio, da estrutura hierárquica de nome de domínio DNS. O arquivo de zona contém mapeamentos entre

nomes de domínio e endereços IP e outros recursos, organizados sob a forma de representações de texto de registros de recursos (RR). Um arquivo de zona pode ser um arquivo mestre DNS, descrevendo autoritariamente uma zona, ou pode ser usado para listar o conteúdo de um cache de DNS.

- Formato de arquivo
  - O formato de um arquivo de zona é definido na RFC 1035 (seção 5) e RFC 1034 (seção 3.6.1). Esse formato foi originalmente usado pelo pacote de software Berkeley Internet Name Domain (BIND), mas tem sido amplamente adotado por outros softwares de servidor DNS - embora alguns deles (por exemplo, NSD, PowerDNS) estão usando a zona de arquivos somente como ponto de partida para compilá-los em formato de banco de dados.
  - Um arquivo de zona é uma seqüência de entradas para os registros de recursos. Cada linha é uma descrição de texto que define um registro de recurso único (RR). A descrição consiste em vários campos separados por brancos espaciais (espaços ou tabs). O primeiro campo é o nome de domínio, chamado o proprietário do registro, mas se deixado em branco, o padrão é o dono do registro anterior. O nome de domínio é seguido pelo campo de tempo de vida, a classe de registro, o tipo de registro, e um ou possivelmente vários campos de tipo de dados específicos.
  - O campo de time-to-live especifica o tempo após o qual um cliente de nome de domínio deve descartar o registro e realizar uma nova operação de resolução para obter novas informações. A classe de registro indica o namespace de informações do registro. O espaço de nomes mais vulgarmente utilizado é o da Internet, indicado pelo parâmetro IN, mas existem outros e estão em uso, por exemplo, CHAOS. O tipo de registro de recurso é uma curta abreviatura para o tipo de informação armazenada no registro e determina o número de parâmetros necessários. O tipo também fornece o nome de cada registro. Por exemplo, um registro de endereço, com a abreviatura A para IPv4 e AAAA para IPv6, mapeia o nome de domínio no primeiro campo para um endereço IP no quarto campo, e um registro de trocador mensagens (tipo MX) especifica o host de correio Simple Mail Transfer Protocol (SMTP) para um domínio.
  - Os registros de recursos podem ocorrer em qualquer ordem em um arquivo de zona. Para a formatação de conveniência, os registros de recursos podem abranger várias linhas colocando entre parênteses um conjunto de parâmetros que se estende por várias linhas, mas pertence ao mesmo registro. O arquivo pode conter texto de comentário procedendo a esse texto com um ponto e vírgula, ou no início de uma linha, ou após o último campo em qualquer linha, ou em uma linha em branco. Comentários terminam no final de uma linha. O arquivo de zona pode conter qualquer número de linhas em branco com ou sem comentários.
  - O arquivo de zona também podem conter várias diretivas que são marcadas com uma palavra-chave começando com o cifrão. O mais notável é a palavra-chave $ORIGIN, que especifica o ponto de partida para a zona na hierarquia DNS. Se esta palavra-chave é omitida a partir de um arquivo de zona, a origem é inferida pelo software do servidor a partir da referência ao arquivo de zona na sua configuração do servidor.
  - Um exemplo de um arquivo de zona para o domínio example.com é o seguinte:
    - $ORIGIN example.com. ; designa o início desse arquivo de zona no espaço de nomes

- $TTL 1h ; tempo de expiração padrão para todos os recursos sem seus próprios valores TTL
- example.com. IN SOA ns.example.com. username.example.com. (2007120710 1d 2h 4w 1h)
- example.com. IN NS ns ; ns.example.com é um servidor de nomes para example.com
- example.com. IN NS ns.somewhere.example.com ; ns.somewhere.example é um servidor de nomes de backup para example.com
- example.com. IN MX 10 mail.example.com. ; mail.example.com é o servidor de email para example.com
- @ IN MX 20 mail2.example.com. ; equivalente a linha acima, "@" representa a origem da zona
- @ IN MX 50 mail3 ; equivalente a linha acima, mas usando nome de host relativo
- example.com. IN A 192.0.2.1 ; endereço IPv4 para example.com
- IN AAAA 2001:db8:10::1 ; endereço IPv6 para example.com
- ns IN A 192.0.2.2 ; endereço IPv4 para ns.example.com
- IN AAAA 2001:db8:10::2 ; endereço IPv6 para ns.example.com
- www IN CNAME example.com. ; www.example.com é um apelido para example.com
- wwwtest IN CNAME www ; wwwtest.example.com é outro apelido para www.example.com
- mail IN A 192.0.2.3 ; endereço IPv4 para mail.example.com
- mail2 IN A 192.0.2.4 ; endereço IPv4 para mail2.example.com
- mail3 IN A 192.0.2.5 ; endereço IPv4 para mail3.example.com

    - No mínimo, o arquivo de zona deve especificar o registro Início de Autoridade (SOA) com o nome do servidor de nomes mestre autoritário para a zona e do endereço de e-mail de alguém responsável pela gestão do servidor de nomes. Os parâmetros do registro SOA também especificam uma lista de temporização e parâmetros de expiração (número de série, período de atualização de escravo, tempo de repetição de escravo, tempo de expiração de escravo, e o tempo máximo para armazenar em cache o registro). Alguns softwares de servidor DNS, como BIND, também requerem pelo menos um registro de servidor de nomes adicional. O endereço de e-mail no SOA RR tem o símbolo @ substituído por um ponto. No arquivo de zona, os nomes de host que não terminam em um ponto são em relação à origem da zona. Por exemplo, no exemplo acima, www refere-se a www.example.com, e example.com. é example.com, e não example.com.example.com. Nomes que terminam com um ponto são ditos serem nomes de domínio totalmente qualificados (FQDN).
    - Um arquivo de zona é referenciado pelo arquivo de configuração do software servidor de nomes como o bind, tipicamente por uma declaração como:
        - zone "example.com" { type master; file "/var/named/db.example.com"; };
- Zona de raiz e de nível superior domínios
    - Os arquivos de zona para zona de raiz DNS e para o conjunto de domínios de alto nível contêm registros de recursos apenas para os servidores de nome de domínio com autoridade para cada nome de domínio.
- Localhost

- Alguns softwares de servidor configura automaticamente os registros de recursos para domínios ou nomes de host especialmente reconhecidos, como localhost, mas um arquivo mestre de zona personalizado pode ser usado.
- Um exemplo para a configuração manual da zona de encaminhamento para localhost é o seguinte:
  - $ORIGIN localhost.
  - @ 1D IN SOA @ root 1999010100 3h 15m 1w 1d
  - @ 1D IN NS @
  - @ 1D IN A 127.0.0.1
  - @ 1D IN AAAA ::1
- A definição de zona inversa correspondente é:
  - ;; arquivo de zona reversa para 127.0.0.1 e ::1
  - $TTL 3W
  - @ 3W IN SOA localhost. root.localhost. 1999010100 3h 15m 1w 1d
  - @ 3W IN NS localhost.
  - 1 3W IN PTR localhost.
- Esse arquivo não especifica a origem, de modo que ele pode ser usado para IPv4 e IPv6 com esta configuração:
  - zone "0.0.127.in-addr.arpa" IN { type master; file "r.local"; };
  - zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN { type master; file "r.local"; };
- Arquivos mestre de zona similares podem ser criados pela resolução inversa do endereço de broadcast e o endereço nulo.
- Tais arquivos de zona impedem um servidor DNS se referir a outros servidores DNS, possivelmente externos.

Vários métodos para adicionar um novo host nos arquivos de zona, incluindo zonas reversas

- RR [3]
  - Essa lista de alguns tipos de registro DNS fornece uma visão geral dos tipos de registros de recursos (registros de banco de dados) armazenados em arquivos de zona do Domain Name System (DNS).
  - O DNS implementa um distribuído, hierárquico e redundante banco de dados para informações associadas a nomes de domínio da Internet e endereços. Nesses servidores de domínio, diferentes tipos de registro são utilizados para fins diferentes.

Tipo	Descrição
A	Endereço IPv4
AAAA	Endereço IPv6
CNAME	Apelido (Canonical name)
MX	Trocador de email (Mail exchange)
NS	Servidor de nomes (Name server)
PTR	Ponteiro (Pointer)
SOA	Início de autoridade (Start of authority)

TXT	Texto (Text)

Termos e utilitários

- /var/named/ - diretório de dados do BIND
- sintaxe de arquivo de zona - [2]
- formatos de registro de recurso - [3]
- dig - (1) utilitário de pesquisa DNS
- nslookup - (1) consulta servidores de nome de Internet interativamente
- host - (1) utilitário de pesquisa DNS

Referências

1. https://kb.isc.org/article/AA-01031
2. http://en.wikipedia.org/wiki/Zone_file
3. http://en.wikipedia.org/wiki/List_of_DNS_record_types

Exercícios práticos

1. Preparação para exercícios
    a. Parar o serviço iptables (Ex.: service <serviço> stop)

    b. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
        i. Opções
            1. Escutar em (listen-on): 127.0.0.1 <endereço do IP principal> 10.200.<terceiro campo do IP principal>.<quarto campo do IP principal>
        ii. Visão
            1. Nome: “Interna”
            2. Classe: Internet
            3. Corresponder clientes (match-clients): 127.0.0.1 10.200.0.0/16
            4. Aceitar consulta (allow-query): 127.0.0.1 10.200.0.0/16
        iii. Visão
            1. Nome: “Externa”
            2. Classe: Internet
            3. Corresponder clientes (match-clients): <endereço da rede do IP principal>
            4. Aceitar consulta (allow-query): <endereço da rede do IP principal>
        iv. Zonas
            1. Mover a zona “.” para a visão Externa
        v. Opções
            1. Remover opção aceitar consultas

    c. Recarregar o daemon named através do comando rndc (Ex.: rndc <opções>)

    d. Preparar uma segunda instância do BIND

i. Instalar os pacotes bind, initscripts e vim-minimal, usando como raiz de instalação o diretório /srv/bind-chroot e a versão de lançamento igual ao do sistema (verificar no arquivo /etc/redhat-release) (Ex.: yum --installroot=<diretório> --releasever=<versão de lançamento> install <pacote>)

ii. Definir o endereço IP 10.200.30.<quarto campo do IP principal>, com a máscara 255.255.0.0, para a interface eth1:0 (Ex.: ifconfig <opções>)

2.

i. Desabilitar o SELinux temporariamente (setenforce 0)

ii. Alterar a raiz para o diretório /srv/bind-chroot (Ex.: chroot <diretório>)

iii. Criar o arquivo vazio /etc/fstab (Ex.: touch <arquivo>)

iv. Configurar o ponto de montagem /proc, com a origem proc, o sistema de arquivos proc, as opções padrões, sem dump e sem fsck (Ex.: vi <arquivo>)

v. Montar todos os sistemas de arquivo (Ex.: mount <opções>)

vi. Criar o dispositivo virtual random (MAKEDEV /dev/random)

vii. Remover todos os objetos do diretório /etc começados com named (Ex.: rm <opções> <objeto>)

viii. Remover o conteúdo do diretório /var/named/ (Ex.: rm <opções> <objeto>)

ix. Criar o arquivo vazio /etc/named.conf (Ex.: touch <arquivo>)

x. Alterar os atributos do arquivo /etc/named.conf
   1. Usuário Dono: root (chown <usuário> <objeto>)
   2. Grupo Dono: named (chgrp <grupo> <objeto>)
   3. Permissão: 640 (chmod <permissão> <objeto>)
   4. Contexto SELinux: system_u:object_r:named_conf_t:s0 (chcon <contexto> <objeto>)

xi. Alterar os atributos do diretório /var/named
   1. Usuário Dono: root (chown <usuário> <objeto>)
   2. Grupo Dono: named (chgrp <grupo> <objeto>)
   3. Permissão: 750 (chmod <permissão> <objeto>)
   4. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

xii. Criar os diretórios /var/named/{data,slaves} (Ex.: mkdir <diretório>)

xiii. Alterar os atributos dos diretórios /var/named/{data,slaves}
1. Usuário Dono: named (chown <usuário> <objeto>)

2. Grupo Dono: named (chgrp <grupo> <objeto>)

3. Permissão: 770 (chmod <permissão> <objeto>)

4. Contexto SELinux: system_u:object_r:named_cache_t:s0 (chcon <contexto> <objeto>)

xiv. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
1. Opções
a. Diretório: /var/named
b. Escutar em (listen-on): 10.200.30.<quarto campo do IP principal>

xv. Através do rndc-confgen, configurar o acesso do rndc ao named, alterando o endereço 127.0.0.1 para 10.200.30.<quarto campo do IP principal> (Ex.: rndc-confgen)

xvi. Iniciar o serviço named (Ex.: service <serviço> start)

xvii. Verificar o estado do named através do comando rndc (Ex.: rndc <opções>)

xviii. Encerrar a sessão atual do bash para voltar a raiz do sistema de arquivos principal (Ex.: exit)

3. Gerencia de zonas
a. Zona diretas
i. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
1. Visão “Interna”
a. Zona
i. Nome: “example.cluster”
ii. Classe: Internet
iii. Tipo: Mestre
iv. Arquivo: “data/example.cluster.zone”

ii. Criar o arquivo vazio referente a zona criada (Ex.: touch <arquivo>)

iii. Alterar os atributos do arquivo criado
1. Usuário Dono: named (chown <usuário> <objeto>)

2. Grupo Dono: named (chgrp <grupo> <objeto>)

3. Permissão: 640 (chmod <permissão> <objeto>)

4. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

iv. Preencher o arquivo de zona criado com as seguintes informações (Ex.: vi <arquivo>)
    1. Tempo de vida: 1D
    2. Origem: example.cluster
        a. RR
            i. Tipo: Início de autoridade
            ii. Servidor Mestre: ns.example.cluster
            iii. Email: root@example.cluster
            iv. Número de Série: <YYYYMMDD>01
            v. Período de Atualização de Escravo: 1H
            vi. Tempo de Repetição de Escravo: 15M
            vii. Tempo de Expiração de Escravo: 1W
            viii. Tempo de Cache: 1D
        b. RR
            i. Tipo: Servidor de Nomes
            ii. Endereço: ns.example.cluster
        c. RR
            i. Tipo: Trocador de correio
            ii. Prioridade: 10
            iii. Endereço: mail.example.cluster
        d. RR
            i. Nome: ns
            ii. Tipo: Endereço IPv4
            iii. Endereço: 10.200.<terceiro campo do IP principal>.1
        e. RR
            i. Nome: mail
            ii. Tipo: Endereço IPv4
            iii. Endereço: 10.200.<terceiro campo do IP principal>.2

v. Validar o arquivo de zona criado através do comando named-checkzone (Ex.: named-checkzone <opções>)

vi. Recarregar o daemon named através do comando rndc (Ex.: rndc <opções>)

vii. Usando o servidor de nomes local, resolver o início de autoridade para o domínio example.cluster (Ex.: host <opções>)

viii. Usando o servidor de nomes local, resolver os servidores de nome para o domínio example.cluster (Ex.: host <opções>)

ix. Usando o servidor de nomes local, resolver os servidores de correio para o domínio example.cluster (Ex.: host <opções>)

x. Usando o servidor de nomes local, resolver o endereço IPv4 do servidor ns.example.cluster (Ex.: host <opções>)

xi. Usando o servidor de nomes local, resolver o endereço IPv4 do servidor mail.example.cluster (Ex.: host <opções>)

xii. Alterar a raiz para o diretório /srv/bind-chroot (Ex.: chroot <diretório>)

xiii. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
1. Zona
    a. Nome: “subdomain.example.cluster”
    b. Classe: Internet
    c. Tipo: Mestre
    d. Arquivo: “data/subdomain.example.cluster.zone”
    e. Aceitar transferência (allow-transfer): 10.200.<terceiro campo do IP principal>.<quarto campo do IP principal>

xiv. Criar o arquivo vazio referente a zona criada (Ex.: touch <arquivo>)

xv. Alterar os atributos do arquivo criado
1. Usuário Dono: named (chown <usuário> <objeto>)
2. Grupo Dono: named (chgrp <grupo> <objeto>)
3. Permissão: 640 (chmod <permissão> <objeto>)
4. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

xvi. Preencher o arquivo de zona criado com as seguintes informações (Ex.: vi <arquivo>)
1. Tempo de vida: 1D
2. Origem: subdomain.example.cluster
    a. RR
        i. Tipo: Início de autoridade
        ii. Servidor Mestre: ns.subdomain.example.cluster
        iii. Email: root@subdomain.example.cluster
        iv. Número de Série: <YYYYMMDD>01
        v. Período de Atualização de Escravo: 1H
        vi. Tempo de Repetição de Escravo: 15M
        vii. Tempo de Expiração de Escravo: 1W
        viii. Tempo de Cache: 1D
    b. RR
        i. Tipo: Servidor de Nomes
        ii. Endereço: ns.subdomain.example.cluster
    c. RR
        i. Nome: ns
        ii. Tipo: Endereço IPv4
        iii. Endereço: 10.200.30.<quarto campo do IP principal>

xvii. Validar o arquivo de zona criado através do comando named-checkzone (Ex.: named-checkzone <opções>)

xviii. Reconfigurar o daemon named através do comando rndc (Ex.: rndc <opções>)

xix. Encerrar a sessão atual do bash para voltar a raiz do sistema de arquivos principal (Ex.: exit)

b. Zonas reversas
   i. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
      1. Visão “Interna”
         a. Zona
            i. Nome: “<terceiro campo do IP principal>.200.10.in-addr.arpa.”
            ii. Classe: Internet
            iii. Tipo: Mestre
            iv. Arquivo: “data/<terceiro campo do IP principal>.200.10.in-addr.arpa.zone”

   ii. Criar o arquivo vazio referente a zona criada (Ex.: touch <arquivo>)

   iii. Alterar os atributos do arquivo criado
      1. Usuário Dono: named (chown <usuário> <objeto>)

      2. Grupo Dono: named (chgrp <grupo> <objeto>)

      3. Permissão: 640 (chmod <permissão> <objeto>)

      4. Contexto SELinux: system_u:object_r:named_zone_t:s0 (chcon <contexto> <objeto>)

   iv. Preencher o arquivo de zona criado com as seguintes informações (Ex.: vi <arquivo>)
      1. Tempo de vida: 1D
      2. Origem: <terceiro campo do IP principal>.200.10.in-addr.arpa
         a. RR
            i. Tipo: Início de autoridade
            ii. Servidor Mestre: ns.example.cluster
            iii. Email: root@example.cluster
            iv. Número de Série: <YYYYMMDD>01
            v. Período de Atualização de Escravo: 1H
            vi. Tempo de Repetição de Escravo: 15M
            vii. Tempo de Expiração de Escravo: 1W
            viii. Tempo de Cache: 1D
         b. RR
            i. Tipo: Servidor de Nomes
            ii. Endereço: ns.example.cluster
         c. RR
            i. Nome: 1

ii. Tipo: Ponteiro
iii. Endereço: ns.example.cluster.

d. RR
i. Nome: 2
ii. Tipo: Ponteiro
iii. Endereço: mail.example.cluster.

v. Validar o arquivo de zona criado através do comando named-checkzone (Ex.: named-checkzone <opções>)

vi. Reconfigurar o daemon named através do comando rndc (Ex.: rndc <opções>)

vii. Usando o servidor de nomes local, resolver o início de autoridade para o domínio <terceiro campo do IP principal>.200.10.in-addr.arpa (Ex.: host <opções>)

viii. Usando o servidor de nomes local, resolver os servidores de nome para o domínio <terceiro campo do IP principal>.200.10.in-addr.arpa (Ex.: host <opções>)

ix. Usando o servidor de nomes local, resolver o nome para o endereço IPv4 10.200.<terceiro campo do IP principal>.1 (Ex.: hosts <opções>)

x. Usando o servidor de nomes local, resolver o nome para o endereço IPv4 10.200.<terceiro campo do IP principal>.1 (Ex.: hosts <opções>)

c. Delegar zonas
i. No arquivo de configuração da zona example.cluster, declarar (Ex.: vi <arquivo>)
1. RR
a. Nome: subdomain
b. Tipo: Servidor de Nomes
c. Endereço: ns.subdomain.example.cluster

ii. Validar o arquivo de zona criado através do comando named-checkzone (Ex.: named-checkzone <opções>)

iii. Recarregar o daemon named através do comando rndc (Ex.: rndc <opções>)

iv. Verificar o arquivo de log geral do sistema, referente a última recarga do serviço named (Ex.: tail <arquivo>)

v. Usando o servidor de nomes 10.200.30.<quarto campo do IP principal>, resolver o início de autoridade para o domínio subdomain.example.cluster (Ex.: host <opções>)

vi. Usando o servidor de nomes 10.200.30.<quarto campo do IP principal>, resolver os servidores de nome para o domínio subdomain.example.cluster (Ex.: host <opções>)

4. Iniciar o serviço iptables (Ex.: service <serviço> start)

Simulado

1. No arquivo de configuração do named, a declaração que se refere a domínios é ....

2. Quando se está criando um domínio no servidor de nomes local, sendo esse servidor o mestre do domínio, o tipo de zona declarado no arquivo de configuração é soa.
    a. V
    b. F

3. No arquivo de configuração do named, é possível interromper a funcionalidade de recursão do servidor, através da opção ….

4. No utilitário rndc, o argumento … é usado exclusivamente reenviar notificações referentes a uma zona.

5. Para consultar qual é o servidor mestre do domínio google.com, através do utilitário dig, qual comando deve ser usado?

6. Através do comando nslookup, é possível consultar todos os tipos de RR.
    a. V
    b. F

7. Através do comando host, é possível determinar se um subdomínio de um domínio é delegado a outro servidor.
    a. V
    b. F

8. São tipos de RR válidos:
    a. soa, ns, mx, aa
    b. a, aa, ptr, ns
    c. aaaa, soa, txt, ptr
    d. ns, aa, spf, txt

9. Pode se omitir o nome de um RR, porém o nome do último RR com definição de nome será usado.
    a. V
    b. F

10. … é o tipo de RR usado para apelidos.

11. São informações válidas para um RR do tipo início de autoridade:
    a. email
    b. servidor de nomes
    c. tempo de vida do registro
    d. definição de temporização dos registros

12. Todos os arquivos de zona devem necessariamente possuir no mínimo um servidor de nomes que tenha registro de endereço IP.
    a. V

b. F

13. Para se delegar um subdomínio ao um outro servidor, basta criar um RR do tipo NS, especificando o nome do subdomínio e o servidor de nomes responsável.
    a. V
    b. F

14. No RR do tipo MX, a prioridade maior de entrega é representada pelo maior número.
    a. V
    b. F

15. O nome do domínio reverso para a subrede 10.0.0.0/8 é ….

207.3 Protegendo um servidor DNS

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de configurar um servidor DNS para executar como um usuário não privilegiado e executar em uma jaula chroot. Esse objetivo inclui segurança em troca de dados entre servidores DNS.
Áreas de conhecimentos chave:

- Arquivos de configuração do BIND 9
- Configurando o BIND para executar em uma jaula chroot
- Dividir a configuração do BIND usando encaminhamentos
- Configurando e usando assinatura de transações (TSIG)
- Consciência do DNSSEC e ferramentas básicas

Termos e utilitários:

- /etc/named.conf
- /etc/passwd
- DNSSEC
- dnssec-keygen
- dnssec-signzone

Áreas de conhecimentos chave

Arquivos de configuração do BIND 9

- /etc/named.conf

Configurando o BIND para executar em uma jaula chroot

- chroot [1]
    - A chroot em sistemas operacionais Unix é uma operação que altera o diretório raiz aparente para o processo de execução atual e seus filhos. Um programa que é executado em um ambiente tão modificado não pode nomear (e, portanto, normalmente não acessar) arquivos fora da árvore de diretórios designada. O termo "chroot" pode se referir ao chroot(2) chamada de sistema ou o chroot(8) programa wrapper. O ambiente modificado é chamado de "jaula".

Dividir a configuração do BIND usando encaminhamentos

- Encaminhamentos [2]
    - Um encaminhador é um servidor Domain Name System (DNS) em uma rede usado para encaminhar consultas DNS de nomes DNS externos para servidores DNS fora dessa rede. Você também pode encaminhar consultas de acordo com os nomes de domínio específicos usando encaminhadores condicionais.
    - Um servidor DNS em uma rede é designado como um encaminhador por ter os outros servidores DNS na rede encaminhando as consultas que não conseguem resolver localmente para esse servidor DNS. Usando um encaminhador, pode-se gerenciar a resolução de nomes para nomes fora da rede, tais como nomes na Internet, e melhorar a eficiência da resolução de nomes para os computadores de rede local.

- Sem ter um servidor DNS específico designado como encaminhador, todos os servidores DNS podem enviar consultas fora de uma rede usando suas dicas de raiz. Como resultado, uma grande quantidade de informação DNS interna, e possivelmente crítica, pode ser exposta na Internet. Além dessa questão de segurança e privacidade, esse método de resolução pode resultar em um grande volume de tráfego externo que é caro e ineficiente para uma rede com uma conexão de Internet lenta ou uma empresa com alto custo do serviço de Internet.
- Quando se designa um servidor DNS como um encaminhador, o encaminhador responsável passa a controlar o tráfego externo, limitando assim a exposição do servidor DNS para a Internet. Um encaminhador irá acumular um grande cache de informações de DNS externo, porque todas as consultas DNS externas na rede são resolvidos por ele. Em um pequeno espaço de tempo, um encaminhador irá resolver uma boa parte das consultas DNS externos usando esses dados em cache e, assim, diminuir o tráfego de Internet através da rede e o tempo de resposta para os clientes DNS.
- Um servidor DNS configurado para usar um encaminhador irá se comportar de forma diferente de um servidor DNS que não está configurado para usar um encaminhador. Um servidor DNS configurado para usar um encaminhador se comporta como se segue:
  - Quando o servidor DNS recebe uma consulta, ele tenta resolver essa consulta usando as zonas primárias e secundárias que hospeda e seu cache.
  - Se a consulta não pode ser resolvida usando esses dados locais, então ele vai encaminhar a consulta para o servidor DNS designado como encaminhador.
  - O servidor DNS irá esperar brevemente para uma resposta do encaminhador antes de tentar entrar em contato com os servidores DNS em suas dicas de raiz.
  - Quando um servidor DNS encaminha uma consulta para um encaminhador ele envia uma consulta recursiva para o encaminhador. Isso é diferente do que a consulta iterativa que um servidor DNS irá enviar a um outro servidor DNS durante a resolução de nomes padrão (resolução de nomes que não envolve um encaminhador).

Configurando e usando assinatura de transações (TSIG)

- TSIG [3]
  - TSIG (assinatura de transação) é um protocolo de rede de computador definido na RFC 2845. Ele é usado principalmente pelo Domain Name System (DNS) para fornecer um meio de autenticar as atualizações de um banco de dados DNS. É mais comumente usado para atualizar o DNS dinâmico ou um servidor DNS secundário/slave. O TSIG usa chaves secretas compartilhadas e hash unidirecional para fornecer um meio seguro criptográfico de autenticação de cada ponta de uma conexão, sendo permitido fazer ou responder a uma atualização de DNS.
  - Embora as consultas ao DNS podem ser feitas de forma anônima (ver DNSSEC), alterações de DNS devem ser autenticadas, uma vez que fazem mudanças duradouras para a estrutura do sistema de nomes da Internet. Como a solicitação de atualização pode ser feita através de um canal inseguro (a Internet), deve-se tomar medidas para garantir a autenticidade e integridade do pedido. O uso de uma chave partilhada pelo cliente que faz a atualização e o servidor de DNS, ajuda a garantir a autenticidade e integridade do pedido de atualização. A função hash unidirecional é usado para prevenir observadores maliciosos de modificar a atualização e encaminhamento para o destino, e assim, garantir a integridade da mensagem da origem para o destino.

- O timestamp é incluído no protocolo TSIG para evitar que respostas registradas sejam reutilizadas, o que permitiria um invasor violar a segurança de TSIG. Isso coloca uma exigência em servidores DNS dinâmicos e clientes TSIG de conterem um relógio preciso. Desde que servidores DNS são conectados a uma rede, o Network Time Protocol pode ser usado para fornecer uma fonte de tempo preciso.
- Atualizações de DNS, como consultas, normalmente são transportados via UDP, uma vez que requer uma sobrecarga menor do que o TCP. No entanto, os servidores DNS suportam ambas as solicitações UDP e TCP.

Consciência do DNSSEC e ferramentas básicas

- DNSSEC [4]
  - O Domain Name System Security Extensions (DNSSEC) é um conjunto de especificações Internet Engineering Task Force (IETF) para garantir certos tipos de informações fornecidas pelo Domain Name System (DNS), usado em redes de Protocolo de Internet (IP). É um conjunto de extensões para DNS que fornecem aos clientes DNS (resolvedores) autenticação da origem dos dados do DNS, negação autenticada de existência e integridade dos dados, mas não disponibilidade ou confidencialidade.
  - Visão geral
    - O projeto original do Domain Name System (DNS) não incluia a segurança; em vez disso, foi projetado para ser um sistema distribuído escalável. O Domain Name System Security Extensions (DNSSEC) tenta adicionar segurança, mantendo a compatibilidade com versões anteriores. A RFC 3833 documenta algumas das ameaças conhecidas do DNS e como DNSSEC responde a essas ameaças.
    - O DNSSEC foi projetado para proteger as aplicações (e resolvedores de cache que servem essas aplicações) de usar dados do DNS falsos ou manipulados, como a criada por envenenamento de cache DNS. Todas as respostas provenientes de zonas protegidas DNSSEC são assinadas digitalmente. Ao verificar a assinatura digital, um resolvedor DNS é capaz de verificar se a informação é idêntica (ou seja, sem modificações e completa) com a informação publicada pela zona titular e servido em um servidor de DNS autoritário. Enquanto proteger os endereços IP é a preocupação imediata para muitos usuários, o DNSSEC pode proteger todos os dados publicados no DNS, incluindo registros de texto (TXT), registros de troca de correio (MX), e pode ser usado para iniciar outros sistemas de segurança que publicam as referências a certificados criptográficos armazenados no DNS, como Certificados de Registros (registros CERT, RFC 4398), impressões digitais SSH (SSHFP, RFC 4255), chaves públicas IPSec (IPSECKEY, RFC 4025) e Âncoras de Confiança TLS (TLSA, RFC 6698).
    - O DNSSEC não garante a confidencialidade dos dados; em particular, todas as respostas DNSSEC são autenticados, mas não criptografadas. Ele não protege contra ataques DoS diretamente, ainda que indiretamente, fornece algum benefício (porque a verificação de assinatura permite o uso de partes potencialmente não confiáveis; isso é verdade somente se o servidor DNS está usando um certificado auto-assinado, não recomendado para servidores DNS voltados para a Internet).
    - Outras normas (não DNSSEC) são usadas para proteger dados em massa (como uma transferência de zona DNS) enviadas entre os servidores DNS. Conforme documentado na IETF RFC 4367, alguns usuários e desenvolvedores fazem falsas suposições sobre nomes DNS, como assumir que o nome comum de uma empresa mais ".com" é sempre o seu nome de domínio. O DNSSEC não pode proteger contra

falsas suposições; ele só pode autenticar que os dados são realmente ou não, disponíveis a partir do proprietário do domínio.
    - As especificações DNSSEC (chamadas DNSSEC-bis) descrevem o protocolo DNSSEC atual em grande detalhe. Veja RFC 4033, RFC 4034 e RFC 4035. Com a publicação dessas novas RFCs (Março de 2005), uma RFC anterior, RFC 2535, tornou-se obsoleta.
    - Acredita-se que proteger o DNS é extremamente importante para proteger a Internet como um todo, mas a implantação de DNSSEC especificamente tem sido dificultada (partir de 22 de Janeiro de 2010) por várias dificuldades:
        - A necessidade de conceber um padrão compatível para trás (back-compatibility) que pode ser dimensionada para o tamanho da Internet
        - Prevenção de "enumeração de zonas" onde desejar
        - Implantação de implementações de DNSSEC em uma ampla variedade de servidores DNS e resolvedores (clientes)
        - Desacordo entre implementadores sobre quem deve possuir as chaves de raiz de domínio de nível superior
        - Superar a complexidade percebida do DNSSEC e sua implantação
    - O Microsoft Windows usa um resolvedor stub, e o Windows 7 e superiores, em particular, utilizam um resolvedor stub sem validação (mas consciente do DNSSEC). Para o resolvedor stub sem validação colocar qualquer dependência real sobre serviços DNSSEC, o resolvedor stub deve confiar em ambos os servidores de nome recursivos em questão (que normalmente são controlados pelo Provedor de Serviços de Internet) e os canais de comunicação entre si e os servidores de nome, utilizando métodos como IPsec, SIG(0), ou TSIG. O uso de IPsec é não generalizado.
- Ferramentas
    - dnssec-keygen(8) - ferramenta de geração de chave DNSSEC
    - dnssec-signzone(8) - ferramenta de assinatura de zona DNSSEC

Termos e utilitários

- /etc/named.conf - (5) arquivo de configuração para o named
- /etc/passwd - (5) arquivo de senha
- DNSSEC
- dnssec-keygen - (8) ferramenta de geração de chave DNSSEC
- dnssec-signzone - (8) ferramenta de assinatura de zona DNSSEC

Referências

1. http://en.wikipedia.org/wiki/Chroot
2. https://technet.microsoft.com/en-us/library/cc782142(v=ws.10).aspx
3. http://en.wikipedia.org/wiki/TSIG
4. http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

Exercícios práticos

1. Configurando o BIND para executar em uma jaula chroot

a. Parar o serviço named, se estiver em execução (Ex.: service <serviço> stop)

b. Listar o conteúdo do diretório /var/named (Ex.: ls <diretório>)

c. Exibir o conteúdo do arquivo /etc/sysconfig/named (Ex.: cat <arquivo>)

d. Exibir o conteúdo do script /etc/init.d/named (Ex.: cat <arquivo>)

e. Instalar o pacote bind-chroot (Ex.: yum install <pacote>)

f. Listar o conteúdo do pacote bind-chroot (Ex.: rpm <opções>)

g. Iniciar o serviço named (Ex.: service <serviço> start)

h. Verificar os sistemas de arquivos montados (Ex.: mount)

2. Dividir a configuração do BIND usando encaminhamentos
   a. Através do comando dig, consultar o example.com usando o servidor de nomes local (Ex.: dig <opções>)

   b. Visualizar os últimos registros gerais do sistema (Ex.: tail <arquivo>)

   c. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
      i. Visão Interna
         1. Encaminhar (forward): Apenas
         2. Encaminhadores (forwarders): 8.8.8.8

   d. Reiniciar o serviço named (Ex.: service <serviço> restart)

   e. Através do comando dig, consultar o domínio example.com, usando o servidor de nomes local (Ex.: dig <opções>)

   f. Visualizar os últimos registros gerais do sistema (Ex.: tail <arquivo>)

   g. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
      i. Visão Interna
         1. Zona
            a. Nome: google.com
            b. Classe: Internet
            c. Tipo: Encaminhar
            d. Encaminhar: Apenas
            e. Encaminhadores: 8.8.4.4

   h. Recarregar o serviço named (Ex.: rndc <opções>)

   i. Resolver o endereço www.google.com, usando o servidor de nomes local (Ex.: host <opções>)

3. Configurando e usando assinatura de transações (TSIG)
    a. Verificar o conteúdo do diretório /var/named/slaves (Ex.: ls <diretório>)

    b. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
        i. Visão Interna
            1. Zona
                a. Nome: subdomain.example.cluster
                b. Tipo: Escravo
                c. Arquivo: “slaves/subdomain.example.cluster.zone”
                d. Mestres: 10.200.30.<quarto campo do IP principal>

    c. Recarregar o serviço named (Ex.: rndc <opções>)

    d. Verificar o conteúdo do diretório /var/named/slaves (Ex.: ls <diretório>)

    e. Alterar a raiz para o diretório /srv/bind-chroot (Ex.: chroot <diretório>)

    f. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
        i. Chave
            1. Nome: repl-example.cluster-zones
            2. Algorítimo: hmac-md5
            3. Chave: “kW3zYJUtvFYH+FzepVPkWw==”
        ii. Servidor
            1. Endereço: 10.200.<terceiro campo do IP principal>.<quarto campo do IP principal>
            2. Chave: repl-example.cluster-zones

    g. Recarregar o serviço named (Ex.: rndc <opções>)

    h. Encerrar a sessão atual do bash para voltar a raiz do sistema de arquivos principal (Ex.: exit)

    i. No arquivo de configuração do named, declarar (Ex.: vi <arquivo>)
        i. Chave
            1. Nome: repl-example.cluster-zones
            2. Algorítimo: hmac-md5
            3. Chave: “kW3zYJUtvFYH+FzepVPkWw==”
        ii. Servidor
            1. Endereço: 10.200.30.<quarto campo do IP principal>
            2. Chave: repl-example.cluster-zones

    j. Retransferir a zona subdomain.example.cluster (rndc <opções>)

    k. Visualizar os últimos registros gerais do sistema (Ex.: tail <arquivo>)

    l. Recarregar o serviço named (Ex.: rndc <opções>)

    m. Retransferir a zona subdomain.example.cluster (rndc <opções>)

n. Visualizar os últimos registros gerais do sistema (Ex.: tail <arquivo>)

o. Habilitar o SELinux temporariamente (setenforce 1)

Simulado

1. As declarações …, … e … do arquivo de configuração do named são usadas respectivamente para especificar as informações de chaves para uso em autenticação e autorização usando TSIG, definir chaves DNSSEC de confiança e listar chaves DNSSEC para serem mantidas atualizadas usando manutenção de âncoras de confiança - RFC 5011.

2. As opções de chaves TSIG podem ser usadas em escopo global, opções globais, canais de controle, visões e zonas, enquanto as configurações de DNSSEC podem ser usadas nas opções globais, visões e zonas.
   a. V
   b. F

3. O uso do BIND enjaulado previne que um comprometimento do serviço diminua os riscos de acesso não autorizado ao sistemas de arquivos do servidor.
   a. V
   b. F

4. Para se utilizar o BIND em uma jaula chroot, basta informar como argumento para o daemon named, o diretório ao qual se pretende usar como raiz.
   a. V
   b. F

5. Usar encaminhamentos de resolução de nome reduz o tráfego de informações externas no serviço DNS local, aumentando o desempenho e melhorando a segurança do servidor.
   a. V
   b. F

6. É possível usar encaminhamento de forma global ou por domínios (encaminhamento condicional).
   a. V
   b. F

7. Após definir uma chave TSIG, deve-se referenciá-las nas opções dos recursos que irão utilizá-las.
   a. V
   b. F

8. As chaves TSIG são usadas para garantir a autenticidade de atualizações, mas não das consultas.
   a. V
   b. F

9. O DNSSEC somente garante a autenticidade e integridade da informação. A confidencialidade não pode ser garantida pelo mesmo, uma vez que os dados não são criptografados.
   a. V

b. F

10. As ferramentas … e … são usadas respectivamente para gerar chaves de assinatura e assinar arquivos de zonas.

Tópico 208: Serviços Web

208.1 Implementando um servidor web

Visão geral

Peso: 4
Descrição: Os candidatos devem ser capazes de instalar e configurar um servidor web. Esse objetivo inclui monitorando a carga e a performance do servidor, restringindo acesso do usuário cliente, configurando suporte para linguagens de script como módulos e definindo autenticação para usuários clientes. Também inclui configurando opções de servidor para restringir o uso de recursos. Os candidatos devem ser capazes de configurar o servidor web para usar hosts virtuais e personalizar o acesso a arquivos.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários do Apache 2.x
- Arquivos de configuração de logs do Apache e conteúdo
- Métodos de restrição de acesso e arquivos
- Configuração do mod_perl e PHP
- Arquivos de configuração de autenticação de clientes e utilitários
- Configuração do máximo de requisições, mínimo e máximo de servidores e clientes
- Usando redirecionamentos nos arquivos de configuração do Apache para personalizar o acesso a arquivos

Termos e utilitários:

- logs de acesso e logs de erro
- .htaccess
- httpd.conf
- mod_auth
- htpasswd
- AuthUserFile, AuthGroupFile
- apache2ctl
- httpd

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários do Apache 2.x

- Apache HTTP Server [1]
    - O Apache HTTP Server, coloquialmente chamado Apache, é o software de servidor web mais utilizado do mundo. Originalmente baseado no servidor NCSA HTTPd, o desenvolvimento do Apache começou no início de 1995, após o trabalho no código NCSA paralisar. O Apache desempenhou um papel fundamental no crescimento inicial da World Wide Web, rapidamente ultrapassando o NCSA HTTPd como o servidor HTTP dominante, e manteve-se o servidor HTTP mais popular desde Abril de 1996. Em 2009, tornou-se o primeiro software de servidor web a atender mais de 100 milhões de websites.
    - Apache é desenvolvido e mantido por uma comunidade aberta de desenvolvedores, sob os auspícios da Apache Software Foundation. Mais comumente usado em um sistema Unix-like (geralmente Linux), o software está disponível para uma grande variedade de sistemas operacionais, incluindo Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows, OS/2 , TPF, OpenVMS e eComStation. Lançado sob a licença Apache, Apache é um software livre e de código aberto.

- Em Junho de 2013, o Apache foi estimado em servir 54,2% de todos os sites ativos e 53,3% dos principais servidores em todos os domínios.
- Características
  - O Apache suporta uma variedade de recursos, muitos implementados como módulos compilados que ampliam a funcionalidade do núcleo. Esses podem variar de suporte a linguagem de programação server-side, à sistemas de autenticação. Algumas interfaces de linguagem comuns suportam Perl, Python, Tcl, e PHP. Os módulos de autenticação mais populares incluem mod_access, mod_auth, mod_digest e mod_auth_digest, o sucessor do mod_digest. Uma amostra de outras características incluem Secure Sockets Layer e suporte a Transport Layer Security (mod_ssl), um módulo de proxy (mod_proxy), um reescrevedor de URL (mod_rewrite), arquivos de log personalizado (mod_log_config) e suporte de filtragem (mod_include e mod_ext_filter).
  - Métodos de compressão populares no Apache incluem o módulo de extensão externa, mod_gzip, implementado para ajudar com a redução do tamanho (peso) de páginas Web servidas sobre HTTP. O ModSecurity é um mecanismo de detecção e prevenção de intrusão de código aberto para aplicações Web. Os logs do Apache podem ser analisados através de um navegador da Web usando scripts livres, como AWStats/W3Perl ou Visitors.
  - Virtual Hosting permite uma instalação do Apache servir muitos sites diferentes. Por exemplo, uma máquina com uma instalação do Apache pode servir simultaneamente www.example.com, www.example.org, test47.test-server.example.edu, etc.
  - O Apache apresenta mensagens de erro configuráveis, bases de dados de autenticação baseados em DBMS e negociação de conteúdo. Ele também é suportado por várias interfaces gráficas de usuário (GUIs).
  - Ele suporta a autenticação de senha e autenticação de certificado digital. Como o código fonte está disponível livremente, qualquer um pode adaptar o servidor para necessidades específicas, e há uma grande biblioteca pública de add-ons Apache.
- Performance
  - Em vez de implementar uma arquitetura única, O Apache fornece uma variedade de módulos de multiprocessamento (MPMs), que permitem o Apache executar baseado em processos, híbrido (processo e thread) ou em um modo de evento-híbrido, para melhor atender as demandas de cada infraestrutura especial. Isso implica que a escolha correta do MPM e a configuração correta é importante. Onde compromissos de desempenho precisam ser feitos, o desenho do Apache é reduzir a latência e aumentar o tráfego, em relação a simplesmente manipular mais pedidos, garantindo assim o processamento consistente e confiável de pedidos dentro de prazos razoáveis.
  - Para a entrega de páginas estáticas, a série do Apache 2.2 foi considerada significativamente mais lenta do que nginx. Para resolver essa questão, a versão Apache considerada pela Fundação Apache como provedora de alta performance é a versão multi-threaded, que mistura o uso de vários processos e várias threads por processo. Essa arquitetura, e da forma como foi implementada na série do Apache 2.4, prevê um desempenho equivalente ou ligeiramente melhor do que servidores web baseados em eventos, como é alegado pelo presidente da Fundação Apache, Jim Jagielski. No entanto, alguns benchmarks independentes mostram que ainda é a metade da velocidade nginx.

- Arquivos de configuração

    - RH e derivados
        - /etc/httpd/conf/httpd.conf - arquivo de configuração principal
        - /etc/httpd/conf.d/ - diretório para inclusão de arquivos adicionais *.conf
    - Debian e derivados
        - /etc/apache2/apache2.conf - arquivo de configuração principal
        - /etc/apache2/ports.conf - arquivo de configuração de escuta de portas
        - /etc/apache2/conf.d/ - diretório para inclusão de arquivos adicionais *.conf
        - /etc/apache2/sites-available/ - diretório que contêm a configuração de todos os virtual hosts para definir diferentes web sites
        - /etc/apache2/sites-enabled/ - diretório que estabelece quais virtual hosts estão sendo usados (links simbólicos para os arquivos do /etc/apache2/sites-available)
        - /etc/apache2/mods-available/ - diretório que estabelece quais módulos estão disponíveis
        - /etc/apache2/mods-enabled/ - diretório que estabelece quais módulos estão sendo usados (links simbólicos para os arquivos do diretório /etc/apache2/mods-available)
- Utilitários
    - apache2ctl(apachectl(8)) - Interface de Controle do Apache HTTP Server
    - Exemplos:
        - apachectl start - inicia o servidor httpd
        - apachectl stop - para o servidor httpd
        - apachectl restart - reinicia o servidor httpd
        - apachectl status - verifica o estado do servidor httpd
        - apachectl fullstatus - exibe o relatório do mod_status
        - apachectl configtest - executa um teste de sintaxe na configuração
    - httpd(8) - Apache Hypertext Transfer Protocol Server
        - Exemplos:
            - httpd - inicia o servidor httpd
            - httpd -d <diretório> - define o DocumentRoot padrão
            - httpd -f <arquivo> - define o arquivo de configuração
            - httpd -k <ação> - inicia|reinicia|para o serviço httpd
            - httpd -l - lista os módulos compilados
            - httpd -M - despeja uma lista de módulos estaticos e dinâmicos carregados
            - httpd -S - exibe as definições interpretadas do arquivo de configuração (atualmente só mostra definições de virtualhost)
            - httpd -t - executa um teste de sintaxe na configuração

Arquivos de configuração de logs do Apache e conteúdo

- Arquivos de log [2]
    - A fim de gerenciar eficazmente um servidor web, é necessário obter feedback sobre a atividade e desempenho do servidor, bem como os problemas que possam estar ocorrendo. O Apache HTTP Server fornece recursos de registro muito abrangentes e flexíveis.
    - Log de erro
        - O log de erro do servidor, cujo nome e localização é definida pela diretiva ErrorLog, é o arquivo de log mais importante. Esse é o lugar onde o Apache httpd irá enviar informações de diagnóstico e registrar quaisquer erros que ele encontra no processamento de requisições. É o primeiro lugar para olhar quando ocorre um problema com a inicialização do servidor ou com o funcionamento do servidor, uma vez que, muitas vezes, contêm detalhes sobre o que deu errado e como corrigí-lo.

- O log de erro geralmente é gravado em um arquivo (normalmente error_log em sistemas Unix e error.log no Windows e OS/2). No sistema Unix, é também possível que o servidor envie erros ao syslog ou canalizá-los para um programa.
- O formato do log de erro é relativamente de forma livre e descritivo. Mas há certas informações que estão contidas na maioria dos entradas de log de erro. Por exemplo, aqui está uma mensagem típica.
    - [Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration: /export/home/live/ap/htdocs/test
- O primeiro item na entrada do registro é a data e hora da mensagem. O segundo item lista a gravidade do erro que está sendo relatado. A diretiva LogLevel é usada para controlar os tipos de erros que são enviados para o log de erro ao restringir o nível de gravidade. O terceiro item dá o endereço IP do cliente que gerou o erro. Além disso é a própria mensagem, que, neste caso, indica que o servidor foi configurado para negar o acesso do cliente. O servidor relata o caminho do sistema de arquivos (em oposição ao caminho web) do documento solicitado.
- Uma grande variedade de mensagens diferentes podem aparecer no log de erro. A maioria parece similar ao exemplo acima. O log de erro irá conter também a depuração de saída a partir de scripts CGI. Qualquer informação escrita para o stderr por um script CGI será copiada diretamente para o log de erro.
- Não é possível personalizar o log de erro, adicionando ou removendo informações. No entanto, as entradas de log de erro tratando determinadas solicitações têm entradas correspondentes no log de acesso. Por exemplo, a entrada do exemplo acima corresponde a uma entrada de log de acesso com código de status 403. Uma vez que é possível personalizar o log de acesso, você pode obter mais informações sobre as condições de erro usando esse arquivo de log.

- Log de acesso
    - O log de acesso do servidor registra todas as requisições processadas pelo servidor. A localização e o conteúdo do log de acesso são controladas pela diretiva CustomLog. A diretiva LogFormat pode ser usada para simplificar a seleção dos conteúdos dos registros.
    - Várias versões do Apache httpd têm usado outros módulos e diretrizes para controlar o log de acesso, incluindo mod_log_referer, mod_log_agent e a diretiva TransferLog. A diretiva CustomLog agora agrupa as funcionalidades de todas as diretivas mais velhas.
    - O formato do log de acesso é altamente configurável. O formato é especificado usando uma seqüência de formato que se parece muito com uma string de formato printf(1) estílo C.
    - Common Log Format
        - Uma configuração típica para o log de acesso pode parecer como se segue.
            - LogFormat "%h %l %u %t \"%r\" %>s %b" common
            - CustomLog logs/access_log common
        - Isso define o apelido comum e o associa com uma seqüência de formato de log particular. A seqüência de formato consiste de diretivas por cento, cada um dos quais diz ao servidor para registrar uma determinada peça de informação. Caracteres literais também podem ser colocados na seqüência de formato e serão copiados diretamente para a saída de log. O caractere de aspas (") deve ser precedido pela colocação de uma barra invertida antes para evitar que ele seja interpretado como o fim da seqüência de formato. A

seqüência de formato também pode conter os caracteres de controle especial "\n" para nova linha e "\t" para tab.
- A diretiva CustomLog estabelece um novo arquivo de log usando o apelido definido. O nome do arquivo de log de acesso é relativo ao ServerRoot a menos que comece com uma barra.
- A configuração acima vai escrever entradas de log em um formato conhecido como o Formato de Log Comum (CLF). Esse formato padrão pode ser produzido por muitos servidores web diferentes e lido por muitos programas de análise de log. As entradas do arquivo de log produzidos em CLF será algo parecido com isso:
  - 127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

■ Combined Log Format
- Outra seqüência de formato comumente usado é chamado o Formato de Log Combinado. Ele pode ser usado como segue:
  - LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
  - CustomLog log/access_log combined
- Esse formato é exatamente o mesmo que o formato log comum, com a adição de mais dois campos. Cada um dos campos adicionais usa a diretiva por cento %{header}i, onde header pode ser qualquer cabeçalho de solicitação HTTP. O log de acesso sob esse formato será parecido com:
  - 127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I ;Nav)"

○ Logs condicionais

■ Há momentos em que é conveniente excluir determinadas entradas dos logs de acesso com base em características da requisição do cliente. Isso é facilmente conseguido com a ajuda de variáveis de ambiente. Em primeiro lugar, uma variável de ambiente deve ser definida para indicar que a requisição atende certas condições. Isso geralmente é realizado com SetEnvIf. Em seguida, o env=cláusula da diretiva CustomLog é usado para incluir ou excluir as requisições em que a variável de ambiente é definida. Alguns exemplos:
- # Marca requisições da interface loopback
- SetEnvIf Remote_Addr "127\.0\.0\.1" dontlog
- # Marca requisições para o arquivo robots.txt
- SetEnvIf Request_URI "^/robots\.txt$" dontlog
- # Registra o que sobra
- CustomLog logs/access_log common env=!dontlog

■ Como outro exemplo, considerar as requisições de registro de falantes Inglês para um arquivo de log, e quem não fala inglês em um arquivo de log diferente.
- SetEnvIf Accept-Language "en" english
- CustomLog logs/english_log common env=english
- CustomLog logs/non_english_log common env=!english

■ Embora tenha-se acabado de mostrar que o registro condicional é muito poderoso e flexível, não é a única maneira de controlar o conteúdo dos logs. Os arquivos de log são mais úteis quando contêm um registro completo da atividade do servidor. Muitas

vezes, é mais fácil simplesmente pós-processar os arquivos de log para remover os pedidos que não se quer considerar.

Métodos de restrição de acesso e arquivos

- Controle de acesso [3]
  - O controle de acesso refere-se a qualquer meio de controlar o acesso a qualquer recurso. Esse é separado da autenticação e autorização.
  - Módulos e diretivas relacionados
    - O controle de acesso pode ser feito por vários módulos diferentes. O mais importante deles é mod_authz_host. Outros módulos incluem mod_setenvif e mod_rewrite.
  - Controle de acesso por host
    - Se quiser restringir o acesso a partes do site com base no endereço de host de visitantes, isso é mais facilmente feito usando o mod_authz_host.
    - As diretivas Allow e Deny permitem e negam o acesso com base no nome do host ou endereço do host, da máquina requisitando um documento. A diretiva Order caminha lado a lado com essas duas, e diz ao Apache em qual ordem aplicar os filtros.
    - O uso destas diretivas é:
      - Allow from <endereço>
    - onde endereço é um endereço IP (ou um endereço de IP parcial) ou um nome de domínio totalmente qualificado (ou um nome de domínio parcial); pode-se fornecer vários endereços ou nomes de domínio, se desejar.
    - Por exemplo, caso se tem alguém a enviar spam para o quadro de mensagem, e quer mantê-lo para fora, pode-se fazer o seguinte:
      - Deny from 10.252.46.165
    - Os visitantes provenientes desse endereço não serão capazes de ver o conteúdo abrangido pela diretiva. Se, em vez disso, tem-se um nome de máquina, em vez de um endereço IP, pode-se usar isso.
      - Deny from host.example.com
    - E, caso se gostaria de bloquear o acesso de um domínio inteiro, pode-se especificar apenas parte de um nome de endereço ou de domínio:
      - Deny from 192.168.205
      - Deny from phishers.example.com moreidiots.example
      - Deny from ke
    - Usando Order vai deixa-se ter certeza de que se realmente está restringindo as coisas para o grupo que pretende deixar entrar, pela combinação das diretivas Allow e Deny
      - Order deny,allow
      - Deny from all
      - Allow from dev.example.com
    - Listando apenas a diretiva Allow não iria fazer o que se quer, porque vai deixar pessoas daquele host entrar, além de deixar todo mundo. O que se quer é deixar apenas essas pessoas entrar.
  - Controle de acesso por variáveis de ambiente
    - mod_authz_host, em conjunto com mod_setenvif, pode ser usado para restringir o acesso ao site com base no valor de variáveis de ambiente arbitrárias. Isso é feito com a sintaxe Allow from env= e Deny from env=.

                - SetEnvIf User-Agent BadBot GoAway=1
                - Order allow,deny
                - Allow from all
                - Deny from env=GoAway
            - No exemplo acima, a variável de ambiente GoAway é definida como 1 se o User-Agent corresponde a string BadBot. Em seguida, nega o acesso para qualquer pedido quando essa variável está definida. Isso bloqueia esse agente a um determinado do site.
            - Um teste de variável de ambiente pode ser negado usando a sintaxe =!:
                - Allow from env=!GoAway
    - Controle de acesso com mod_rewrite
        - A flag [F] do RewriteRule provoca uma resposta Forbidden 403 para ser enviada. Usando isso, pode-se negar o acesso a um recurso baseado em critérios arbitrários.
        - Por exemplo, caso se deseja bloquear o acesso a um recurso entre 20:00 e 06:00, pode-se fazer isso usando mod_rewrite.
            - RewriteEngine On
            - RewriteCond %{TIME_HOUR} >20 [OR]
            - RewriteCond %{TIME_HOUR} <07
            - RewriteRule ^/fridge - [F]
        - Isso irá retornar uma resposta Forbidden 403 para qualquer pedido após 08:00 ou antes de 07:00. Essa técnica pode ser usada para qualquer critério que se deseja verificar. Também se pode redirecionar, ou de outra forma, reescrever esses pedidos, se é preferível a abordagem.

Configuração do mod_perl e PHP

- mod_perl [4]
    - mod_perl é um módulo opcional para o servidor Apache HTTP. Ele incorpora um interpretador Perl para o servidor Apache. Além de permitir que os módulos do Apache sejam escritos em Perl, ele permite que o servidor web Apache seja configurado de forma dinâmica por programas Perl. No entanto, o seu uso mais comum é para que o conteúdo dinâmico produzido por scripts Perl, possa ser servido em resposta a requisições de entrada, sem a sobrecarga significativa de relançar o interpretador Perl para cada solicitação.
    - Slashcode, que administra o site Slashdot, é escrito usando mod_perl. As primeiras versões do PHP foram implementados em Perl usando mod_perl.
    - Comparação ao Common Gateway Interface (CGI)
        - O mod_perl pode emular um ambiente de Common Gateway Interface (CGI), para que os scripts Perl CGI existentes possam se beneficiar do aumento de desempenho sem ter que ser reescritos.
        - Ao contrário do CGI (e a maioria dos outros ambientes de aplicativos web), o mod_perl fornece acesso completo a API Apache, permitindo programadores a escrever manipuladores para todas as fases do ciclo de requisição Apache, manipular tabelas internas e mecanismos de estado do Apache, compartilhar dados entre processos ou threads do Apache, alterar ou estender o analisador de arquivo de configuração do Apache, e adicionar o código Perl para o próprio arquivo de configuração, entre outras coisas.
    - Configuração [5]
        - Para carregar o mod_perl construido como um DSO, adicione ao httpd.conf

        - LoadModule perl_module modules/mod_perl.so
    - Se quiser rodar um código mod_perl 1.0 em um servidor mod_perl 2.0, habilite a camada de compatibilidade
        - PerlModule Apache2::compat
    - Para registrar scripts adicione ao httpd.conf
        - <Contexto de localização ou diretório>
        - SetHandler perl-script
        - PerlResponseHandler ModPerl::Registry
        - PerlOptions +ParseHeaders
        - Options +ExecCGI
- PHP [6]
    - PHP é uma linguagem de script do lado do servidor projetada para o desenvolvimento web, mas também usada como uma linguagem de programação de propósito geral. A partir de Janeiro de 2013, o PHP foi instalado em mais de 240 milhões de sites (39% da amostragem) e 2,1 milhões de servidores web. Criado originalmente por Rasmus Lerdorf em 1994, a implementação de referência do PHP (alimentado pelo Zend Engine) agora é produzido pelo Grupo PHP. Enquanto o PHP originalmente significava Personal Home Page, agora está para PHP:Hypertext Preprocessor, que é um backronym recursivo.
    - O código PHP pode ser simplesmente misturado com o código HTML, ou pode ser usado em combinação com vários motores de templates e frameworks da web. O código PHP geralmente é processado por um intérprete PHP, o que geralmente é implementado como módulo nativo de um servidor web ou um executável Common Gateway Interface (CGI). Depois que o código PHP é interpretado e executado, o servidor web envia o resultante de saída para o seu cliente, geralmente na forma de uma parte da página web gerada; por exemplo, o código PHP pode gerar o código de uma página web HTML, uma imagem, ou alguns outros dados. O PHP também evoluiu para incluir uma interface de linha de comando (CLI) e pode ser usado em aplicações gráficas independentes.
    - O intérprete PHP canônico, alimentado pelo Zend Engine, é um software livre liberado sob a licença PHP. O PHP tem sido amplamente portado e pode ser implantado na maioria dos servidores web em quase todos os sistemas operacionais e plataformas, de forma gratuita.
    - Apesar de sua popularidade, nenhuma especificação escrita ou padrão existia para a linguagem PHP até 2014, deixando o intérprete PHP canônico como um padrão de fato. Desde 2014, há trabalhos em curso sobre a criação de uma especificação formal PHP.
    - Instalação e configuração
        - Há duas maneiras principais para adicionar suporte para PHP em um servidor web - como um módulo de servidor web nativo, ou como um executável CGI. O PHP tem uma interface direta de módulo chamado Server Application Programming Interface (SAPI), que é suportado por muitos servidores web, incluindo Apache HTTP Server, Microsoft IIS, Netscape (hoje extinto) e iPlanet. Alguns outros servidores web, tais como OmniHTTPd, suportam a Internet Server Application Programming Interface (ISAPI), que é uma interface de módulo de servidor web da Microsoft. Se o PHP não tiver suporte de módulo para um servidor web, ele sempre pode ser usado como um Common Gateway Interface (CGI) ou processador FastCGI; nesse caso, o servidor web está configurado para usar executável CGI do PHP para processar todas as solicitações para arquivos PHP.
        - PHP-FPM (FastCGI Process Manager) é uma implementação FastCGI alternativa para PHP, que vem com a distribuição PHP oficial desde a versão 5.3.3. Quando

comparado com a implementação mais velha FastCGI, contém algumas características adicionais, principalmente útil para servidores web muito carregados.
- Quando se usa o PHP para script de linha de comando, um executável de interface de linha de comando (CLI) PHP é necessário. O PHP suporta um SAPI CLI a partir do PHP 4.3.0. O foco principal desse SAPI é o desenvolvimento de aplicações shell usando PHP. Existem algumas diferenças entre a CLI SAPI e outras SAPIs, embora eles compartilham muitos dos mesmos comportamentos.
- O PHP também pode ser usado para a escrita de aplicações desktop de interface gráfica do usuário (GUI), usando a extensão PHP-GTK. O PHP-GTK não está incluído na distribuição oficial do PHP, e como uma extensão pode ser usado somente com as versões do PHP 5.1.0 e mais recentes. A forma mais comum de instalar o PHP-GTK é compilá-lo a partir do código-fonte.
- Quando o PHP é instalado e utilizado em ambientes de nuvem, kits de desenvolvimento de software (SDKs) são fornecidos para o uso de recursos específicos de nuvem. Por exemplo:
  - Amazon Web Services fornece o SDK AWS para PHP
  - Windows Azure pode ser usado com o Windows Azure SDK para PHP
- Inúmeras opções de configuração são suportadas, afetando tanto os recursos do núcleo e extensões PHP. O arquivo de configuração php.ini é procurado em locais diferentes, dependendo da maneira como PHP é usado. O arquivo de configuração é dividido em vários seções, enquanto algumas das opções de configuração pode também ser definido dentro da configuração do servidor web.

- Apache 2.x em sistemas Unix e derivados [7]
  - Não é recomendado o uso do PHP em um MPM com threads em produção com o Apache 2. Use o MPM prefork, que é o MPM padrão com Apache 2.0 e 2.2.
  - Configure o php.ini (normalmente /etc/php.ini)
  - Carregue o módulo php no Apache (LoadModule php5_module modules/libphp5.so)
  - Ative a manipulação de arquivos do tipo PHP
    - <FilesMatch \.php$>
    - SetHandler application/x-httpd-php
    - </FilesMatch>
  - Ou
    - AddHandler php5-script .php
    - AddType text/html .php
  - Reinicie o servidor web

Arquivos de configuração de autenticação de clientes e utilitários

- Autenticação e autorização [8]
  - A autenticação é qualquer processo pelo qual se verifica que alguém é quem diz que é. Autorização é qualquer processo pelo qual alguém é permitido estar onde ele quer ir, ou ter a informação que ele quer ter.
  - Módulos e diretivas relacionadas
    - Existem três tipos de módulos envolvidos no processo de autenticação e autorização. Geralmente se precisa escolher pelo menos um módulo de cada grupo.
    - Tipo de autenticação (diretiva AuthType)
      - mod_auth_basic
      - mod_auth_digest
    - Provedor de autenticação (diretivas AuthBasicProvider e AuthDigestProvider)

                - mod_authn_alias
                - mod_authn_dbd
                - mod_authn_file
                - mod_authnz_ldap
            - Autorização (diretiva Require)
                - mod_authnz_ldap
                - mod_authz_dbm
                - mod_authz_groupfile
                - mod_authz_host
                - mod_authz_user
            - O módulo mod_authnz_ldap é tanto um provedor de autenticação e autorização. O módulo mod_authn_alias não é um provedor de autenticação, por si só, mas permite que outros provedores de autenticação sejam configurado de forma flexível.
            - O módulo mod_authz_host fornece autorização e controle de acesso com base no nome do host, endereço IP ou as características da requisição, mas não é parte do sistema de provedor de autenticação.
        - Pré-requisitos
            - As diretrizes deverão estar no arquivo de configuração do servidor principal (normalmente em uma seção <Directory>), ou em arquivos de configuração por diretório (arquivos .htaccess).
            - Caso se pretenda usar arquivos .htaccess, será necessário ter uma configuração de servidor que permite colocar diretrizes de senha nesses arquivos. Isso é feito com a diretiva AllowOverride, que especifica quais são as diretivas, se houver, poderão ser colocadas em arquivos de configuração por diretório.
                - AllowOverride AuthConfig
- mod_auth_basic [9]
    - Esse módulo permite o uso de autenticação básica HTTP para restringir o acesso, observando os usuários dos provedores de dados. Autenticação HTTP Digest é fornecida por mod_auth_digest. Esse módulo deverá normalmente ser combinado com pelo menos um módulo de autenticação tal como mod_authn_file e um módulo de autorização como mod_authz_user.
    - Diretiva AuthBasicProvider
        - A diretiva AuthBasicProvider define qual provedor é usado para autenticar os usuários para este local. O provedor de arquivo padrão é implementado pelo módulo mod_authn_file.
        - Exemplo
            - <Location /secure>
            - AuthType basic
            - AuthName "private area"
            - AuthBasicProvider dbm
            - AuthDBMType SDBM
            - AuthDBMUserFile /www/etc/dbmpasswd
            - Require valid-user
            - </Location>
        - Provedores são consultados na ordem, até que um provedor encontre uma correspondência para o nome de usuário solicitado, altura em que esse único fornecedor tentará verificar a senha. Uma falha para verificar a senha resulta no controle não sendo repassado aos provedores subseqüentes.

        - Provedores são implementadas por mod_authn_dbd, mod_authn_file e mod_authnz_ldap.
- mod_authn_file [10]
    - Esse módulo fornece para frontends de autenticação tais como mod_auth_digest e mod_auth_basic autenticar usuários, observando os usuários em arquivos de senhas em texto puro. Funcionalidade semelhante é fornecida pelo mod_authn_dbd.
    - Ao usar mod_auth_basic ou mod_auth_digest, esse módulo é invocado pelo AuthBasicProvider ou AuthDigestProvider com o valor “file”.
    - Diretiva AuthUserFile
        - A diretiva AuthUserFile define o nome de um arquivo textual que contém a lista de usuários e senhas para autenticação do usuário. File-path é o caminho para o arquivo de usuário. Se não é absoluto, ela é tratada como relativo ao ServerRoot.
        - Cada linha do arquivo de usuário contém um nome de usuário seguido por dois pontos, seguido da senha criptografada. Se o mesmo ID do usuário é definido várias vezes, mod_authn_file vai usar a primeira ocorrência para verificar a senha.
        - O utilitário htpasswd que é instalado como parte da distribuição binária, ou que pode ser encontrado em src/support, é usado para manter o arquivo de senhas para autenticação básica HTTP. Veja a página de manual para mais detalhes. Em resumo:
            - Crie um arquivo de senhas com nome de usuário como o ID inicial. Ele irá solicitar a senha:
                - htpasswd -c <nome do arquivo> <nome do usuário>
            - Adicione ou modifique o nome de usuário 2 no arquivo de senhas:
                - htpasswd <nome do arquivo> <nome do usuário 2>
        - Note que busca em grandes arquivos de texto é muito ineficiente; AuthDBMUserFile deve ser usado em seu lugar.
        - Caso se estiver usando autenticação Digest HTTP, a ferramenta htpasswd não é suficiente. Tem-se que usar htdigest no lugar. Observe que não se pode misturar dados de usuário para autenticação Digest e autenticação básica dentro do mesmo arquivo.
- mod_authz_user [11]
    - Esse módulo fornece recursos de autorização para que aos usuários autenticados possam ser permitidos ou negados o acesso a partes do site.O mod_authz_user concede acesso se o usuário autenticado é listado na diretiva Require user. Alternativamente Require valid-user pode ser usado para conceder acesso a todos os usuários autenticados com sucesso.
- Exemplo
    - AuthType Basic
    - AuthName “Acesso restrito”
    - AuthBasicProvider file
    - AuthUserFile .htpasswd
    - Require valid-user

Configuração do máximo de requisições, mínimo e máximo de servidores e clientes

- MPM [12]
    - O servidor Apache HTTP é projetado para ser um servidor web poderoso e flexível que pode trabalhar em uma ampla variedade de plataformas, em uma variedade de diferentes ambientes. Diferentes plataformas e ambientes diferentes muitas vezes exigem características diferentes, ou podem ter diferentes formas de implementar o mesmo recurso

de forma mais eficiente. O Apache sempre têm acomodado uma grande variedade de ambientes através do seu design modular. Esse projeto permite que o webmaster escolha quais recursos serão incluídos no servidor selecionando quais módulos carregar tanto em tempo de compilação ou em tempo de execução.

- O Apache 2.0 estende esse design modular para as funções mais básicas de um servidor web. O servidor é fornecido com uma seleção de módulos de multiprocessamento (MPMs), que são responsáveis pela ligação às portas de rede na máquina, aceitando requisições e despachando processos filhos para lidar com as requisições.
- Estendendo o design modular para esse nível do servidor, permite dois importantes benefícios:
  - O Apache pode mais limpamente e eficientemente suportar uma ampla variedade de sistemas operacionais. Em particular, a versão Windows do Apache é agora muito mais eficiente, uma vez que o mpm_winnt pode usar recursos de rede nativas no lugar da camada POSIX usado no Apache 1.3. Esse benefício também se estende a outros sistemas operacionais que implementam MPMs especializados.
  - O servidor pode ser melhor personalizado para as necessidades do local particular. Por exemplo, os sites que precisam de uma grande quantidade de escalabilidade podem optar por usar um threaded MPM como worker ou event, enquanto sites que exigem estabilidade ou compatibilidade com programas antigos podem usar um prefork.
- No nível de usuário, MPMs aparecem bem como outros módulos do Apache. A principal diferença é que um e apenas um MPM tem de ser carregado no servidor, a qualquer momento. A lista de MPMs disponíveis aparece na página de índice de módulo.
- Escolhendo um MPM
  - MPMs devem ser escolhido durante a configuração, e compilados no servidor. Os compiladores são capazes de otimizar uma série de funções, se threads são usadas, mas apenas se eles sabem que threads estão sendo usadas.
  - Para realmente escolher o MPM desejado, use o argumento --with-mpm=<NOME> com o script configure. <NOME> é o nome do MPM desejado.
  - Uma vez que o servidor foi compilado, é possível determinar qual MPM foi escolhida usando ./httpd -l. Esse comando irá listar cada módulo que é compilado no servidor, incluindo o MPM.
- MPMs padrão
  - A tabela a seguir lista os MPMs padrão para vários sistemas operacionais. Esse será o MPM selecionado caso se não fizer outra escolha em tempo de compilação.
  - BeOS: beos
  - Netware: mpm_netware
  - OS/2: mpmt_os2
  - Unix: prefork
  - Windows: mpm_winnt
- MPM Prefork [13]
  - Um processo de controle único é responsável pelo lançamento de processos filho que escutam as conexões e as atendem quando elas chegam. O Apache sempre tenta manter vários processos do servidor de reserva ou ociosos, que estão prontos para atender as solicitações recebidas. Dessa forma, os clientes não precisam esperar por um novo processo filho a ser bifurcado antes de suas requisições serem atendidas.

        - O StartServers, MinSpareServers, MaxSpareServers e MaxClients regulam a forma como o processo pai cria filhos para atender as solicitações. Em geral, o Apache é muito auto-regulado, de modo que a maioria dos sites não precisa ajustar essas diretivas de seus valores padrão. Sites que precisam servir mais de 256 solicitações simultâneas podem precisar aumentar MaxClients, enquanto sites com memória limitada podem precisar diminuir MaxClients para evitar o servidor de entrar em modo “thrashing” (troca de memória para o disco e RAM). Mais informações sobre o processo de criação de ajustes são fornecidas na documentação de dicas de desempenho.
        - Enquanto o processo pai é normalmente iniciado como root sob o Unix, a fim de se vincular à porta 80, os processos filhos são lançados pelo Apache como um usuário com menos privilégios. As diretivas de usuário e de grupo são usadas para definir os privilégios dos processos filho do Apache. Os processos filhos devem ser capazes de ler todo o conteúdo que vai ser servido, mas devem ter o mínimo de privilégios para mais, possível.
        - MaxRequestsPerChild controla a freqüência com que o servidor recicla processos por matar os antigos e lançando novos.
    - MPM Worker [14]
        - Um processo de controle único (o pai) é responsável pelo lançamento de processos filho. Cada processo filho cria um número fixo de threads do servidor, conforme especificado na diretiva ThreadsPerChild, bem como uma thread ouvinte que recebe as conexões e as passa para um segmento do servidor para processamento quando elas chegam.
        - O Apache tenta sempre manter um pool de threads do servidor de espera ou ociosos, que estão prontos para atender as solicitações recebidas. Desta forma, os clientes não precisam esperar por uma nova threads ou processos a serem criados antes que suas requisições possam ser atendidas. O número de processos que lançará inicialmente é definido pela diretiva StartServers. Durante a operação, O Apache avalia o número total de segmentos ociosos em todos os processos, e bifurca ou mata os processos para manter esse número dentro dos limites especificados pelo MinSpareThreads e MaxSpareThreads. Uma vez que este processo é muito auto-regulado, raramente é necessário alterar essas diretivas de seus valores padrão. O número máximo de clientes que podem ser servidos simultaneamente (isto é, o número total máximo de segmentos em todos os processos) é determinado pela diretiva MaxClients. O número máximo de processos filho ativos é determinado pela diretiva MaxClients dividido pela diretiva ThreadsPerChild.
        - Duas diretivas definem limites rígidos sobre o número de processos filho ativos e o número de threads do servidor em um processo filho, e só pode ser alterado por parar completamente o servidor e, em seguida, iniciá-lo novamente. O ServerLimit é um limite rígido sobre o número de processos filho ativos, e deve ser maior ou igual à diretiva MaxClients dividido pela diretiva ThreadsPerChild. ThreadLimit é um limite rígido do número de threads do servidor, e deve ser maior ou igual a diretiva ThreadsPerChild. Se os valores não-padrão são especificados para essas diretrizes, eles devem aparecer antes das outras diretivas do worker.
        - Além do conjunto de processos filho ativos, podem haver processos filho adicionais, que são terminados, mas que pelo menos um segmento do servidor ainda está a lidar com uma conexão do cliente existente. Até o MaxClients processos de

terminação podem estar presente, embora o número real pode ser esperado a ser muito menor. Esse comportamento pode ser evitado, desativando a terminação de processos filhos individuais, o que é conseguido com a seguinte:
    - definir o valor da MaxRequestsPerChild a zero
    - definir o valor da MaxSpareThreads para o mesmo valor como MaxClients
- A configuração típica dos controles processo de thread no MPM Worker poderia parecer como se segue:
    - ServerLimit 16
    - StartServers 2
    - MaxClients 150
    - MinSpareThreads 25
    - MaxSpareThreads 75
    - ThreadsPerChild 25
- Enquanto o processo pai é normalmente iniciado como root sob o Unix, a fim de se vincular à porta 80, os processos filhos e threads são lançados pelo Apache como um usuário com menos privilégios. As diretivas de usuário e de grupo são usados para definir os privilégios dos processos filho do Apache. Os processos filho devem ser capazes de ler todo o conteúdo que vai ser servido, mas devem ter o mínimo de privilégios para mais, possível. Além disso, a menos que o suexec seja usado, essas diretivas também definem os privilégios que serão herdados por scripts CGI.
- O MaxRequestsPerChild controla a freqüência com que o servidor recicla processos por matar os antigos e lançando novos.

Implementação de hosts virtuais do Apache 2.x (com e sem endereços IPs dedicados)

- Virtual Host [15]
    - O termo Virtual Host (host virtual) refere-se à prática de executar mais de um web site (como company1.example.com e company2.example.com) em uma única máquina. As máquinas virtuais podem ser "baseadas em IP" (IP-based), o que significa que se tem um endereço IP diferente para cada site, ou "base de nome" (name-based), o que significa que se tem vários nomes em execução em cada endereço IP. O fato de que eles estão sendo executados no mesmo servidor físico não é evidente para o usuário final.
    - O Apache foi um dos primeiros servidores a suportar máquinas virtuais baseadas em IP. Versões 1.1 e posteriores do Apache suportam tanto hosts virtuais baseados em IP e nomes (VHosts). A última variante de hosts virtuais é por vezes também chamado hosts virtuais baseado em host (host-based) ou não-IP (non-IP).
    - Virtual Host baseado em Nome [16]
        - Para usar hospedagem virtual baseada em nome, deve-se designar o endereço IP (e talvez a porta) no servidor que estará aceitando requisições para os hosts. Isso é configurado usando a diretiva NameVirtualHost. No caso normal, onde deve ser usado todos e quaisquer endereços IP no servidor, pode-se usar * como o argumento para NameVirtualHost. Caso se estiver planejando usar múltiplas portas (por exemplo, executando SSL), deve-se adicionar uma porta para o argumento, como *:80. Note-se que mencionar um endereço IP em uma diretiva NameVirtualHost não faz automaticamente o servidor ouvir o endereço IP. Além disso, qualquer endereço de IP especificado aqui deve ser associado com uma interface de rede no servidor.
        - O próximo passo é criar um bloco <VirtualHost> para cada host diferente que se gostaria de servir. O argumento para a diretiva <VirtualHost> deve coincidir com

uma diretiva NameVirtualHost definida. (Nesse caso habitual, esse será "*:80"). Dentro de cada bloco <VirtualHost>, vai precisar de pelo menos uma diretiva ServerName para designar qual host é servido e uma diretiva DocumentRoot para mostrar onde no sistema de arquivos está o conteúdo para aquele host.

- Hospedeiro principal vai embora
  - Caso se está adicionando hosts virtuais para um servidor web existente, também se deve criar um bloco <VirtualHost> para o host existente. O ServerName e DocumentRoot incluídos neste hospedeiro virtual deve ser o mesmo que o ServerName e o DocumentRoot global. Liste esse host virtual pela primeira vez no arquivo de configuração que ele irá atuar como o host padrão.
- Por exemplo, suponha que se está servindo o domínio www.domain.tld e que se deseja adicionar o host virtual www.otherdomain.tld, o que aponta para o mesmo endereço IP. Em seguida, basta adicionar o seguinte para httpd.conf:
  - NameVirtualHost *:80
  - <VirtualHost *:80>
  - ServerName www.domain.tld
  - ServerAlias domain.tld *.domain.tld
  - DocumentRoot /www/domain
  - </VirtualHost>
  - <VirtualHost *:80>
  - ServerName www.otherdomain.tld
  - DocumentRoot /www/otherdomain
  - </VirtualHost>
- Alternativamente, pode-se especificar um endereço IP explícito no lugar do * em ambos as diretivas NameVirtualHost e <VirtualHost>. Por exemplo, pode-se querer fazer isso para executar alguns hosts virtuais baseados em nome de um endereço de IP, e quer baseada em IP, ou um outro conjunto à base de nomes de hosts virtuais em outro endereço.
- Muitos servidores querem ser acessíveis por mais de um nome. Isso é possível com a diretiva ServerAlias, colocada dentro da seção <VirtualHost>. Por exemplo, no primeiro bloco <VirtualHost> acima, a diretiva ServerAlias indica que os nomes listados são outros nomes que as pessoas podem usar para ver que mesmo web site:
  - ServerAlias domain.tld *.domain.tld
- então, requisições para todos os hosts no domínio domain.tld serão atendidos pelo host virtual www.domain.tld. Os caracteres curinga * e ? podem ser usados para combinar nomes. Claro, você não pode simplesmente inventar nomes e colocá-los em ServerName ou ServerAlias. Primeiro, você deve ter o seu servidor DNS configurado corretamente para mapear esses nomes para um endereço IP associado com o seu servidor.
- A lista completa de nomes da diretiva VirtualHost são tratados apenas como um (não wildcard) ServerAlias.
- Finalmente, pode-se ajustar a configuração das máquinas virtuais, colocando outras diretivas dentro de containers <VirtualHost>. A maioria das diretivas podem ser colocadas nesses containers e irá então mudar a configuração apenas da máquina virtual em questão. Para saber se uma diretriz em particular é permitida, verifica-se o contexto da diretiva. Diretivas de configuração definidas no contexto do servidor

principal (fora de qualquer container <VirtualHost>) serão usadas somente se não forem substituídas pelas configurações da máquina virtual.
- Agora, quando chega uma solicitação, o servidor irá verificar primeiro se ele estiver usando um endereço IP que corresponde a NameVirtualHost. Se for, então ele vai olhar para cada seção <VirtualHost> com um endereço IP correspondente e tentar encontrar uma em que o nome do servidor ou ServerAlias corresponde ao hostname solicitado. Se encontrar um, então ele usa a configuração para esse servidor. Se nenhum host virtual correspondente for encontrado, então o host virtual listado pela primeira vez que coincide com o endereço de IP será usado.
- Como conseqüência, o host virtual primeiro listado é o host virtual padrão. O DocumentRoot do servidor principal nunca será usado quando um endereço IP corresponde a diretiva NameVirtualHost. Caso se gostaria de ter uma configuração especial para os pedidos que não correspondem a qualquer host virtual particular, simplesmente coloque essa configuração em um recipiente <VirtualHost> e liste-o primeiro no arquivo de configuração.

- Virtual Host baseado por IP [17]
  - Como o termo baseado em IP indica, o servidor deve ter uma combinação endereço IP/porta diferente para cada máquina virtual baseada em IP. Isso pode ser conseguido pela máquina que tem várias conexões de rede física, ou pelo uso de interfaces virtuais que são suportados pela maioria dos sistemas operacionais modernos, e/ou o uso de vários números de porta.
  - Na terminologia do Apache HTTP Server, usando um único endereço IP, mas várias portas TCP, é também de hospedagem virtual baseada em IP.
  - Como configurar o Apache
    - Há duas maneiras de configurar o Apache para suportar múltiplos hosts. Seja pela execução de um daemon httpd separado para cada hostname, ou executando um único daemon que suporta todos os hosts virtuais.
    - Use vários daemons quando:
      - Há questões de particionamento de segurança, como company1 não quer que ninguém da company2 seja capaz de ler os seus dados, exceto através da web. Nesse caso, precisa-se de dois daemons, cada um rodando com diferente usuário, grupo e escuta, e as configurações de ServerRoot.
      - Pode-se arcar com os requisitos de memória e descritor de arquivo de ouvir todos os alias de IP na máquina. É possível apenas para ouvir o endereço de "curinga", ou para endereços específicos. Então, caso se tem uma necessidade de ouvir um endereço específico, por qualquer razão, então vai ser necessário ouvir todos os endereços específicos. (Embora um httpd poderia ouvir N-1 dos endereços, e outra poderia escutar o endereço restante.)
    - Use um único daemon quando:
      - Compartilhando configuração httpd entre hosts virtuais é aceitável.
      - A máquina destinada a um grande número de pedidos, e por isso a perda de desempenho em execuções separadas do daemon pode ser significativa.
  - Definição de vários daemons

- Criar uma instalação httpd separada para cada host virtual. Para cada instalação, use a diretiva Listen no arquivo de configuração para selecionar o endereço IP (ou host virtual) que o daemon serve. Por exemplo
  - Listen 192.168.0.1:80
- É recomendável que se use um endereço IP em vez de um nome de host.

■ Configurando um único daemon com hosts virtuais

- Para esse caso, um único httpd atenderá solicitações para o servidor principal e todos os hosts virtuais. A diretiva VirtualHost no arquivo de configuração é usada para definir os valores de ServerAdmin, ServerName, DocumentRoot, ErrorLog e diretivas de configuração TransferLog ou CustomLog para valores diferentes para cada host virtual. Por exemplo
  - <VirtualHost 192.168.0.1:80>
  - ServerAdmin webmaster@smallco.example.com
  - DocumentRoot /groups/smallco/www
  - ServerName smallco.example.com
  - ErrorLog /groups/smallco/logs/error_log
  - TransferLog /groups/smallco/logs/access_log
  - </VirtualHost>
  - <VirtualHost 192.168.0.2:80>
  - ServerAdmin webmaster@baygroup.example.org
  - DocumentRoot /groups/baygroup/www
  - ServerName baygroup.example.com
  - ErrorLog /groups/baygroup/logs/error_log
  - TransferLog /groups/baygroup/logs/access_log
  - </VirtualHost>
- É recomendável que se use um endereço IP em vez de um nome de host na diretiva <VirtualHost>.
- Endereços IP específicos ou portas têm precedência sobre os seus equivalentes curinga, e qualquer host virtual que corresponda tem precedência sobre a configuração base do servidor.
- Quase qualquer diretiva de configuração pode ser colocada na diretiva VirtualHost, com exceção das diretivas que controlam a criação do processo e algumas outras diretivas. Para saber se uma diretiva pode ser utilizado na diretiva VirtualHost, verifique o contexto usando o índice de diretiva.
- SuexecUserGroup pode ser usado dentro de uma diretiva VirtualHost se o wrapper suEXEC é usado.
- SEGURANÇA: Ao especificar onde gravar os arquivos de log, esteja ciente de alguns riscos de segurança estão presentes se alguém que não seja o usuário que inicia Apache tem acesso escrever para o diretório onde eles são escritos.

Usando redirecionamentos nos arquivos de configuração do Apache para personalizar o acesso a arquivos

- mod_rewrite [18]
  - Esse módulo utiliza um motor de reescrita baseado em regras (com base em um analisador de expressão regular) para reescrever URLs solicitados em execução. Ele suporta um número ilimitado de regras e um número ilimitado de condições de regras conectadas para cada regra, para fornecer um mecanismo de manipulação de URL muito flexível e poderoso. As manipulações de URL podem depender de vários testes, das variáveis de servidor,

variáveis de ambiente, cabeçalhos HTTP, ou carimbos de tempo. Mesmo pesquisas de bancos de dados externos em vários formatos podem ser utilizadas para conseguir uma correspondência de URL altamente granular.

- Esse módulo funciona com as URLs completas (incluindo a parte path-info), tanto em contexto por servidor (httpd.conf) e contexto por diretório (.htaccess) e pode gerar partes query-string no resultado. O resultado reescrito pode levar a subprocessamento interno, o redirecionamento de solicitação externa ou até mesmo a uma taxa de transferência de proxy interno.
- Variáveis de ambiente
  - Esse módulo mantém o controle de duas variáveis de ambiente adicionais (não-padrão) CGI/SSI nomeadas SCRIPT_URL e SCRIPT_URI. Essas contêm a visão web lógica do atual recurso, enquanto as variáveis do padrão CGI/SSI SCRIPT_NAME e SCRIPT_FILENAME contêm visão de sistema físico.
  - Aviso: Essas variáveis mantêm a URI/URL como elas foram inicialmente solicitadas, isto é, antes de qualquer reescrita. Isso é importante notar, porque o processo de escrita é utilizado principalmente para reescrever URLs para caminhos lógicos.
- Reescrita em hosts virtuais
  - Por padrão, as configurações do mod_rewrite do contexto do servidor principal não são herdadas por hosts virtuais. Para fazer as configurações do servidor principal se aplicarem ao hosts virtuais, deve-se colocar as seguintes diretrizes em cada seção <VirtualHost>:
    - RewriteEngine On
    - RewriteOptions Inherit
- Diretiva RewriteBase
  - A diretiva RewriteBase especifica o prefixo da URL para ser usado para diretivas RewriteRule por diretório (.htaccess) que substituem um caminho relativo.
  - É necessária essa diretiva quando usar um caminho relativo em uma substituição em contexto por diretório (.htaccess), a menos que uma das seguintes condições forem verdadeiras:
    - A requisição original, e a substituição, são por baixo do DocumentRoot (em oposição a acessível por outros meios, tais como alias).
    - O caminho do sistema de arquivos para o diretório que contém o RewriteRule, seguida pela substituição relativa também é válido como um caminho de URL no servidor (isso é raro).
  - No exemplo abaixo, o RewriteBase é necessário para evitar reescrever para http://example.com/opt/myapp-1.2.3/welcome.html uma vez que o recurso não foi em relação à raiz do documento. Essa configuração incorreta normalmente faria com que o servidor procure um diretório "opt" sob a raiz do documento.
  - DocumentRoot /var/www/example.com
  - Alias /myapp /opt/myapp-1.2.3
  - <Directory /opt/myapp-1.2.3>
  - RewriteEngine On
  - RewriteBase /myapp/
  - RewriteRule ^index\.html$ welcome.html
  - </Directory>
- Diretiva RewriteCond
  - A diretiva RewriteCond define uma condição de regra. Um ou mais RewriteCond pode preceder uma diretiva RewriteRule. A regra a seguir é então utilizada somente

se tanto o estado atual da URI corresponde ao seu padrão, e se essas condições forem atendidas.

- Exemplo
  - Para reescrever a homepage de um site de acordo com o cabeçalho “User-Agent:” do request, pode-se usar o seguinte:
    - RewriteCond %{HTTP_USER_AGENT} ^Mozilla
    - RewriteRule ^/$ /homepage.max.html [L]
    - RewriteCond %{HTTP_USER_AGENT} ^Lynx
    - RewriteRule ^/$ /homepage.min.html [L]
    - RewriteRule ^/$ /homepage.std.html [L]
  - Explicação: Caso se usar um navegador que se identifica como "Mozilla" (incluindo o Netscape Navigator, Mozilla etc), então se começa a homepage max (o que pode incluir frames, ou outras características especiais). Caso se usa o navegador Lynx (que funciona em um terminal), então se começa a homepage min (o que poderia ser uma versão projetada para navegação fácil, somente texto). Se nenhuma dessas condições se aplicam (se usa qualquer outro navegador, ou o navegador se identifica como algo não-padrão), se começa a std (standard) homepage.

- Diretiva RewriteRule
  - A diretiva RewriteRule é o carro-chefe da reescrita real. A diretiva pode ocorrer mais de uma vez, com cada instância que define uma única regra de reescrita. A ordem em que elas são definidas é importante - essa é a ordem em que eles serão aplicados em tempo de execução.
  - Padrão de correspondência é uma expressão regular compatível com perl. No primeiro RewriteRule é aplicado ao (%-decoded) caminho de URL da requisição; padrões subsequentes são aplicados à saída do último RewriteRule correspondente.
  - O que é correspondido?
    - No contexto VirtualHost, o padrão será inicialmente correspondido contra a parte da URL após o nome do host e porta, e antes da string de consulta (por exemplo, "/app1/index.html").
    - Em diretório e contexto htaccess, o padrão será inicialmente comparado com o caminho do sistema de arquivos, após a remoção do prefixo que levou o servidor para o RewriteRule atual (por exemplo, "app1/index.html" ou "index.html", dependendo de onde as diretivas são definidas).
    - Se você deseja corresponder contra a string de hostname, porta ou consulta, use um RewriteCond com as variáveis %{HTTP_HOST}, %{SERVER_PORT}, ou %{QUERY_STRING} respectivamente.
  - Reescritas por diretório
    - O motor de reescrita pode ser usados em arquivos .htaccess e em seções <Diretctory>, com alguma complexidade adicional.
    - Para ativar o mecanismo de reescrita, nesse contexto, é necessário definir "RewriteEngine On" e "Options FollowSymLinks" deve ser ativado. Se o administrador desativou o override de FollowSymLinks para o diretório de um usuário, então não se pode usar o mecanismo de reescrita. Essa restrição é necessária por razões de segurança.
    - Ao usar o mecanismo de reescrita em arquivos .htaccess o prefixo por diretório (que sempre é o mesmo para um diretório específico) é automaticamente removido para o padrão de correspondência do

RewriteRule e adicionado automaticamente após a substituição de qualquer parente (não começa com uma barra ou nome de protocolo) encontrar a extremidade de um conjunto de regras. Consulte a diretiva RewriteBase para mais informações sobre qual o prefixo será adicionado de volta para substituições relativos.

- Caso se deseja corresponder contra o caminho de URL completo em um RewriteRule por diretório (htaccess), use a variável %{REQUEST_URI} em um RewriteCond.
- O prefixo removido sempre termina com uma barra, ou seja, o correspondente ocorre contra uma string que nunca tem uma barra de liderança. Portanto, um padrão com ^/ nunca corresponde em contexto por diretório.
- Apesar das regras de reescrita são sintaticamente permitidas em seções <Location> e <Files>, isso nunca deve ser necessário e não é suportado.

- mod_alias [19]
  - As diretrizes contidas nesse módulo permitem a manipulação e controle de URLs como as requisições chegam ao servidor. As diretivas Alias e ScriptAlias são utilizados para mapear entre URLs e caminhos de arquivos. Isso permite que o conteúdo que não está diretamente sob a DocumentRoot sirva como parte da árvore de documentos web. A diretiva ScriptAlias tem o efeito adicional de marcar o diretório de destino como contendo apenas scripts CGI.
  - As diretivas de redirecionamento são usadas para instruir os clientes para fazer um novo pedido com uma URL diferente. Eles são freqüentemente usados quando um recurso foi movido para um novo local.
  - O mod_alias é projetado para lidar com tarefas simples manipulação de URL. Para tarefas mais complicadas, como manipular a string de consulta, use as ferramentas fornecidas pelo mod_rewrite.
  - Ordem de processamento
    - Aliases e redirecionamentos ocorrendo em diferentes contextos são processados como outras diretivas de acordo com as regras de mesclagem padrão. Mas quando vários Aliases ou redirecionamentos ocorrem no mesmo contexto (por exemplo, na mesma seção <VirtualHost>) são processados em uma ordem particular.
    - Em primeiro lugar, todos os redirecionamentos são processados antes que os Aliases e, portanto, um pedido que corresponde a um Redirect vs RedirectMatch nunca terá apelido aplicado. Em segundo lugar, os Aliases e redirecionamentos são processados na ordem em que aparecem nos arquivos de configuração, com a primeira correspondência tendo precedência.
    - Por essa razão, quando duas ou mais dessas diretivas aplicam ao mesmo subcaminho, deve-se listar o caminho mais específico em primeiro afim de que todas as diretivas tenham efeito. Por exemplo, a seguinte configuração funcionará como esperado:
      - Alias /foo/bar /baz
      - Alias /foo /gaq
    - Mas se as duas diretivas anteriores foram revertidas em ordem, o Alias /foo sempre corresponde antes do Alias /foo/bar, por isso a última diretiva seria ignorada.
  - Diretiva Redirect
    - A diretiva Redirect mapeia um URL antiga para um nova, pedindo o cliente a buscar novamente o recurso no novo local.

        - A velha URL é um caminho entre maiúsculas e minúsculas (%-decoded) começando com uma barra. Um caminho relativo não é permitido. A nova URL deve ser uma URL absoluta começando com um esquema e hostname. No Apache HTTP Server 2.2.6 e, superior, um caminho de URL que comece com uma barra também podem ser utilizados, caso em que o esquema e o hostname do servidor atual serão adicionados.
        - Então, qualquer pedido começando com URL-path irá retornar uma solicitação de redirecionamento para o cliente no local da URL de destino. Informações de caminho adicionais além da URL-path correspondida serão anexados à URL de destino.
        - Exemplo:
            - Redirect /service http://foo2.example.com/service
        - Se o cliente solicitar http://example.com/service/foo.txt, ele será solicitado a acessar http://foo2.example.com/service/foo.txt no lugar. Somente segmentos completos de caminhos são correspondidos, de modo que o exemplo acima não corresponderia a um pedido de http://example.com/servicefoo.txt. Para correspondência mais complexa usando expressões regulares, consulte a diretiva RedirectMatch.
        - Se nenhum argumento status é concedido, o redirecionamento será (status HTTP 302) "temporário". Isso indica ao cliente que o recurso foi movido temporariamente. O argumento de status pode ser usado para retornar outros códigos de status HTTP:
        - permanent
            - Retorna um status de redirecionamento permanente (301), indicando que o recurso foi movido permanentemente.
        - temp
            - Retorna um status de redirecionamento temporário (302). Este é o padrão.
        - seeother
            - Retorna um status "Veja outro" (303) indicando que o recurso tenha sido substituído.
        - gone
            - Retorna um status de "Gone" (410), indicando que o recurso foi removido permanentemente. Quando esse status é usado o argumento de URL deve ser omitida.
        - Outros códigos de status podem ser devolvidos ao dar o código de status numérico como o valor de status. Se o status for entre 300 e 399, o argumento de URL deve estar presente, caso contrário, ele deve ser omitida. Note-se que o estado deve ser conhecido ao código Apache.
        - Exemplo:
            - Redirect permanent /one http://example.com/two
            - Redirect 303 /three http://example.com/other
    - Diretiva RedirectMatch
        - Essa diretiva é equivalente a Redirect, mas faz uso de expressões regulares, em vez de simples correspondência de prefixo. A expressão regular fornecida é comparado com o caminho de URL, e se ela corresponde, o servidor irá substituir todas as correspondências entre parênteses na seqüência dada e usá-lo como um nome de arquivo. Por exemplo, para redirecionar todos os arquivos GIF para arquivos JPEG com o mesmo nome em outro servidor, pode-se usar:
            - RedirectMatch (.*)\.gif$ http://www.anotherserver.com$1.jpg
    - Diretiva RedirectPermanent

        - Essa diretiva faz com que o cliente saiba que o redirecionamento é permanente (status 301). Exatamente equivalente ao redirecionamento permanente.
    - Diretiva RedirectTemp
        - Essa diretiva faz com que o cliente saiba que o redirecionamento é apenas temporária (status 302). Exatamente equivalente a Redirect temporário.

Termos e utilitários

- logs de acesso e logs de erro
    - logs de acesso - referentes as requisições processadas pelo servidor web
    - logs de erro - referentes as exceções levantadas pelo servidor web
- .htaccess - [20] arquivo de configuração por diretório, suportado por vários servidores web
- httpd.conf - arquivo de configuração principal
- mod_auth - módulo de que permite autenticação básica de usuários até a versão 2.1 do Apache HTTP Server
- htpasswd - (1) gerencia arquivos de usuários para autenticação básica
- AuthUserFile, AuthGroupFile - diretivas de arquivos de autenticação para usuários e grupos
- apache2ctl(apachectl(8)) - Interface de Controle do Apache HTTP Server
- httpd - (8) Apache Hypertext Transfer Protocol Server

Referências

1. http://en.wikipedia.org/wiki/Apache_HTTP_Server
2. http://httpd.apache.org/docs/2.2/logs.html
3. http://httpd.apache.org/docs/2.2/howto/access.html
4. http://en.wikipedia.org/wiki/Mod_perl
5. https://perl.apache.org/
6. http://en.wikipedia.org/wiki/PHP
7. http://php.net/manual/en/install.unix.apache2.php
8. http://httpd.apache.org/docs/2.2/howto/auth.html
9. http://httpd.apache.org/docs/2.2/mod/mod_auth_basic.html
10. http://httpd.apache.org/docs/2.2/mod/mod_authn_file.html
11. http://httpd.apache.org/docs/2.2/mod/mod_authz_user.html
12. http://httpd.apache.org/docs/2.2/mpm.html
13. http://httpd.apache.org/docs/2.2/mod/prefork.html
14. http://httpd.apache.org/docs/2.2/mod/worker.html
15. http://httpd.apache.org/docs/2.2/vhosts/
16. http://httpd.apache.org/docs/2.2/vhosts/name-based.html
17. http://httpd.apache.org/docs/2.2/vhosts/ip-based.html
18. http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html
19. http://httpd.apache.org/docs/2.2/mod/mod_alias.html
20. http://en.wikipedia.org/wiki/.htaccess

Exercícios práticos

1. Preparação para exercícios
    a. Instalar os pacotes firefox, httpd, mod_perl e php (Ex.: yum install <pacote>)

2. Arquivos de configuração, termos e utilitários do Apache 2.x
    a. Listar o conteúdo do diretório /etc/httpd, de forma detalhada e recursiva (Ex.: ls <opções> <diretório>)

    b. Validar a configuração do servidor web através do binário httpd (Ex.: httpd <opções>)

    c. Validar a configuração do servidor web através do utilitário apachectl (Ex.: apachectl <opções>)

3. Arquivos de configuração de logs do Apache e conteúdo
    a. Iniciar o servidor web (Ex.: service <serviço> start)

    b. Exibir o conteúdo do arquivo de log de erro do servidor web (Ex.: cat <arquivo>)

    c. Definir o nível de detalhamento do log de erros para “debug”, no arquivo de configuração principal do servidor web (Ex.: vi <arquivo>)

    d. Reiniciar o servidor web (Ex.: service <serviço> restart)

    e. Exibir o conteúdo do arquivo de log de erro do servidor web (Ex.: cat <arquivo>)

    f. Definir o uso do arquivo de log “logs/agentes.txt” usando o formato “agent”, no arquivo de configuração principal do servidor web (Ex.: vi <arquivo>)

    g. Listar o diretório de logs do servidor web (Ex.: ls <diretório>)

    h. Reiniciar o servidor web (Ex.: service <serviço> restart)

    i. Exibir o conteúdo do arquivo de log “agentes.txt” do servidor web (Ex.: cat <arquivo>)

4. Métodos de restrição de acesso e arquivos
    a. Criar os diretórios /var/www/html/dir{1..3} (Ex.: mkdir <diretório>)

    b. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://127.0.0.1/dir1/

    c. Criar o arquivo vazio /etc/httpd/conf.d/teste.conf (Ex.: touch <arquivo>)

    d. No arquivo de configuração /etc/httpd/conf.d/teste.conf, negar permissão de acesso ao diretório /var/www/html/dir1/ para o endereço 127.0.0.1 (Ex.: vi <arquivo>)

    e. Reiniciar o servidor web (Ex.: service <serviço> restart)

    f. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://127.0.0.1/dir1/

g. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir2/

h. No arquivo de configuração /etc/httpd/conf.d/teste.conf, do servidor web, negar permissão de acesso para a localização “/dir2/”, para o endereço <IP do servidor> (Ex.: vi <arquivo>)

i. Reiniciar o servidor web (Ex.: service <serviço> restart)

j. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir2/

5. Configuração do mod_perl e PHP
   a. No arquivo de configuração do servidor web, do pacote mod_perl (/etc/httpd/conf.d/perl.conf), ativar o apelido para o diretório /var/www/perl e as opções do mesmo (Ex.: vi <arquivo>)

   b. Criar o diretório /var/www/perl (Ex.: mkdir <diretório>)

   c. Criar o arquivo /var/www/perl/teste.pl com o seguinte conteúdo (Ex.: vi <arquivo>)
      i. #!/usr/bin/perl
      ii. print "Content-type: text/plain\n\n";
      iii. print "mod_perl 2.0\n";

   d. Ativar o bit de execução do arquivo teste.pl (Ex.: chmod <opções>)

   e. Reiniciar o servidor web (Ex.: service <serviço> restart)

   f. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/perl/teste.pl

   g. Criar o arquivo /var/www/html/info.php com o seguinte conteúdo (Ex.: vi <arquivo>)
      i. <?php phpinfo(); ?>

   h. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/info.php

   i. Alterar os seguintes parâmetros do arquivo de configuração global do software PHP (Ex.: vi <arquivo>)
      i. Modo seguro (safe_mode): ligado
      ii. Expor PHP (expose_php): desligado
      iii. Uploads de arquivos (file_uploads): desligado

   j. Reiniciar o servidor web (Ex.: service <serviço> restart)

   k. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/info.php

6. Arquivos de configuração de autenticação de clientes e utilitários

a. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir3/

b. Criar o arquivo vazio /var/www/html/dir3/.htaccess (Ex.: touch <arquivo>)

c. No arquivo de configuração /var/www/html/dir3/.htaccess, definir (Ex.: vi <arquivo>)
   i. Tipo de autenticação (AuthType): Basic
   ii. Nome da autenticação (AuthName): “Acesso restrito”
   iii. Provedor de autenticação (AuthBasicProvider): file
   iv. Arquivo de autenticação usuários (AuthUserFile): /var/www/html/dir3/.htpasswd
   v. Requerer (Require): valid-user

d. Através do utilitário htpasswd, criar o arquivo /var/www/html/dir3/.htpasswd, definindo uma senha para o usuário admin (Ex.: htpasswd <opções>)

e. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir3/

f. Na seção de diretório /var/www/html, no arquivo de configuração principal do servidor web, alterar o valor do parâmetro AllowOverride para All (Ex.: vi <arquivo>)

g. Reiniciar o servidor web (Ex.: service <serviço> restart)

h. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir3/

7. Configuração do máximo de requisições, mínimo e máximo de servidores e clientes
   a. Verificar a quantidade de processos httpd que estão em execução (Ex.: ps <opções>)

   b. No arquivo de configuração principal do servidor web, aumentar o número inicial de servidores do MPM prefork para 30 (Ex.: vi <arquivo>)

   c. Reiniciar o servidor web (Ex.: service <serviço> restart)

   d. Verificar a quantidade de processos httpd que estão em execução (Ex.: ps <opções>)

   e. Aguardar 15 segundos e repetir o contagem (Ex.: ps <opções>)

   f. No arquivo de configuração principal do servidor web, aumentar o número máximo de servidores reservas do MPM prefork para 30 (Ex.: vi <arquivo>)

   g. Reiniciar o servidor web (Ex.: service <serviço> restart)

   h. Verificar a quantidade de processos httpd que estão em execução (Ex.: ps <opções>)

   i. Aguardar 15 segundos e repetir o contagem (Ex.: ps <opções>)

   j. Para o serviço httpd (Ex.: service <serviço> stop)

k. No arquivo de configuração do script SysVInit do serviço httpd (/etc/sysconfig/httpd), alterar o binário do serviço httpd para o MPM worker (Ex.: vi <arquivo>)

l. Renomear o arquivo de configuração do pacote do módulo php para /etc/httpd/conf.d/php.conf-noload

m. Iniciar o serviço httpd (Ex.: service <serviço> start)

n. No arquivo de configuração principal do servidor web, aumentar o número máximo de clientes do MPM woker para 1000 (Ex.: vi <arquivo>)

o. Reiniciar o servidor web (Ex.: service <serviço> restart)

p. No arquivo de configuração principal do servidor web, definir o número limite para servidor do MPM woker para 1000 (Ex.: vi <arquivo>)

q. Reiniciar o servidor web (Ex.: service <serviço> restart)

8. Implementação de hosts virtuais do Apache 2.x (com e sem endereços IPs dedicados)
   a. No arquivo de mapeamento de hosts estáticos (/etc/hosts), definir os endereços (Ex.: vi <arquivo>)
      i. exemplo.com - 127.0.0.1
      ii. teste.com.br - <IP do servidor>
      iii. dominio.com - <IP da segunda interface do servidor>

   b. Criar os diretórios /opt/virtualhosts/{exemplo.com,teste.com.br,dominio.com} (Ex.: mkdir <diretório>)

   c. Em cada diretório criado (/opt/virtualhosts/{exemplo.com,teste.com.br,dominio.com}), criar um arquivo index.html, contendo o nome do diretório (Ex.: vi <arquivo>)

   d. Fechar e abrir novamente o navegador firefox, na interface gráfica

   e. No arquivo de configuração /etc/httpd/conf.d/teste.conf, do servidor web, definir (Ex.: vi <arquivo>)
      i. Hosts virtuais por nome (NameVirtualHost): 127.0.0.1:80
      ii. Host virtual
         1. Endereço: 127.0.0.1:80
         2. Nome do servidor (ServerName): localhost
         3. Diretório de documento raiz (DocumentRoot): /var/www/html
      iii. Host virtual
         1. Endereço: 127.0.0.1:80
         2. Nome do servidor (ServerName): exemplo.com
         3. Diretório de documento raiz (DocumentRoot): /opt/virtualhosts/exemplo.com/
      iv. Host virtual
         1. Endereço: <IP do servidor>:80
         2. Nome do servidor (ServerName): teste.com.br

3. Diretório de documento raiz (DocumentRoot): /opt/virtualhosts/teste.com.br/

v. Host virtual
   1. Endereço: <IP da segunda interface do servidor>:80
   2. Nome do servidor (ServerName): dominio.com
   3. Diretório de documento raiz (DocumentRoot): /opt/virtualhosts/dominio.com/

f. Validar a configuração de hosts virtuais do servidor web (Ex.: apachectl <opções>)

g. Reiniciar o servidor web (Ex.: service <serviço> restart)

h. Através da interface gráfica, usar o navegador web firefox e acessar os endereços de hosts virtuais criados

9. Usando redirecionamentos nos arquivos de configuração do Apache para personalizar o acesso a arquivos
   a. No arquivo de configuração /etc/httpd/conf.d/teste.conf, do servidor web, criar o apelido /tmp/ apontando para o diretório /tmp, e negar permissão de acesso para a localização "/tmp/", para o endereço <IP do servidor> (Ex.: vi <arquivo>)

   b. Reiniciar o servidor web (Ex.: service <serviço> restart)

   c. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/tmp/

   d. No arquivo de configuração /etc/httpd/conf.d/teste.conf, do servidor web, criar um redirecionamento temporário de /dir1/ para /dir3/ (Ex.: vi <arquivo>)

   e. Reiniciar o servidor web (Ex.: service <serviço> restart)

   f. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir1/

   g. No arquivo de configuração /etc/httpd/conf.d/teste.conf, do servidor web, criar uma regra de reescrita de /dir2/ para /dir3/ (Ex.: vi <arquivo>)

   h. Reiniciar o servidor web (Ex.: service <serviço> restart)

   i. Através da interface gráfica, usar o navegador web firefox e acessar o endereço http://<IP do servidor>/dir2/

Simulado

1. … é o arquivo principal de configuração do software Apache HTTP Server.

2. O utilitário … é um frontend para o servidor Apache HTTP Server, que tem como binário principal o ….

3. O Apache HTTP Server suporta módulos dinâmicos (DSO - dynamic shared objects), que são bibliotecas externas carregadas durante a inicialização do servidor.
    a. V
    b. F

4. As diretivas … e … são usadas para informar o arquivo onde serão registrados os erros do servidor Apache HTTP Server, e o nível de detalhamento dos registros.

5. Através da diretiva … é possível criar formatos específicos de log que são usados pela diretiva … para definir em quais arquivos, esses formatos serão registrados.

6. Através do módulo … é possível se criar restrições de acesso a recursos baseados em endereço IP ou nome de hosts.

7. É possível combinar módulos e suas diretivas, para aumentar a flexibilidade no controle de acesso a recursos.
    a. V
    b. F

8. Após definir o carregamento de um módulo de interpretação de linguagem, como por exemplo mod_perl ou PHP, é necessário adicionar um manipulador apropriado para um determinado tipo de arquivo, para que o servidor web possa enviar o processamento do arquivo ao interpretador correto.
    a. V
    b. F

9. … é geralmente o arquivo de configuração do software PHP.

10. O mod_perl permite acesso a API completa do servidor Apache HTTP Server.
    a. V
    b. F

11. Através do arquivo … é possível se criar configuração de autenticação por diretório.

12. … é o utilitário de gerencia de arquivos de senha para autenticação do tipo básica.

13. O MPM padrão do servidor Apache HTTP Server no Linux é o … e ele pode ser substituído pelo MPM … que possibilita suporte multi-thread.

14. A diretiva … é responsável por limitar o número máximo de acessos simultâneos ao servidor web.

15. Através das diretivas …, … e … é possível definir quantos processos serão iniciados, o número mínimo e máximo de processos em reserva.

16. Através da diretiva …, é possível habilitar o suporte a hosts virtuais baseados por nome, para um determinado IP.

17. É possível misturar o uso de hosts virtuais por nome e por IP em uma instância do servidor web.

a. V
b. F

18. Quando se usa hosts virtuais, deve-se criar um host virtual padrão para responder as requisições que não sejam destinadas a hosts virtuais específicos.
    a. V
    b. F

19. Através de diretivas como …, … e …, do módulo mod_alias, é possível respectivamente criar um apelido para um diretório, redirecionar uma requisição por correspondência literal e redirecionar uma requisição por correspondência de expressão regular.

20. O módulo … é responsável por reescrever requisições, e as diretivas …, … e … são usadas respectivamente para criar uma sufixo de reescrita, definir condições reescrita e reescrever a requisição.

208.2 Configuração do Apache para HTTPS

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de configurar um servidor web para prover HTTPS.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários SSL
- Habilidade para gerar uma chave privada e um CSR para uma CA comercial
- Habilidade para gerar um certificado auto-assinado de uma CA privada
- Habilidade de instalar uma chave e um certificado
- Consciência de problemas com hosts virtuais e uso de SSL
- Problemas de segurança no uso do SSL

Termos e utilitários:

- arquivos de configuração do Apache2
- SSLEngine, SSLCertificateKeyFile, SSLCertificateFile, SSLCertificateChainFile
- SSLProtocol, SSLCipherSuite, ServerTokens, ServerSignature, TraceEnable
- /etc/ssl/, /etc/pki/
- SSLCACertificateFile, SSLCACertificatePath
- openssl, CA.pl

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários SSL

- SSL e TLS [1]
    - Transport Layer Security (TLS) e seu antecessor, o Secure Sockets Layer (SSL), são protocolos de criptografia projetados para fornecerem a segurança das comunicações através de uma rede de computadores. Eles usam certificados X.509 e, por tanto, criptografia assimétrica, para autenticar a contraparte com quem estão se comunicando, e para negociar uma chave simétrica. Essa chave de sessão é então usada para criptografar os dados que fluem entre as partes. Isso permite a confidencialidade dos dados/mensagem, e códigos de autenticação de mensagens para integridade de mensagem e como um subproduto, autenticação de mensagens. Diversas versões dos protocolos estão em uso generalizado em aplicações como navegação na web, correio eletrônico, fax Internet, mensagens instantâneas e voz sobre IP (VoIP). Uma propriedade importante nesse contexto é o sigilo para frente (forward secrecy), de modo que a chave de sessão de curta duração não possa ser derivada a partir da chave secreta assimétrica de longo prazo.
    - Como conseqüência da escolha de certificados X.509, autoridades de certificação e uma infra-estrutura de chaves públicas são necessárias para verificar a relação entre um certificado e seu proprietário, bem como para gerar, assinar, e administrar a validade dos certificados. Enquanto isso pode ser mais benéfico do que a verificação das identidades através de uma rede de confiança, as divulgações de vigilância em massa de 2013 tornou mais amplamente conhecido que as autoridades de certificação são um ponto fraco do ponto de vista de segurança, permitindo ataques man-in-the-middle (MITM).

- No Internet Protocol Suite, TLS e SSL criptografam os dados de conexões de rede na camada de aplicação. Em equivalências do modelo OSI, o TLS/SSL é inicializado na camada 5 (camada de sessão) e trabalha na camada 6 (a camada de apresentação). A camada de sessão tem um aperto de mão (handshake) usando uma cifra assimétrica, a fim de estabelecer as configurações de criptografia e uma chave compartilhada para a sessão; em seguida, a camada de apresentação encripta o resto da comunicação utilizando um código simétrico e aquela chave de sessão. Em ambos os modelos, o TLS e SSL trabalham em nome da camada de transporte subjacente, cujos segmentos transportam dados criptografados.
- O TLS é um protocolo de normas da Internet Engineering Task Force (IETF), definido pela primeira vez em 1999 e atualizado na RFC 5246 (Agosto de 2008) e RFC 6176 (Março de 2011). É baseado nas especificações SSL anteriores (1994, 1995, 1996), desenvolvido pela Netscape Communications para adicionar o protocolo HTTPS para o seu navegador Navigator.
- Histórico e desenvolvimento
  - Programação de Rede Segura
    - Os primeiros esforços de pesquisa para a segurança da camada de transporte incluíram a interface de programação de aplicações (API) Secure Network Programming (SNP), que, em 1993, explorou a abordagem de ter uma API de camada de transporte segura muito parecida com os soquetes Berkeley, para facilitar a adaptação de aplicações de rede pré-existentes com medidas de segurança.
  - SSL 1.0, 2.0 e 3.0
    - A Netscape desenvolveu os protocolos SSL originais versão 1.0 que nunca foram lançados publicamente por causa de graves falhas de segurança no protocolo; a versão 2.0, lançada em Fevereiro de 1995, contendo uma série de falhas de segurança que levou à concepção do SSL Versão 3.0. O SSL versão 3.0, lançado em 1996, representou uma reformulação completa do protocolo, produzida por Paul Kocher trabalhando com os engenheiros da Netscape Phil Karlton e Alan Freier, com uma implementação de referência por Christopher Allen e Tim Dierks da Consensus Development. Versões mais recentes do SSL/TLS são baseadas no SSL 3.0. O projeto de 1996 do SSL 3.0 foi publicado pelo IETF como um documento histórico na RFC 6101.
    - O Dr. Taher Elgamal, cientista chefe da Netscape Communications de 1995-1998, é reconhecido como o "pai do SSL".
    - A partir de 2014, a versão 3.0 do SSL é considerada insegura, pois é vulnerável ao ataque POODLE que afeta todas as cifras de bloco no SSL; e o RC4, a única cifra não-bloco suportada pelo SSL 3.0, também é viável de ser quebrada quando usada no SSL 3.0.
  - TLS 1.0
    - O TLS 1.0 foi definido pela primeira vez na RFC 2246 em Janeiro de 1999 como uma atualização do SSL versão 3.0, e escrito por Christopher Allen e Tim Dierks do Consensus Development. Como se afirma na RFC, "as diferenças entre esse protocolo e o SSL 3.0 não são dramáticas, mas são significativas o suficiente para impedir a interoperabilidade entre o TLS 1.0 e o SSL 3.0". O TLS 1.0 inclui um meio pelo qual a implementação TLS pode degradar a conexão para o SSL 3.0, enfraquecendo assim a segurança.
  - TLS 1.1

- O TLS 1.1 foi definido na RFC 4346, em Abril de 2006. Trata-se de uma atualização da TLS versão 1.0. As diferenças significativas nessa versão incluem:
  - Proteção adicional contra ataques de encadeamento de criptografia de bloco (chiper-block chaining - CBC).
    - O vetor de inicialização implícito (IV) foi substituído com um IV explícito.
    - Alteração na manipulação de erros de preenchimento.
  - Suporte para registro IANA de parâmetros.

- TLS 1.2
  - O TLS 1.2 foi definido na RFC 5246, em Agosto de 2008. Ele é baseado na especificação anterior TLS 1.1. As principais diferenças incluem:
    - A combinação MD5-SHA-1 na função pseudo-aleatória (PRF) foi substituída com SHA-256, com a opção de usar conjunto de cifras PRFs especificadas.
    - A combinação MD5-SHA-1 no hash de mensagem acabada foi substituída com o SHA-256, com a opção para usar algoritmos de hash específicos de conjunto de codificação. No entanto, o tamanho do hash na mensagem acabada ainda é truncado para 96 bits.
    - A combinação MD5-SHA-1 no elemento assinado digitalmente foi substituída por um único de hash negociado durante o aperto de mão, cujo padrão é SHA-1.
    - Aprimoramento na capacidade do cliente e servidor para especificar quais algoritmos de hash e assinatura eles vão aceitar.
    - Expansão de suporte para cifras de criptografia autenticadas, usadas principalmente para o Mode/Galois Counter (GCM) e o modo CCM de criptografia do Padrão Avançado de Criptografia.
    - Foram adicionados definição de extensões TLS e conjuntos de cifras do Padrão Avançado de Criptografia.
  - Todas as versões TLS foram refinadas na RFC 6176 de Março de 2011, removendo sua compatibilidade com SSL tal que sessões TLS nunca vão negociar o uso do Secure Sockets Layer (SSL) versão 2.0.

- TLS 1.3 (draft)
  - Em Abril de 2015, TLS 1.3 é um projeto, e os detalhes ainda não foram estabelecidos. Ele é baseado na especificação anterior do TLS 1.2. As principais diferenças em relação a TLS 1.2 incluem:
    - Suporte para apertos de mão 1-RTT.
    - Descontinuidade do suporte para muitos recursos inseguros ou obsoletos, incluindo, compressão, renegociação, cifras não-AEAD, RSA estático e troca de chaves DH, grupos de DHE customizados, formato de ponto de negociação, protocolo Change Cipher Spec, tempo UNIX de mensagem Hello, e o tamanho do comprimento de entrada AD para cifras AEAD.
    - Proibição de negociação SSL ou RC4 para compatibilidade com versões anteriores.
    - O uso integrado de hash de sessão.
    - O número da versão da camada registro foi congelado e depreciado para uma melhor compatibilidade com versões anteriores.

- OpenSSL [2]
    - Em redes de computadores, OpenSSL é uma implementação opensource dos protocolos SSL e TLS. A biblioteca central, escrita na linguagem de programação C, implementa funções criptográficas básicas e oferece várias funções utilitárias. Envoltórios (wrappers) que permitem a utilização da biblioteca OpenSSL em uma variedade de linguagens de computador estão disponíveis.
    - As versões estão disponíveis para a maioria dos sistemas operacionais Unix-like (incluindo Solaris, Linux, Mac OS X e os vários sistemas operacionais BSD open-source), OpenVMS e Microsoft Windows. A IBM fornece uma portabilidade para o System i (OS / 400).
    - O OpenSSL é baseado no SSLeay por Eric Andrew Young e Tim Hudson, desenvolvimento que não oficialmente encerrou em 17 de Dezembro de 1998, quando Young e Hudson ambos começaram a trabalhar para a RSA Security.
    - Histórico do projeto
        - O projeto OpenSSL foi fundado em 1998 para inventar um conjunto de ferramentas de criptografia livre para o código usado na Internet. A partir de 2014, dois terços de todos os servidores web o usam. A equipe de gerenciamento de projetos OpenSSL consiste em quatro europeus. O grupo de desenvolvimento inteiro é composto por 11 membros, dos quais 10 são voluntários; há apenas um funcionário em tempo integral, Stephen Henson, o principal desenvolvedor.
        - O projeto tem um orçamento de menos de US $1 milhão por ano e se baseia em parte em doações. Steve Marquess, um ex-consultor militar em Maryland começou a fundação para doações e contratos de consultoria e recebeu o patrocínio do Departamento de Segurança Interna do Estados Unidos e do Departamento de Defesa dos Estados Unidos.
    - Algorítimos
        - Cifragem
            - AES, Blowfish, Camellia, SEED, CAST-128, DES, IDEA, RC2, RC4, RC5, Triple DES, GOST 28147-89
        - Funções de hash criptográficas
            - MD5, MD4, MD2, SHA-1, SHA-2, RIPEMD-160, MDC-2, GOST R 34.11-94
        - Criptografia de chave pública
            - RSA, DSA, Diffie–Hellman key exchange, Elliptic curve, GOST R 34.10-2001
    - Arquivos de configuração do OpenSSL
        - /etc/ssl - diretório para certificados ssl
        - /etc/pki - diretório para certificados e utilitários usados em uma pki
        - /etc/pki/tls/openssl.cnf(config(8)) - arquivos de configuração da biblioteca OpenSSL CONF
    - Utilitários
        - openssl(1) - ferramenta de linha de comando OpenSSL
        - /etc/pki/tls/misc/CA.pl(1) - interface amigável para programas de certificados OpenSSL
        - /etc/pki/tls/misc/c_hash - imprime os valores de hash
        - /etc/pki/tls/misc/c_info - imprime o sujeito
        - /etc/pki/tls/misc/c_issuer - imprime o emissor
        - /etc/pki/tls/misc/c_name - imprime o sujeito

Habilidade para gerar uma chave privada e um CSR para uma CA comercial

- openssl genrsa

    - Exemplos:
        - openssl genrsa -new -o <arquivo> <tamanho> - cria um par de chaves RSA com o tamanho especificado, sem proteção criptográfica
        - openssl genrsa -new -des3 -o <arquivo> <tamanho> - cria um par de chaves RSA com o tamanho especificado, com proteção criptográfica DES3
- openssl rsa
    - Exemplos:
        - openssl rsa -in <chave> -noout -text - exibe informações sobre a chave especificada
        - openssl rsa -in <chave> -out <arquivo> - escreve um novo arquivo com a chave especificada, sem proteção criptográfica (usado para retirar a senha da chave)
- openssl gendsa
    - Exemplos:
        - openssl gendsa -new -o <arquivo> <tamanho> - cria um par de chaves DSA com o tamanho especificado, sem proteção criptográfica
        - openssl gendsa -new -des3 -o <arquivo> <tamanho> - cria um par de chaves DSA com o tamanho especificado, com proteção criptográfica DES3
- openssl dsa
    - Exemplos:
        - openssl dsa -in <chave> -noout -text - exibe informações sobre a chave especificada
        - openssl dsa -in <chave> -out <arquivo> - escreve um novo arquivo com a chave especificada, sem proteção criptográfica (usado para retirar a senha da chave)
- openssl req
    - Exemplos:
        - openssl req -new -out <arquivo> - gera uma requisição de assinatura de certificado, criando um novo par de chaves
        - openssl req -new -newkey rsa:<tamanho> -out <arquivo> - gera uma requisição de assinatura de certificado, criando um novo par de chaves RSA com o tamanho especificado
        - openssl req -new -newkey dsa:<config> -out <arquivo> - gera uma requisição de assinatura de certificado, criando um novo par de chaves DSA, usando o arquivo de configuração especificado para os parâmetros da chave
        - openssl req -new -key <chave> -out <arquivo> - gera uma requisição de assinatura de certificado, usando a chave especificada
        - openssl req -in <requisição> -noout -text - exibe informações sobre a requisição especificada
- CA.pl
    - Exemplos:
        - CA.pl -newreq - cria uma requisição de certificado
        - CA.pl -newreq-nodes - como o -newreq exceto que a chave privada não será criptografada

Habilidade para gerar um certificado auto-assinado de uma CA privada

- openssl x509
    - Exemplos:
        - openssl x509 -req <requisição> -signkey <chave> -days <dias> -out <arquivo> - assina a requisição especificada (formato auto-assinado) com a chave especificada
        - openssl x509 -in <certificado> -noout -text - exibe infromações do certificado especificado

- CA.pl
    - Exemplo: CA.pl -newcert - cria um certificado auto-assinado

Habilidade de instalar uma chave e um certificado

- mod_ssl [3]
    - Esse módulo fornece suporte SSL v2/v3 e TLS v1 para o servidor Apache HTTP. Ele foi contribuído por Ralf S. Engeschall baseado em seu projeto mod_ssl e originalmente derivado do trabalho por Ben Laurie.
    - Esse módulo depende do OpenSSL para fornecer o mecanismo de criptografia.
    - Diretiva SSLEngine
        - Essa diretiva alterna o uso do Engine do Protocolo SSL/TLS. Isso deve ser usado dentro de uma seção <VirtualHost> para ativar o SSL/TLS para um host virtual. Por padrão, o Engine do Protocolo SSL/TLS está desativado para o servidor principal e todos os hosts virtuais configurados.
    - Diretiva SSLCertificateKeyFile
        - Essa diretiva aponta para o arquivo de chave privada codificado em PEM, para o servidor. Se a chave privada não está combinada com o certificado no SSLCertificateFile, use essa diretiva adicional apontar para o arquivo com o stand-alone de chave privada. Quando a diretiva SSLCertificateFile é usada e o arquivo contém o certificado e a chave privada, essa diretiva não precisa ser usada. Mas desencoraja-se fortemente essa prática. Em vez disso, recomenda-se que se separe o certificado e a chave privada. Se a chave privada contida é criptografada, o diálogo “Frase de Senha” é forçado no momento do inicialização. Essa diretiva pode ser usada até três vezes (referenciando nomes de arquivos diferentes), quando ambos as chaves RSA, DSA e uma chave privada baseada ECC são utilizadas em paralelo.
    - Diretiva SSLCertificateFile
        - Essa diretiva aponta para o arquivo do certificado codificado em PEM, para o servidor e, opcionalmente, também para o arquivo correspondente da chave privada RSA ou DSA para ele (contido no mesmo arquivo). Se a chave privada contida é criptografada o diálogo “Frase de Senha” é forçado no momento da inicialização. Essa diretiva pode ser usada até três vezes (referenciando nomes de arquivos diferentes), quando ambas as chaves RSA e DSA, e um certificado de servidor baseado ECC são utilizados em paralelo.
    - Diretiva SSLCertificateChainFile
        - Essa diretiva define o arquivo opcional tudo-em-um, onde pode-se montar os certificados de Autoridades de Certificação (CA) que formam a cadeia de certificados do certificado do servidor. Isso começa com o certificado da CA de emissão do certificado do servidor e pode variar até o certificado da CA raiz. Esse arquivo é simplesmente a concatenação dos diversos arquivos de certificados CA codificados em PEM, geralmente na ordem da cadeia de certificação.
        - Isso deve ser utilizado, em alternativa e/ou adicionalmente para a diretiva SSLCACertificatePath para construir explicitamente a cadeia de certificado do servidor que é enviado para o navegador, além do certificado de servidor. É especialmente útil para evitar conflitos com os certificados da CA ao usar a autenticação do cliente. Porque embora a colocação de um certificado CA da cadeia de certificado do servidor no SSLCACertificatePath, tem o mesmo efeito para a construção da cadeia de certificados, esse tem o efeito colateral dos certificados de,

cliente emitidos por esse mesmo certificado CA, também são aceitos na autenticação do cliente.
        - Mas cuidado: fornecer a cadeia de certificado funciona apenas somente se estiver usando um único certificado de servidor baseado em RSA ou DSA. Se estiver usando um par de certificado RSA + DSA acoplados, isso vai funcionar somente se, na verdade, ambos os certificados usam a mesma cadeia de certificados. Senão os navegadores irão ser confundir nessa situação.
    - Diretiva SSLCACertificateFile
        - Essa diretiva define o arquivo tudo-em-um, onde pode-se montar os Certificados de Autoridades de Certificação (CA) cujos clientes são lidados. Esses são usados para autenticação do cliente. Esse arquivo é simplesmente a concatenação dos diversos arquivos de certificado codificado em PEM, em ordem de preferência. Isso pode ser utilizado em alternativa e/ou adicionalmente para a diretiva SSLCACertificatePath.
    - Diretiva SSLCACertificatePath
        - Essa diretiva define o diretório onde é mantido os certificados de Autoridades de Certificação (CAs) cujos clientes são lidados. Esses são utilizados para verificar o certificado de cliente em Autenticação do cliente.
        - Os arquivos nesse diretório tem que ser codificados em PEM e são acessados através nomes de arquivos de hash. Por isso, normalmente não se pode simplesmente colocar os arquivos de certificados lá: também tem que se criar links simbólicos chamados de hash-value.N. E deve-se sempre certificar-se que esse diretório contém os links simbólicos apropriados.
    - Diretiva SSLProtocol
        - Essa diretiva pode ser usada para controlar os sabores de protocolo SSL que o mod_ssl deve usar na elaboração do ambiente de servidor. Os clientes então só podem se conectar com um dos protocolos fornecidos.
        - Os protocolos (maiúsculas e minúsculas) disponíveis são:
            - SSLv2 - esse é o protocolo Secure Sockets Layer (SSL) versão 2.0 . É o protocolo SSL original como projetado pela Netscape Corporation. Embora seu uso foi depreciado, por causa de deficiências na segurança do protocolo.
            - SSLv3 - esse é o protocolo Secure Sockets Layer (SSL) versão 3.0, da Netscape Corporation. É o sucessor do SSLv2 e o antecessor de TLSv1. Ele é suportado por quase todos os browsers populares.
            - TLSv1 - esse é o protocolo Transport Layer Security (TLS) versão 1.0 . É o sucessor do SSLv3 e é definido na RFC 2246.
            - TLSv1.1 (quando se utiliza o OpenSSL 1.0.1 e posterior) - a revisão do protocolo TLS 1.0, conforme definido na RFC 4346.
            - TLSv1.2 (quando se utiliza o OpenSSL 1.0.1 e posterior) - a revisão do protocolo TLS 1.1, conforme definido na RFC 5246.
            - All - esse é um atalho para “+SSLv2 +SSLv3 +TLSv1 " ou - quando se utiliza OpenSSL 1.0.1 e posterior - “+SSLv2 +SSLv3 +TLSv1 +TLSv1.1 +TLSv1.2 ", respectivamente.
    - Diretiva SSLCipherSuite
        - Essa diretiva complexa utiliza uma string cifra-spec separada por dois pontos que consiste em especificações de criptografia OpenSSL para configurar o conjunto de codificação que o cliente está autorizado a negociar na fase handshake SSL. Note-se que essa diretiva pode ser usada tanto por contexto de servidor e por diretório. No contexto por servidor se aplica ao handshake SSL padrão quando uma conexão

é estabelecida. No contexto por diretório ele força uma renegociação SSL com o reconfigurado Cipher Suíte após a solicitação HTTP ser lida, mas antes que a resposta HTTP é enviada.
        - Uma especificação de criptografia SSL em cifra-spec é composta por 4 principais atributos mais alguns extras menores:
            - Algorítimo de troca de chaves: variantes RSA or Diffie-Hellman.
            - Algorítimo de autenticação: RSA, Diffie-Hellman, DSS ou nenhum.
            - Algorítimo de cifragem/criptografia: DES, Triple-DES, RC4, RC2, IDEA ou nenhum.
            - Algorítimo MAC Digest: MD5, SHA ou SHA1.
        - Uma cifra SSL pode ser uma cifra de exportação, e é também uma cifra SSLv2 ou SSLv3/TLSv1 (aqui TLSv1 é equivalente a SSLv3). Para especificar quais as cifras para usar, pode-se especificar todas as cifras, um de cada vez, ou usar pseudônimos para especificar a preferência e ordem para as cifras
- Outras diretivas [5]
    - Diretiva ServerTokens
        - Essa diretiva controla se o campo de cabeçalho de resposta do servidor que é enviado de volta para clientes inclui uma descrição do tipo de sistema operacional genérico do servidor, bem como informações sobre os módulos compilados.
            - ServerTokens Prod[uctOnly]
                - Servidor envia (por exemplo): Servidor: Apache
            - ServerTokens Major
                - Servidor envia (por exemplo): Servidor: Apache / 2
            - ServerTokens Minor
                - Servidor envia (por exemplo): Servidor: Apache / 2.0
            - ServerTokens Min[imal]
                - Servidor envia (por exemplo): Servidor: Apache / 2.0.41
            - ServerTokens OS
                - Servidor envia (por exemplo): Servidor: Apache / 2.0.41 (Unix)
            - ServerTokens Full (ou não especificado)
                - Servidor envia (por exemplo): Servidor: Apache / 2.0.41 (Unix) PHP/4.2.2 MyMod/1.2
        - Essa definição aplica-se a todo o servidor, e não pode ser ativada ou desativada por host virtual.
        - Depois da versão 2.0.44, essa diretiva também controla as informações apresentadas pela diretiva ServerSignature.
    - Diretiva ServerSignature
        - A diretiva ServerSignature permite a configuração de uma linha de rodapé sob documentos gerados pelo servidor (mensagens de erro, listagens de diretórios ftp mod_proxy, saída mod_info, ...). A razão por que se quer para permitir tal linha de rodapé é que, em uma cadeia de proxies, o usuário muitas vezes não tem a possibilidade de dizer qual dos servidores encadeados realmente produziu uma mensagem de erro retornada.
        - A definição Off, que é a padrão, suprime a linha de rodapé (e é, portanto, compatível com o comportamento de Apache-1.2 e anterior). A configuração On simplesmente adiciona uma linha com o número da versão do servidor e ServerName do host virtual que serve, e a configuração EMail adicionalmente cria uma referência "mailto:" ao ServerAdmin do referido documento.

- Diretiva TraceEnable
  - Essa diretiva substitui o comportamento de TRACE, para ambos o servidor núcleo e o mod_proxy. O padrão TraceEnable On permite requisições de rastreio pela RFC 2616, que proíbe qualquer corpo da solicitação acompanhar o pedido. O TraceEnable Off faz com que o servidor núcleo e o mod_proxy retornem um erro 405 (Método não permitido) para o cliente.
  - Finalmente, apenas para fins de testes e diagnóstico, os corpos das solicitações podem ser autorizados usando a diretiva TraceEnable extendida, em não conformidade. O núcleo (como um servidor de origem) irá restringir o corpo da solicitação para 64k (mais 8k para pedaço de cabeçalhos se o Transfer-Encoding: chunked é usado). O núcleo vai refletir os cabeçalhos completos e todos os pedaços de cabeçalhos com o corpo da resposta. Como um servidor proxy, o corpo do solicitação não é restrito a 64k.

Consciência de problemas com hosts virtuais e uso de SSL

- Host virtuais baseados em nome [6]
  - Como regra, é impossível hospedar mais de um host virtual SSL no mesmo endereço IP e porta. Isso porque o Apache precisa saber o nome do host, a fim de escolher o certificado correto para configurar a camada de criptografia. Mas o nome do host que está sendo requerido consta apenas nos cabeçalhos de solicitação HTTP, que fazem parte do conteúdo criptografado. Portanto, não está disponível até depois da criptografia já estar negociada. Isso significa que o certificado correto não pode ser selecionado, e os clientes receberão avisos de incompatibilidade de certificado e ser vulneráveis a ataques man-in-the-middle.
  - Na realidade, o Apache irá permitir que se configure hosts virtuais SSL baseados em nome, mas ele sempre usará a configuração do primeiro host virtual listado (no endereço IP e porta selecionado) para configurar a camada de criptografia. Em determinadas circunstâncias, é aceitável usar uma configuração SSL única para vários hosts virtuais. Em particular, isso vai funcionar se o certificado SSL se aplica a todos os hosts virtuais.

Problemas de segurança no uso do SSL

- Segurança [1]
  - SSL 2.0
    - O SSL 2.0 é falho em uma variedade de formas:
      - Chaves criptográficas idênticas são usadas para autenticação de mensagens e criptografia.
      - O SSL 2.0 tem uma construção MAC fraca que usa a função hash MD5 com um prefixo secreto, tornando-o vulnerável a ataques de extensão comprimento.
      - O SSL 2.0 não tem qualquer proteção para o aperto de mão, o que significa um ataque de rebaixamento man-in-the-middle pode passar despercebido.
      - O SSL 2.0 usa o fim de conexão TCP indicar o fim dos dados. Isso significa que os ataques de truncamento são possíveis: o atacante simplesmente forja um TCP FIN, deixando o destinatário sem consciência de um fim ilegítimo de mensagem de dados (o SSL 3.0 corrige esse problema por ter um alerta de encerramento explícito).
      - O SSL 2.0 assume um serviço único e um certificado de domínio fixo, que se choca com o recurso padrão de hospedagem virtual em servidores Web. Isso

significa que a maioria dos sites estão praticamente impossibilitados de usar SSL.
        - O SSL 2.0 é desabilitado por padrão, começando com o Internet Explorer 7, Mozilla Firefox 2, Opera 9.5 e Safari. Depois que ele envia uma TLS "ClientHello", se o Mozilla Firefox considera que o servidor é incapaz de completar o aperto de mão, ele tentará voltará a usar SSL 3.0 com um SSL 3.0 "ClientHello" em SSL formato 2.0 para maximizar a probabilidade de sucesso aperto de mão com servidores mais antigos. O suporte para SSL 2.0 (e cifras fracas de 40 bits e de 56 bits) foi removido completamente do Opera a partir da versão 10.
    - SSL 3.0
        - O SSL 3.0 foi melhorado do SSL 2.0, adicionando cifras e suporte baseados em SHA-1 para a autenticação de certificado.
        - Do ponto de vista da segurança, o SSL 3.0 deve ser considerado menos desejável do que TLS 1.0. Os conjuntos de cifra do SSL 3.0 têm um processo de derivação de chave mais fraca; metade da chave mestra que é estabelecida é totalmente dependente da função hash MD5, o que não é resistente às colisões e é, por consequência, não considerado seguro. Sob o TLS 1.0, a chave mestra que é estabelecida depende em ambos MD5 e SHA-1 para que o seu processo de derivação não seja atualmente considerado fraco. É por esta razão que implementações do SSL 3.0 não podem ser validadas sob o FIPS 140-2.
        - Em Outubro de 2014, a vulnerabilidade no projeto de SSL 3.0 foi relatada, o que torna o modo de operação CBC com SSL 3.0 vulnerável ao ataque de preenchimento (veja ataque POODLE).
    - TLS
        - TLS tem uma variedade de medidas de segurança:
            - Proteção contra um rebaixamento do protocolo para uma versão anterior (menos seguro) ou um conjunto de codificação mais fraca.
            - Numerando registros registros de aplicação subsequentes com um número de seqüência e usando esse número de seqüência nos códigos de autenticação de mensagens (MACs).
            - Usando uma síntese da mensagem reforçada com uma chave (para que apenas um porta-chaves possa verificar o MAC). A construção HMAC usada pela maioria das suites de cifra TLS é especificada no RFC 2104 (o SSL 3.0 usava um diferente hash baseado em MAC).
            - A mensagem que termina o aperto de mão ("Fininshed") envia um hash de todas as mensagens trocadas no aperto de mão visto por ambas as partes.
            - A função pseudo-aleatório divide os dados de entrada em metade e processa cada um com um diferente algoritmo de hashing (MD5 e SHA-1), em seguida, aplica XOR para criar o MAC. Isso fornece proteção, mesmo que um desses algoritmos é encontrado sendo vulnerável.

Termos e utilitários

- Arquivos de configuração do Apache2
    - ssl.conf - arquivo de configuração do mod_ssl
- /etc/ssl/, /etc/pki/
    - /etc/ssl - diretório para certificados ssl
    - /etc/pki - diretório para certificados e utilitários usados em uma pki

- openssl, CA.pl
  - openssl - (1) ferramenta de linha de comando OpenSSL
  - CA.pl - (1) interface amigável para programas de certificados OpenSSL
- SSLEngine, SSLCertificateKeyFile, SSLCertificateFile, SSLCertificateChainFile
- SSLCACertificateFile, SSLCACertificatePath
- SSLProtocol, SSLCipherSuite, ServerTokens, ServerSignature, TraceEnable

Referências

1. http://en.wikipedia.org/wiki/Transport_Layer_Security
2. http://en.wikipedia.org/wiki/OpenSSL
3. http://httpd.apache.org/docs/2.2/mod/mod_ssl.html
4. http://en.wikipedia.org/wiki/X.509#Certificate_filename_extensions
5. http://httpd.apache.org/docs/2.2/mod/core.html
6. https://wiki.apache.org/httpd/NameBasedSSLVHosts

Exercícios práticos

1. Preparação para exercícios
   a. Instalar os pacotes openssl-perl e mod_ssl (Ex.: yum install <pacote>)

2. Habilidade para gerar uma chave privada e um CSR para uma CA comercial
   a. Através do utilitário openssl, gerar um arquivo de chaves rsa, de tamanho 2048, com criptografia DES3, com o nome “temp.key” (Ex.: openssl genrsa <opções>)

   b. Exibir o conteúdo do arquivo temp.key (Ex.: cat <arquivo>)

   c. Verificar o arquivo de chave gerado pelo utilitário openssl, exibindo as informações como texto (Ex.: openssl rsa <opções>)

   d. Através do utilitário openssl, gerar um arquivo de requisição de assinatura, usando o arquivo “temp.key” como chave, o FQDN do servidor como nome comum do certificado, e com o nome “temp.csr” (Ex.: openssl req <opções>)

   e. Exibir o conteúdo do arquivo temp.csr (Ex.: cat <arquivo>)

   f. Verificar o arquivo de requisição gerado pelo utilitário openssl, exibindo as informações como texto (Ex.: openssl req <opções>)

   g. Através do utilitário openssl, gerar um arquivo de requisição de assinatura, criando um novo arquivo de chave, usando o FQDN do servidor como nome comum do certificado, e com o nome “temp2.csr” (Ex.: openssl req <opções>)

   h. Verificar o arquivo de requisição gerado pelo utilitário openssl, exibindo as informações como texto (Ex.: openssl req <opções>)

i. Através do utilitário CA.pl, criar uma nova requisição de assinatura, usando o FQDN do servidor como nome comum do certificado (Ex.: CA.pl <opções>)

j. Verificar o arquivo de requisição gerado pelo utilitário CA.pl, exibindo as informações como texto (Ex.: openssl req <opções>)

3. Habilidade para gerar um certificado auto-assinado de uma CA privada
   a. Através do utilitário openssl, assinar a requisição “temp.csr”, com a chave “temp.key”, 365 dias de validade, e usando como nome “temp.crt” (Ex.: openssl x509 <opções>)

   b. Verificar o conteúdo do arquivo temp.crt (Ex.: cat <arquivo>)

   c. Verificar o certificado assinado, exibindo as informações como texto (Ex.: openssl x509 <opções>)

   d. Verificar o hash do certificado temp.crt, usando o script c_hash (Ex.: c_hash <certificado>)

   e. Verificar informações do certificado temp.crt, usando o script c_info (Ex.: c_info <certtificado>)

   f. Através do utilitário CA.pl, criar um novo certificado auto-assinado, usando o fqdn do servidor como nome comum do certificado (Ex.: CA.pl <opções>)

   g. Verificar o certificado auto-assinado gerado pelo utilitário CA.pl, exibindo as informações como texto (Ex.: openssl x509 <opções>)

4. Habilidade de instalar uma chave e um certificado
   a. Copiar o arquivo temp.key para /etc/pki/tls/private/<FQDN>.key (Ex.: cp <origem> <destino>)

   b. Copiar o arquivo temp.crt para /etc/pki/tls/certs/<FQDN>.crt (Ex.: cp <origem> <destino>)

   c. Configurar o arquivo do mod_ssl (ssl.conf), definindo: (Ex.: vi <arquivo>)
      i. Arquivo de certificado SSL (SSLCertificateFile): /etc/pki/tls/certs/<FQDN>.crt
      ii. Arquivo de chave de certificado SSL (SSLCertificateKeyFile): /etc/pki/tls/private/<FQDN>.key

   d. Reiniciar o servidor httpd (Ex.: service <serviço> restart)

   e. Decriptografar o arquivo de chave, usando o comando openssl (Ex.: openssl rsa <opções>)

   f. Reiniciar o servidor httpd (Ex.: service <serviço> restart)

   g. No arquivo de configuração do mod_ssl, desativar o uso dos protocolos SSL v2 e v3 (Ex.: vi <arquivo>)

   h. Reiniciar o servidor httpd (Ex.: service <serviço> restart)

Simulado

1. São utilitários do pacote openssl
    a. c_hash
    b. c_signer
    c. CA.pl
    d. c_detail

2. … é o arquivo de configuração padrão do mod_ssl.

3. Sobre a geração de chaves de certificado, não é correto afirmar:
    a. chaves podem ser geradas de forma criptografada e protegidas por senhas
    b. chaves podem ser geradas individualmente ou durante a criação de requisição de assinatura de certificados
    c. chaves podem ser geradas com tamanhos diferentes
    d. chaves podem ser geradas com data de expiração

4. Para gerar a chave “server1.key”, de 2048 bits, usando o algorítimo RSA e proteção criptográfica des3, o comando … deve ser usado.

5. Para gerar uma requisição de assinatura, criando uma chave durante o processo, o comando … deve ser usado.

6. É possível gerar um certificado auto-assinado usando como argumento --newcert no comando ….

7. Para auto-assinar um certificado, basta informar a chave usada no processo de criação da requisição de assinatura.
    a. V
    b. F

8. A diretiva … é usada para definir a cadeia de certificados da Autoridade Certificadora.

9. A diretiva … pode ser usada para informar o certificado e sua respectiva chave, se os mesmos estiverem em um único arquivo codificado em PEM.

10. As diretivas … e … são responsáveis respectivamente para definir os protocolos seguros e o conjunto de cifras que estarão disponíveis para os clientes.

11. Os protocolos TLS v1.1 e v1.2 só podem ser usados em conjunto com o OpenSSL 1.0.1 ou posterior.
    a. V
    b. F

12. Em servidores virtuais é possível usar mais de um arquivo de certificado e de chave simultaneamente.
    a. V
    b. F

13. Servidores virtuais baseados em nome só conseguem usar o certificado definido no host virtual padrão, para uma dada combinação de endereço e porta.
    a. V
    b. F

14. O protocolo SSL v2 usa o fechamento de conexão do TCP, o que permite ataques onde o atacante envia pacotes forjados de fechamento, fazendo com que o destinatário se desconecte, sem perceber um ilegítimo término de dados.
    a. V
    b. F

15. O protocolo SSL v3 é vulnerável a ataques do tipo POODLE e por isso, seu uso não é recomendado.
    a. V
    b. F

208.3 Implementando um servidor proxy

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de instalar e configurar um servidor proxy, incluindo políticas de acesso, autenticação e uso de recursos.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários do Squid 3.x
- Métodos de restrição de acesso
- Métodos de autenticação de usuários clientes
- Esquema e conteúdo de ACL nos arquivos de configuração do Squid

Termos e utilitários:

- squid.conf
- http_access
- acl

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários do Squid 3.x

- Squid [1]
    - Squid é um cache e proxy web de encaminhamento. Ele tem uma grande variedade de usos, de acelerar um servidor web através do cache de requisições repetidas; a caching web, DNS e outras pesquisas de rede de computadores para um grupo de pessoas que compartilham recursos de rede; para auxiliar de segurança, filtrando o tráfego. Embora usado principalmente para HTTP e FTP, ele inclui suporte limitado para vários outros protocolos, incluindo TLS, SSL, Internet Gopher e HTTPS.
    - O Squid foi originalmente concebido para ser executado como um daemon em sistemas Unix-like. A portabilidade do Windows foi mantida até a versão 2.7. Novas versões disponíveis no Windows usam o ambiente Cygwin. O Squid é um software livre liberado sob a licença GNU General Public.
    - História
        - O Squid foi originalmente desenvolvido como o cache de objeto Harvest, parte do projeto Harvest da Universidade de Colorado Boulder. Os futuros trabalhos sobre o programa foram concluídos na Universidade da Califórnia, San Diego e financiado através de duas subvenções da Fundação Nacional de Ciência. Duane Wessels bifurcou a "última versão pré-comercial de Harvest" e renomeou para o Squid para evitar confusão com o fork comercial chamado Cache 2.0, que se tornou NetCache. O Squid a versão 1.0.0 foi lançado em Julho de 1996.
        - O Squid está agora desenvolvido quase que exclusivamente através de esforços voluntários.
        - Proxy cache Web é uma maneira de armazenar objetos Internet solicitados (por exemplo, dados como páginas da web) disponíveis através do HTTP, FTP, Gopher e protocolos em um sistema mais próximo do local requerente. Os navegadores da Web podem usar o cache Squid local como um servidor HTTP proxy, reduzindo o tempo de acesso, bem como o consumo de largura de banda. Isso é muitas vezes útil para provedores de serviços de Internet, para aumentar a velocidade de seus

clientes, e LANs que compartilham uma conexão de Internet. Como os servidores de cache são controlados pelo operador de serviço web, caching proxies não anonimizam o usuário e não devem ser confundido com proxies de anonimato.
        - Um programa cliente (por exemplo browser) ou tem que especificar explicitamente o servidor proxy que pretende utilizar (típico para clientes ISP), ou poderia estar usando um proxy sem qualquer configuração extra: "cache transparente", no caso em que todas as solicitações HTTP de saída são interceptados pelo Squid e todas as respostas são armazenadas em cache. O último é tipicamente um setup corporativo (todos os clientes estão na mesma LAN) e muitas vezes introduz as preocupações com a privacidade acima mencionados.
        - O Squid tem algumas características que podem ajudar a tornar conexões anônimas, como desabilitar ou alterar campos de cabeçalho específicos em solicitações HTTP de um cliente. Se esses estão definidos, e o que eles estão definidos para fazer, depende da pessoa que controla o computador que executa o Squid. As pessoas que solicitam páginas através de uma rede que utiliza de forma transparente o Squid podem não saber se essa informação está sendo registrada. No Reino Unido, pelo menos nas organizações, os usuários devem ser informados se os computadores ou conexões de internet estão sendo monitoradas.
    - Proxy reverso
        - O setup acima - caching o conteúdo de um número ilimitado de servidores web para um número limitado clientes - é o clássico. Outra configuração é "proxy reverso" ou "aceleração de servidor web" (usando http_port 80 accel vhost). Nesse modo, o cache serve um número ilimitado de clientes para um número limitado de - ou apenas um - servidores web.
        - Como exemplo, se slow.example.com é um servidor web "real", e www.example.com é o servidor de cache Squid que "acelera", a primeira vez que qualquer página é solicitada www.example.com, o servidor de cache iria obter a página real de slow.example.com, mas requisições posteriores iriam receber a cópia armazenada diretamente do acelerador (por um período configurável, que após a cópia armazenada seria descartada). O resultado final, sem qualquer ação pelos clientes, é menos tráfego para o servidor de origem, o que significa menos CPU e memória, e menor necessidade de largura de banda. Isso significa, porém, que o servidor de origem não pode informar com precisão sobre seus números de tráfego sem configuração adicional, como todas as requisições parecem ter vindo do proxy reverso. Uma maneira de se adaptar o relato no servidor de origem é usar o cabeçalho HTTP X-Forwarded-For relatado pelo proxy reverso, para obter o endereço IP real do cliente.
        - É possível um único servidor Squid servir tanto como um proxy normal e um proxy reverso simultaneamente. Por exemplo, uma empresa pode hospedar seu próprio site em um servidor web, com um servidor Squid atuando como um proxy reverso entre clientes (clientes acessando o site a partir de fora da empresa) e o servidor web. O mesmo servidor Squid poderia atuar como um cache web clássico, caching solicitações HTTP de clientes dentro da empresa (ou seja, os funcionários que acessam a Internet a partir de suas estações de trabalho), de modo a acelerar o acesso à web e reduzindo a demanda por banda larga.
- Arquivos de configuração
    - /etc/squid/squid.conf - arquivo de configuração principal

    - /etc/httpd/conf.d/squid.conf - configuração de cgi para acesso via web ao gerenciador de cache
- Utilitários
    - /usr/lib{,64}/squid/ - diretórios de utilitários e ajudantes do Squid

Métodos de restrição de acesso
- Através de listas de controle de acesso externas (usando programas auxiliares)
- Através de listas de controle de acesso
- Através de listas de controle de acesso com elementos externos (arquivos de texto), com ou sem expressões regulares, para
    - Endereço
    - Domínio
    - Tempo
    - URL
    - Porta
    - Protocolo
    - Método HTTP
    - Status HTTP
    - Navegador
    - Autenticação
    - SNMP
    - Cabeçalho
    - Certificado
    - Usuário

Métodos de autenticação de usuários clientes
- Autenticação [2]
    - Os usuários serão autenticados se o Squid está configurado para usar ACLs proxy_auth.
    - Os navegadores enviam credenciais de autenticação do usuário no cabeçalho de solicitação HTTP Authorization.
    - Se o Squid recebe um pedido e a lista de regra http_access obtém uma ACL proxy_auth ou uma ACL externa (external_acl_type) com o parâmetro %LOGIN, o Squid olha para o cabeçalho Autorização. Se o cabeçalho está presente, o Squid decodifica e extrai as credenciais do usuário.
    - Se o cabeçalho está faltando, o Squid retorna uma resposta HTTP com o status 407 (Proxy Authentication Required). O agente de usuário (navegador) recebe a resposta 407 e, em seguida, tenta localizar as credenciais do usuário. Às vezes isso significa uma busca de fundo, às vezes uma linha de pop-up para que o usuário digite um nome e uma senha. O nome e a senha são codificados e enviados no cabeçalho de Autorização para solicitações subseqüentes para o proxy.
    - NOTA: O nome e a senha são codificados utilizando o "base64" (Veja a seção 11.1 da RFC 2616). No entanto, base64 é uma codificação binária para texto somente, ele não criptografa as informações que codifica. Isso significa que o nome de usuário e senha são essencialmente "texto puro" entre o navegador e o proxy. Portanto, provavelmente não se deveria usar o mesmo nome de usuário e senha que se usa para o login de outras contas.
    - A autenticação é realmente realizada fora do principal processo do Squid. Quando o Squid é iniciado, ele gera um número de subprocessos de autenticação. Esses processos lêem as credenciais do usuário no stdin, e respondem com "OK" ou "ERR" no stdout. Essa técnica

permite que se use um número de diferentes protocolos de autenticação (chamados "esquemas" neste contexto). Quando vários esquemas de autenticação são oferecidos pelo servidor (Squid, nesse caso), é papel do User-Agent escolher um e autenticar usando ele. Pela RFC ele deve escolher o mais seguro que ele pode lidar; na prática, geralmente o Microsoft Internet Explorer escolhe o primeiro que foi oferecido que ele possa manipular e navegadores Mozilla são compatíveis com o bug do sistema Microsoft nesse campo.

- Além da autenticação básica bem conhecida, o Squid também suporta os esquemas de autenticação NTLM, Negotiate e Digest que fornecem métodos de autenticação mais seguros, em que a senha não é trocada em texto simples sobre o fio. Cada esquema tem seu próprio conjunto de ajudantes (helpers) e configurações auth_param. Observe que ajudantes para diferentes esquemas de autenticação utilizam protocolos diferentes para conversar com o Squid , então eles não podem ser misturados.
- Para obter informações sobre como configurar a autenticação NTLM ver exemplos de configuração NTLM.
- O código fonte do Squid empacota alguns backends de autenticação ("helpers") para autenticação. Esses incluem:
  - DB: Usa um banco de dados SQL.
  - getpwam: Usa o arquivo de senhas Unix old-fashioned.
  - LDAP: Usa o Lightweight Directory Access Protocol.
  - MSNT: Usa um domínio de autenticação do Windows NT.
  - MSNT-multi-domínio: Permite registrar-se para um dos vários domínios do Windows NT.
  - NCSA: Usa um arquivo de nome de usuário e senha NCSA-style.
  - NIS (ou YP): Usa o banco de dados NIS
  - PAM: Usa o esquema de Unix Pluggable Authentication Modules.
  - POP3: Usa um servidor de e-mail para validar as credenciais. Útil para single-signon para proxy e-mail.
  - RADIUS: Usa um servidor RADIUS para validação de login.
  - SASL: Usa bibliotecas SASL.
  - SMB: Usa um servidor SMB, como o Windows NT ou Samba.
  - SSPI: Autenticador nativo do Windows
- A documentação para cada um desses ajudantes podem ser encontradas em http://www.squid-cache.org/Doc/man/. Devido à sua simplicidade a autenticação básica tem, de longe, o maior número de ajudantes, mas os outros sistemas também têm vários ajudantes disponíveis.
- A fim de autenticar os usuários, precisa-se compilar e instalar um dos ajudantes de autenticação fornecidos, um dos outros, ou fornecer um próprio.
- Se diz ao Squid que programa ajudante de autenticação usar, com a diretiva auth_param no squid.conf. Especifica-se o nome do programa, além de todas as opções de linha de comando, se necessário. Por exemplo:
  - auth_param basic program /usr/local/squid/bin/ncsa_auth /usr/local/squid/etc/passwd

Esquema e conteúdo de ACL nos arquivos de configuração do Squid

- Lista de controle de acesso [3]
  - O sistema de controle de acesso do Squid é relativamente abrangente e difícil para algumas pessoas entenderem. Há dois componentes diferentes: elementos ACL, e listas de acesso. Uma lista de acesso consiste numa ação permitir ou negar seguido por um número de elementos de ACL.

- Ao carregar o arquivo de configuração, o Squid processa todas as linhas acl (diretivas) na memória como testes que podem ser realizados contra qualquer transação de solicitação. Os tipos de testes estão descritos em Elementos ACL. Por si só esses testes não fazem nada. Por exemplo; a palavra "Sunday" corresponde a um dia da semana, mas não indica em que dia da semana se está lendo isso.
- Para processar uma transação outro tipo de linha é usada. À medida que cada ação de processamento deve ocorrer uma verificação em execução testa que ação ou limitações devem ocorrer para a transação. Os tipos de verificações são descritos nas listas de acesso seguido de detalhes de como as verificações operam.
- Elementos ACL
  - src: endereço IP de origem (cliente)
  - dst: endereço IP de destino (servidor)
  - myip: o endereço IP local de uma conexão do cliente
  - arp: endereço Ethernet (MAC) correspondente
  - srcdomain: nome de domínio de origem (cliente)
  - dstdomain: nome de domínio de destino (servidor)
  - srcdom_regex: padrão de correspondência de expressão regular de origem (cliente)
  - dstdom_regex: padrão de correspondência de expressão regular de destino (servidor)
  - src_as: número de Sistema Autônomo de origem (cliente)
  - dst_as: número de Sistema Autônomo de destino (servidor)
  - peername: tag de nome atribuída ao cache_peer onde a requisição é esperada para ser enviada
  - time: hora do dia, e dia da semana
  - url_regex: padrão de correspondência de expressão regular de URL
  - urlpath_regex: padrão de correspondência de expressão regular de caminho de URL; deixa de fora o protocolo e hostname
  - port: número da porta do destino (servidor)
  - myport: número de porta local que o cliente está conectado
  - myportname: tag de nome atribuída a porta do squid de escuta que o cliente está conectado
  - proto: protocolo de transferência (HTTP, FTP, etc)
  - method: método de solicitação HTTP (GET, POST, etc)
  - http_status: status de resposta HTTP (200 302 404 etc.)
  - browser: padrão de correspondência de expressão regular sobre o cabeçalho de requisição user-agent
  - referer_regex: padrão de correspondência de expressão regular sobre o cabeçalho de requisição http-referer
  - ident: string correspondendo o nome do usuário
  - ident_regex: padrão de correspondência de expressão regular em nome de usuário
  - proxy_auth: autenticação de usuário via processos externos
  - proxy_auth_regex: padrão de correspondência de expressão regular na autenticação do usuário através de processos externos
  - snmp_community: string correspondendo a comunidade SNMP
  - maxconn: um limite para o número máximo de conexões a partir de um endereço IP único cliente
  - max_user_ip: um limite para o número máximo de endereços IP que um usuário pode acessar a partir de

        - req_mime_type: padrão de correspondência de expressão regular sobre o cabeçalho de solicitação do tipo de conteúdo
        - req_header: padrão de correspondência de expressão regular sobre um conteúdo de cabeçalho de solicitação
        - rep_mime_type: padrão de correspondência de expressão regular à resposta (conteúdo baixado) do tipo de conteúdo cabeçalho. Isso só é utilizável na diretiva http_reply_access, não http_access.
        - rep_header: padrão de correspondência de expressão regular a um teor de cabeçalho de resposta. Isso só é utilizável na diretiva http_reply_access, não http_access.
        - external: consulta via ajudante acl externo definido por external_acl_type
        - user_cert: corresponde contra os atributos em um certificado SSL do usuário
        - ca_cert: corresponde contra os atributos de usuários que emitem certificado SSL CA
        - ext_user: corresponde no campo user= retornado pelo ajudante acl externo definido por external_acl_type
        - ext_user_regex: padrão de correspondência de expressão regular no campo user= retornado pelo ajudante acl externo definido por external_acl_type
    - Listas de acessos
        - http_access: Permite que os clientes HTTP (browsers) acessem a porta HTTP. Essa é a lista de controle de acesso primário.
        - http_reply_access: Permite que os clientes HTTP (browsers) recebam a resposta ao seu pedido. Isso ainda restringe as permissões dadas por http_access, e destina-se principalmente a ser usado em conjunto com rep_mime_type acl para bloquear diferentes tipos de conteúdo.
        - icp_access: Permite caches vizinho consultar o cache local com ICP.
        - miss_access: Permite determinados clientes de transmitir a falta de cache através do cache local. Isso ainda restringe as permissões dadas por http_access, e destina-se principalmente a ser utilizado para fazer cumprir as relações entre irmãos, negando irmãos de encaminhar falta de cache através de seu cache.
        - cache: Define as respostas que não devem ser armazenados em cache.
        - url_rewrite_access: Controla quais as solicitações são enviadas através do pool redirecionador.
        - ident_lookup_access: Controla quais solicitações precisam de uma pesquisa de Ident.
        - allways_direct: Controla quais solicitações devem ser sempre enviadas diretamente para servidores de origem.
        - never_direct: Controla quais solicitações nunca devem ser enviadas diretamente para servidores de origem.
        - snmp_access: Controla o acesso de cliente SNMP oo cache.
        - broken_posts: Define as solicitações para os quais o squid acrescenta um CRLF extra após os corpos de mensagens de POST, como exigido por alguns servidores de origem quebrados.
        - cache_peer_access: Controla quais as solicitações podem ser encaminhadas para um determinado vizinho (cache_peer).
        - htcp_access: Controla quais máquinas remotas são capazes de fazer pedidos HTCP.
        - htcp_clr_access: Controla quais máquinas remotas são capazes de fazer pedidos HTCP CLR.

        - request_header_access: Controla quais cabeçalhos de solicitações são removidos quando violarem o protocolo HTTP.
        - reply_header_access: Controla quais cabeçalhos de respostas são removidos da entrega para o cliente quando violarem o protocolo HTTP.
        - delay_access: Controla quais as solicitações são tratadas pelo pool de atraso
        - icap_access: (substituído por adaptation_access em Squid-3.1) Quais as solicitações poderão ser enviadas para um servidor ICAP particular.
        - adaptation_access: Quais as solicitações poderão ser enviadas para um serviço de filtro específico ICAP ou PAEC.
        - log_access: Controla quais as solicitações são registradas. Esse é global e substitui as listas de acesso de arquivo específicas anexas à diretivas access_log.
    - Formatos:
        - acl <nome> <elemento> <restrição> - sintaxe padrão para define listas de elementos
        - <lista de acesso> <ação> <nome> - sintaxe padrão para definir listas de acessos
    - Exemplos:
        - acl localhost src 127.0.0.1/32 ::1 - define a lista de elementos localhost com as origens 127.0.0.1/32 e ::1
        - acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1 - define a lista de elementos to_localhost com os destinos 127.0.0.0/8, 0.0.0.0/32 e ::1
        - http_access allow localhost - aceita o acesso da lista localhost
        - http_access deny to_localhost - nega o acesso a lista to_localhost

Termos e utilitários

- /etc/squid/squid.conf - arquivo de configuração principal
- acl - diretiva de definição de elementos para controle de acesso
- http_access - diretiva de aplicação de controle de acesso nos elementos definidos

Referências

1. http://en.wikipedia.org/wiki/Squid_(software)
2. http://wiki.squid-cache.org/Features/Authentication
3. http://wiki.squid-cache.org/SquidFaq/SquidAcl

Exercícios práticos

1. Preparação para exercícios
    a. Instalar o pacote squid (Ex.: yum install <pacote>)

2. Métodos de restrição de acesso
    a. Verificar os binários do diretório /usr/lib64/squid (Ex.: ls <diretório>)

    b. Exibir o conteúdo do arquivo de configuração principal do squid (Ex.: cat <arquivo>)

c. Iniciar o serviço squid (Ex.: service <serviço> start)

d. Através da interface gráfica, configurar o navegador Firefox para conectar via proxy, em todos os protocolos, no endereço 127.0.0.1:3128

e. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

f. Verificar as últimas linhas do arquivo de log /var/log/squid/access.log (Ex.: tail <arquivo>)

g. Através da interface gráfica, configurar o navegador Firefox para conectar via proxy, em todos os protocolos, no endereço <IP do servidor>:3128

h. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

i. Verificar as últimas linhas do arquivo de log /var/log/squid/access.log (Ex.: tail <arquivo>)

j. Editar o arquivo de configuração principal do squid, antes da diretiva “http_access allow localnet”, definindo: (Ex.: vi <arquivo>)
    i. acl
        1. Nome: bloqueio_noturno
        2. Tipo: Horário
        3. Restrição: 00:00-06:00
    ii. acl
        1. Nome: rede_local
        2. Tipo: Endereço de Origem
        3. Restrição: <endereço de rede do IP principal/máscara de rede>
    iii. lista de acesso
        1. Ação: Negar
        2. acl: bloqueio_noturno
    iv. lista de acesso
        1. Ação: Permitir
        2. acl: rede_local

k. Reiniciar o serviço squid (Ex.: service <serviço> restart)

l. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

3. Métodos de autenticação de usuários clientes
    a. Através do utilitário htpasswd, crie o arquivo /etc/squid/squid.passwd, adicionando os usuários admin e user1 (Ex.: htpasswd <opções>)

    b. Editar o arquivo de configuração principal do squid, antes da diretiva “acl rede_local”, definindo: (Ex.: vi <arquivo>)
        i. auth_param
            1. Tipo: Básico
            2. Programa: /usr/lib64/squid/ncsa_auth
            3. Arquivo: /etc/squid/squid.passwd
        ii. acl
            1. Nome: usuarios
            2. Tipo: Autenticação
            3. Restrição: Requerido
        iii. lista de acesso
            1. Ação: Permitir
            2. acl: usuarios

    c. Recarregar o serviço squid (Ex.: service <serviço> reload)

    d. Através da interface gráfica, fechar o navegador Firefox

    e. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

4. Esquema e conteúdo de ACL nos arquivos de configuração do Squid
    a. Crie os arquivos vazios /etc/squid/squid.domain-blacklist e /etc/squid/squid.domain-whitelist (Ex.: touch <arquivo>)

    b. Editar o arquivo de configuração principal do squid, antes das diretivas anteriores, definindo: (Ex.: vi <arquivo>)
        i. acl

1. Nome: domain-whitelist
2. Tipo: Domínio de destino
3. Restrição: “/etc/squid/squid.domain-whitelist”

ii. acl

1. Nome: domain-blacklist
2. Tipo: Domínio de destino
3. Restrição: “/etc/squid/squid.domain-blacklist”

iii. lista de acesso

1. Ação: Permitir
2. acl: domain-whitelist

iv. lista de acesso

1. Ação: Negar
2. acl: domain-blacklist

c. Recarregar o serviço squid (Ex.: service <serviço> reload)

d. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

e. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com.br

f. Adicionar no arquivo squid.domain-blacklist o domínio google.com (Ex.: vi <arquivo>)

g. Recarregar o serviço squid (Ex.: service <serviço> reload)

h. Através da interface gráfica, usar o navegador Firefox para acessar o endereço www.google.com

Simulado

1. … é o arquivo de configuração do Squid.

2. São métodos de restrições válidos:
   a. por tempo
   b. por endereço de origem/destino
   c. por carga do servidor
   d. por número de conexões

3. É possível usar arquivos externos como fonte de correspondência para ACLs.
    a. V
    b. F

4. Elementos acl permitem ou negam uma regra enquanto lista de acessos define quais são os objetos que vão serem afetados pela ação.
    a. V
    b. F

5. São métodos de autenticação válidos:
    a. LDAP
    b. SMTP
    c. MSNT
    d. Kerberos

6. O método de autenticação POP3 permite proxy desse protocolo.
    a. V
    b. F

7. No Squid, ajudantes são programas externos que auxiliam no processo de autenticação.
    a. V
    b. F

8. Para usar o arquivo externo /etc/squid/sites-bloqueados.txt com bloqueio literal, as diretivas com os parâmetros … e ... devem ser usadas.

9. Para restringir acesso ao Squid durante 01:00 e 04:00 horas, as diretivas com os parâmetros … e ... devem ser usadas.

10. É possível criar uma lista de elementos com multiplos parâmetros do mesmo tipo, através de repetição das diretivas acl, usando o mesmo nome da lista.
    a. V
    b. F

208.4 Implementando Nginx como um servidor web e proxy reverso

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de instalar e configurar um proxy reverso Nginx. Configuração básica de Nginx como um servidor HTTP é incluído.
Áreas de conhecimentos chave:

- Nginx
- Proxy Reverso
- Servidor Web Básico

Termos e utilitários:

- /etc/nginx/
- nginx

Áreas de conhecimentos chave

Nginx

- Nginx [1]
    - Nginx (pronuncia-se "engine x") é um servidor web com um forte foco em alta concorrência, desempenho e baixo uso de memória. Ele também pode atuar como um servidor proxy reverso para HTTP, HTTPS, SMTP, POP3, IMAP e protocolos, bem como um balanceador de carga e um cache de HTTP.
    - Criado por Igor Sysoev em 2002, Nginx é executado em Unix, Linux, variantes BSD, Mac OS X, Solaris, AIX, HP-UX e Microsoft Windows. Lançado sob os termos de uma licença BSD-like, Nginx é livre e software de fonte aberta.
    - Características
        - O Nginx pode ser implantado para servir conteúdo dinâmico HTTP na rede usando FastCGI, manipuladores SCGI para scripts, servidores de aplicação WSGI ou módulo Phusion Passenger, e pode servir como um software balanceador de carga.
        - O Nginx usa uma abordagem assíncrona orientada a eventos para o tratamento de requisições, em vez do modelo Apache HTTP Server, que o padrão é uma abordagem de threads ou orientada a processos, onde o MPM Event é necessário para o processamento assíncrono. A arquitetura modular orientada a eventos do Nginx pode fornecer um desempenho mais previsível sob altas cargas.
        - Características do HTTP proxy e servidor WEB
            - Habilidade para lidar com mais de 10.000 conexões simultâneas com um baixo consumo de memória (~2,5 MB por 10k de conexões inativas HTTP keep-alive)
            - Manipulação de arquivos estáticos, arquivos de índice e indexação automática
            - Proxy reverso com caching
            - Balanceamento de carga com exames de saúde em banda
            - Tolerância a falhas
            - Suporte a TLS/SSL com SNI e OCSP stapling, via OpenSSL
            - Suporte com caching FastCGI, SCGI e uWSGI
            - Servidores virtuais baseados em nome e endereço IP
            - Compatível com IPv6

                - Suporte ao protocolo SPDY
                - WebSockets e HTTP/1.1 Upgrade (101 Switching Protocols)
                - Streaming de FLV e MP4
                - Autenticação de acesso à página Web
                - Compressão e descompressão gzip
                - Reescrita de URL
                - Log personalizado com compressão gzip on-the-fly
                - Limitação de taxa de resposta e solicitações simultâneas
                - Controle de Banda
                - Server Side Includes
                - Geolocalização baseado no endereço de IP
                - Rastreamento do usuário
                - WebDAV
                - Processamento de dados XSLT
                - Scripting Perl embarcado
            - Características do proxy de email
                - Suporte TLS/SSL
                - Suporte STARTTLS
                - Proxy SMTP, POP3 e IMAP
                - Autenticação usando um servidor HTTP externo
            - Outras características incluem atualização do executável e configuração sem perda de conexões de clientes, e uma arquitetura baseada em módulo.
- Arquivos de configuração
    - /etc/nginx/ - diretório de configuração do Nginx
        - nginx.conf - arquivo de configuração principal
        - conf.d - diretório de arquivos de configuração adicionais
            - default.conf - arquivo de configuração padrão do servidor
        - default.d - diretório de arquivos de configuração para o bloco do servidor padrão
- Configuração [2]
    - O nginx é composto por módulos que são controlados por diretivas específicas no arquivo de configuração. Diretivas estão divididas em diretrizes simples e diretrizes de bloco. A diretiva simples consiste no nome e parâmetros separados por espaços e termina com um ponto e vírgula (;). A diretiva de bloco tem a mesma estrutura que a diretiva simples, mas em vez de o ponto e vírgula que termina com um conjunto de instruções adicionais cercadas por chaves ({ e }). Se uma diretiva de bloco pode ter outras diretivas entre chaves, ela é chamada de um contexto (exemplos: eventos, http, servidor e localização).
    - Diretivas colocados no arquivo de configuração fora de qualquer contexto são considerados no contexto principal. Os eventos e diretrizes http residem no contexto principal, servidor no http, e localização no servidor.
    - O resto de uma linha após o símbolo # é considerado um comentário.
- Utilitários
    - nginx - (8) servidor HTTP e de proxy reverso, servidor de proxy de email
        - Exemplos:
            - nginx -s stop - desligamento rápido
            - nginx -s quit - desligamento elegante
            - nginx -s reload - recarrega a configuração
            - nginx -s reopen - reabre os arquivos de log

Proxy Reverso

- Exemplo [4]

```
http {
    proxy_cache_path /data/nginx/cache keys_zone=one:10m;
    server {
        proxy_cache one;
        location / {
            proxy_pass http://localhost:8080;
        }
        location /myapp1 {
            proxy_pass http://server1.mydomain.com:8080;
        }
    }
}
```

Servidor Web Básico

- Exemplo [5]

```
http {
    server {
        location / {
            root /data/www;
            index index.php index.html;
        }
        location /images/ {
            autoindex on;
            root /data/images;
    }
}
```

Termos e utilitários

- /etc/nginx/ - diretório de configuração do Nginx
- nginx - (8) servidor HTTP e de proxy reverso, servidor de proxy de email

Referências

1. http://en.wikipedia.org/wiki/Nginx
2. http://nginx.org/en/docs/beginners_guide.html
3. http://nginx.com/resources/admin-guide/
4. http://nginx.com/resources/admin-guide/web-server
5. http://nginx.com/resources/admin-guide/reverse-proxy/

Exercícios práticos

1. Preparação para exercícios
    a. Instalar o epel-release (Ex.: yum install <pacote>)

b. Instalar o pacote nginx (Ex.: yum install <pacote>)

c. Desabilitar permanentemente o repositório EPEL (Ex.: vi <arquivo>)

d. Renomear o arquivo de configuração customizado do Apache (teste.conf) para teste.conf-noload (Ex.: mv <origem> <destino>)

e. No arquivo de configuração principal do Apache, redefinir a escuta para o IP 127.0.0.1 na porta 80 (Ex.: vi <arquivo>)

f. Remover o módulo mod_ssl e seus arquivos associados (Ex.: yum remove <pacote>)

g. Reiniciar o serviço httpd (Ex.: service <serviço> restart)

h. Através da interface gráfica, remover a configuração de proxy do navegador Firefox

2. Proxy Reverso
   a. Verificar o conteúdo do arquivo de configuração principal do nginx (Ex.: cat <arquivo>)

   b. Verificar o conteúdo do arquivo de configuração padrão do bloco de servidor (Ex.: cat <arquivo>)

   c. No arquivo de configuração padrão do bloco do servidor, alterar o endereço de escuta do nginx para <IP do servidor> na porta 80 (Ex.: vi <arquivo>)

   d. Iniciar o nginx (Ex.: service <serviço> start)

   e. No arquivo de configuração padrão do bloco do servidor, comentar a localização / (Ex.: vi <arquivo>)

   f. Criar o arquivo vazio /etc/nginx/default.d/proxy.conf (Ex.: touch <arquivo>)

   g. No arquivo de configuração criado, definir: (Ex.: vi <arquivo>)
      i. Localização
         1. Caminho: /
         2. Proxy Reverso (proxy_pass): http://127.0.0.1:80/

   h. Recarregar a configuração do nginx através do utilitário nginx (Ex.: nginx <opções>)

   i. Através da interface gráfica, usar o navegador Firefox para acessar o endereço http://<IP do servidor>/

3. Servidor Web Básico
   a. Criar o arquivo vazio /etc/nginx/default.d/docs.conf

   b. No arquivo de configuração criado, definir: (Ex.: vi <arquivo>)
      i. Localização
         1. Caminho: /doc/

2. Índice automático: ligado
3. Raiz: /usr/share/

c. Recarregar a configuração do nginx através do utilitário nginx (Ex.: nginx <opções>)

d. Através da interface gráfica, usar o navegador Firefox para acessar o endereço http://<IP do servidor>/doc/

Simulado

1. O nginx suporta os protocolos HTTP, HTTPS, FTP, SMTP, POP3 e IMAP.
   a. V
   b. F

2. São características do nginx:
   a. proxy reverso com cache
   b. proxy de email
   c. compatível com IPv6
   d. suporta o protocolo SPDY

3. … é o arquivo de configuração principal do nginx.

4. A configuração do nginx suporta dois tipos de diretivas: diretivas simples e de contexto.
   a. V
   b. F

5. Para se ativar o nginx como proxy reverso, basta usar o contexto location junto com a diretiva proxy_pass.
   a. V
   b. F

6. A diretiva … é necessária para ativar o mecanismo de caching.

7. É possível usar uma instância nginx para servidor de proxy reverso para várias localidades, simultaneamente.
   a. V
   b. F

8. Para servir conteúdo web básico, basta usar o contexto location junto com a diretiva documentroot.
   a. V
   b. F

9. Para definir o uso de índice automático em um a localização, a diretiva … deve ser usada.

10. A diretiva index é usada para definir a ordem dos arquivos de índice que são procurados nos diretórios.
    a. V
    b. F

Tópico 209: Compartilhando Arquivo

209.1 Configuração do Servidor Samba

Visão geral

Peso: 5
Descrição: Os candidatos devem ser capazes de configurar um servidor SAMBA para vários clientes. Esse objetivo também inclui configurando o Samba para login de clientes e configurando o grupo de trabalho no qual o servidor participa e definindo diretórios compartilhados e impressoras. Também está coberto configurando um cliente Linux para usar um servidor Samba. Resolução de problemas de instalação também é testado.
Áreas de conhecimentos chave:

- Documentação do Samba 3
- Arquivos de configuração do Samba
- Ferramentas e utilitários Samba
- Montando compartilhamentos Samba no Linux
- Daemons Samba
- Mapeando nomes de usuários Windows para nomes de usuários Linux
- Segurança de Nível de Usuário e Nível de Compartilhamento

Termos e utilitários:

- smbd, nmbd
- smbstatus, testparm, smbpasswd, nmblookup
- smbclient
- net
- /etc/smb/
- /var/log/samba/

Áreas de conhecimentos chave

Documentação do Samba 3

- Samba [1]
  - Samba é um software livre da reimplementação do protocolo de rede SMB/CIFS, e foi originalmente desenvolvido por Andrew Tridgell. Ele fornece serviços de arquivo e impressão para vários clientes Windows e pode ser integrado com um domínio do Windows Server, ou como um controlador de domínio (DC) ou como um membro do domínio. A partir da versão 4, suporta Active Directory e domínios do Windows NT.
  - O Samba roda na maioria dos Unix, OpenVMS e sistemas Unix-like, como o Linux, Solaris, AIX e as variantes BSD, incluindo OS X Server, da Apple, e cliente OS X (versão 10.2 e superior). Ele é padrão em quase todas as distribuições do Linux e é comumente incluído como um serviço básico do sistema em outros sistemas operacionais baseados em Unix também. O Samba é liberado sob os termos da GNU General Public License. O nome Samba vem de SMB (Server Message Block), o nome do protocolo padrão usado pelo sistema de arquivos de rede Microsoft Windows.
  - História antiga
    - Andrew Tridgell desenvolveu a primeira versão do Samba Unix em Dezembro de 1991 e Janeiro de 1992, como estudante de doutorado na Universidade Nacional da

Austrália, usando um packet sniffer para fazer análise de rede do protocolo usado pelo software de servidor DEC Pathworks. Na época dos primeiros lançamentos, versões 0.1, 0.5 e 1.0, todos a partir do primeiro semestre de Janeiro de 1992, ele não tinha um nome próprio, e Tridgell apenas o chamou de "um servidor de arquivos Unix para Dos Pathworks". No momento da versão 1.0, ele percebeu que "tinha de fato implementado o protocolo NetBIOS" e que "esse software poderia ser usado com outros clientes de PC".

- Com foco na interoperabilidade com o LAN Manager da Microsoft, Tridgell lançou "NetBIOS para Unix", nbserver, versão 1.5 em Dezembro de 1993. Essa versão foi o primeiro a incluir um software cliente, bem como um servidor. Além disso, neste momento o GPL2 foi escolhido como licença.
- No meio da série 1.5, o nome foi mudado para SMBSERVER. No entanto, Tridgell teve um aviso de marca registrada da empresa "Sintaxy", que vendeu um produto chamado TotalNet Advanced Server e detinha a marca de "SMBServer". O nome "Samba" foi derivado executando o comando grep do Unix através do dicionário do sistema à procura de palavras que continham as letras S, M e B, nessa ordem (ou seja grep -i '^s.*m.*b' /usr/share/dict/words).
- As versões 1.6, 1.7, 1.8 e 1.9, seguiram de forma relativamente rápida, com essa última a ser lançada em Janeiro de 1995. Tridgell considera a adoção do CVS em Maio 1996 para marcar o nascimento do Samba Team, embora tivesse havido contribuições de outras pessoas, especialmente Jeremy Allison, anteriormente.
- A versão 2.0.0 foi lançado em Janeiro de 1999, e a versão 2.2.0 em Abril de 2001.

- História da versão
  - A versão 3.0.0, lançado em 23 de Setembro de 2003, foi uma grande atualização. O Samba ganhou a habilidade de se juntar Active Directory como um membro, embora não como um controlador de domínio. Subsequentes lançamentos pontuais para a 3,0, foram adicionados novos recursos menores. Atualmente, a versão mais recente dessa série é a 3.0.37, lançada em 1º de Outubro de 2009, e enviada numa base voluntária. A série 3.0.x oficialmente chegou ao fim de vida em 5 de Agosto de 2009.
  - A versão 3.1 foi utilizada apenas para o desenvolvimento.
  - Com a versão 3.2, o projeto decidiu se mudar para versões em tempos. Novas versões principais, tais como 3.3, 3.4, etc. aparecem a cada 6 meses. Os novos recursos serão adicionados somente quando um lançamento maior é feito, lançamentos pontuais serão apenas para correções de bugs. Além disso, 3,2 marcou uma mudança de licença da GPL2 a GPL3, com algumas partes liberado sob LGPL3. A principal mudança técnica na versão 3.2 foi a gerar automaticamente a maior parte do código DCE/RPC que costumava ser artesanal. A versão 3.2.0 foi lançada em 1 de Julho de 2008. Será atualizada em uma base como necessária para questões de segurança apenas e sua versão atual é 3.2.15 a partir de 1 de Outubro de 2009. A série 3.2.x oficialmente chegou ao final de sua vida útil, em 1 de Março de 2010.
  - A versão 3.3 foi lançada em 27 de Janeiro de 2009 e está agora na versão 3.3.16 nesse ramo.
  - A versão 3.4 foi lançada em 03 de Julho de 2009. Essa foi a primeira versão a incluir tanto o código fonte Samba 3 e Samba 4.
  - A versão 3.4.17 foi lançada em 30 de Abril de 2012. É a mais recente versão estável da série Samba 3.4.

- A versão 3.5 foi lançada em 1º de Março de 2010. Essa foi a primeira versão a incluir suporte experimental para SMB2.
- A versão 3.6 foi lançada em 9 de Agosto de 2011. Esse é o primeiro ramo que inclui suporte completo para SMB2.
- A versão 4 foi lançada em 11 de Dezembro de 2012. Trata-se de uma grande reformulação que permite o Samba ser um controlador de domínio do Active Directory, participar plenamente em um Windows Active Directory Domain. Sua primeira prévia técnica (4.0.0TP1) foi lançada em Janeiro de 2006, após 3 anos de desenvolvimento.
- A versão 4.1 foi lançada em 11 de Outubro de 2013. Ela possui suporte para SMB3.
- A versão 4.2 foi lançada em 4 de Março de 2015. Ela suporta compactação de arquivos baseados em Btrfs, snapshots e integração winbind.

- Características
  - O Samba permite o compartilhamento de arquivos e impressoras entre computadores com o Windows e computadores que executam Unix. É uma aplicação de dezenas de serviços e uma dezena de protocolos, incluindo:
    - NetBIOS sobre TCP/IP (NBT)
    - SMB
    - CIFS (uma versão melhorada do SMB)
    - DCE/RPC ou mais especificamente, MSRPC, a suíte de protocolos de vizinhança de rede
    - Um servidor WINS também conhecido como um NetBIOS Name Server (NBNS)
    - A suíte NT Domain de protocolos que inclui Logons no domínio NT
    - Banco de dados Gerente de Contas de Segurança (SAM)
    - Serviço de Autoridade de Segurança Local (LSA)
    - Serviço de impressão estilo NT (SPOOLSS), NTLM e mais recentemente Active Directory Logon que envolve uma versão modificada do Kerberos e uma versão modificada do LDAP
    - Servidor DFS
  - Todos esses serviços e protocolos são freqüentemente erradamente referidos como apenas NetBIOS ou SMB. Os protocolos NetBIOS e WINS são depreciados no Windows.
  - O Samba cria compartilhamentos de rede para diretórios Unix escolhidos (incluindo todos contidos subdiretórios). Esses parecem aos usuários do Microsoft Windows como pastas normais do Windows acessíveis através da rede. Usuários Unix podem montar os compartilhamentos diretamente como parte de sua estrutura de arquivo usando o comando smbmount ou, em alternativa, pode usar um utilitário, smbclient (libsmb) instalado com o Samba para ler os compartilhamentos com uma interface semelhante a uma linha de comando padrão de programa FTP. Cada diretório pode ter diferentes privilégios de acesso sobreposto sobre as proteções normais de arquivos Unix. Por exemplo: os diretórios home teriam acessos de leitura/escrita para todos os utilizadores conhecidos, permitindo que cada acessassem seus próprios arquivos. No entanto, eles ainda não teriam acessos aos arquivos de outros, a menos que a permissão normalmente exista. Note que o compartilhamento netlogon, normalmente distribuído como um compartilhamento somente leitura do /etc/samba/netlogon, é o diretório logon para scripts de logon do usuário.
  - Os serviços Samba são implementados como dois daemons:

        - smbd, que fornece os serviços de partilha de arquivo e impressora e
        - nmbd, que fornece o serviço de nome NetBIOS para endereço IP. O NetBIOS sobre TCP/IP requer algum método para mapeamento nomes de computadores NetBIOS para os endereços IP de uma rede TCP/IP.
    - A configuração do Samba é conseguida através da edição de um único arquivo (normalmente instalado como /etc/smb.conf ou /etc/samba/smb.conf). O Samba também pode fornecer scripts de logon do usuário e implementação da política de grupo através poledit.
    - O Samba está incluído na maioria das distribuições Linux e é iniciado durante o processo de inicialização. No Red Hat, por exemplo, o script é executado /etc/rc.d/init.d/smb no momento da inicialização, e começa ambos os daemons. Ele não está incluído no Solaris 8, mas uma versão compatível com Solaris 8 está disponível no site do Samba.
    - O Samba inclui uma ferramenta de administração web chamado Samba Web Administration Tool (SWAT). O SWAT foi removido a partir da versão 4.1.
- Documentação [2]
  - Os documentos a seguir são distribuídos no empacotamento do código
    - Samba-3-HOWTO
    - Samba-3-ByExample

Arquivos de configuração do Samba

- /etc/samba
  - smb.conf - arquivo de configuração principal
  - lmhosts - mapeamento estático de nome NetBIOS para endereço IP (similiar ao /etc/hosts)
  - smbusers - mapeamento de contas de usuários
- smb.conf [3]
  - Sinopse
    - O arquivo smb.conf é um arquivo de configuração para a suíte Samba. O smb.conf contém informações de configuração de tempo de execução para os programas do Samba. O arquivo smb.conf é projetado para ser configurado e administrado pelo programa SWAT(8).
  - Formato do arquivo
    - O arquivo consiste em seções e parâmetros. A seção começa com o nome da seção entre colchetes e continua até a próxima seção começar. Seções contêm parâmetros da forma:
      - nome = valor
    - O arquivo é baseado em linha - ou seja, cada linha terminada em nova linha representa ou um comentário, um nome de seção ou um parâmetro.
    - Seção e nomes de parâmetro não são case-sensitive.
    - Apenas o primeiro sinal de igualdade em um parâmetro é significativo. Espaços em branco antes ou depois do primeiro sinal de igual é descartado. Espaços em branco iniciando e finalizando e na seção interna e nomes de parâmetro são irrelevantes. Espaços em branco iniciando e finalizando um valor de parâmetro são descartados. Espaços em branco interno dentro de um valor de parâmetro é mantido na íntegra.
    - Qualquer linha que começa com um caractere ponto e vírgula (";") ou um hash ("#") é ignorado, assim como as linhas que contêm apenas espaço em branco.
    - Qualquer linha que termina em um "\" continua na próxima linha na moda UNIX habitual.

        - Os valores a seguir ao sinal de igualdade nos parâmetros são todos uma string (sem aspas necessárias) ou um boolean, que pode ser dada como yes/no, 1/0 ou true/false. Maiúsculas e minúsculas não são significativas em valores booleanos, mas são preservadas em valores de string. Alguns itens, como máscaras de criação são numéricos.
    - Descrição das seções
        - Cada seção no arquivo de configuração (exceto para a seção [global]) descreve um recurso compartilhado (conhecido como um "share"). O nome da seção é o nome do recurso compartilhado e os parâmetros na seção definem os atributos do compartilhamento.
        - Há três seções especiais, [global], [homes] e [printers], que são descritos nas seções especiais. As notas a seguir aplicam-se a descrições de seção comuns.
        - Um compartilhamento consiste em um diretório cujo acesso está sendo dado, mais uma descrição dos direitos de acesso que são concedidos ao usuário do serviço. Algumas opções de limpeza também são especificáveis.
        - Seções são ou serviços de compartilhamento de arquivos (usadas pelo cliente como uma extensão de seus sistemas de arquivos nativos), ou serviços de impressão (usadas pelo cliente para acessar os serviços de impressão na máquina executando o servidor).
        - Seções podem ser designadas serviços de convidados, caso em que nenhuma senha é necessária para acessá-las. A conta de convidado Unix especificada é usada para definir privilégios de acesso nesse caso.
        - Com exceção dos serviços de convidados, seções irão solicitar uma senha para acessá-las. O cliente fornece o nome de usuário. Como os clientes mais antigos só fornecem senhas e não nomes de usuários, pode-se especificar uma lista de nomes de usuário para verificar contra a senha usando a opção user = na definição do compartilhamento. Para clientes modernos, como Windows 95/98/ME/NT/2000, isso não deve ser necessário.
        - Os direitos de acesso concedidos pelo servidor são mascarados pelos direitos de acesso concedidos ao usuário Unix especificado ou convidado, pelo sistema host. O servidor não concede mais acesso do que as subvenções do sistema host.
        - A seção de exemplo a seguir define um espaço de compartilhamento de arquivo. O usuário tem acesso de escrita ao caminho /home/bar. O compartilhamento é acessado através do nome de compartilhamento foo:
            - [foo]
            - path = /home/bar
            - read only = no
        - A seção de exemplo a seguir define um compartilhamento de impressão. A participação é somente leitura, mas para impressão. Ou seja, o único acesso de gravação permitido é através de chamadas para abrir, gravar e fechar um arquivo spool. O parâmetro guest ok significa que o acesso será permitido ao usuário convidado padrão (especificado em outros lugares):
            - [aprinter]
            - path = /usr/spool/public
            - read only = yes
            - printable = yes
            - guest ok = yes
    - Seções especiais

- A seção [global]
  - Parâmetros nessa seção aplicam-se ao servidor como um todo, ou são padrões para as seções que não definem especificamente determinados itens. Veja as notas de acordo com parâmetros para obter mais informações.
- A seção [homes]
  - Se uma seção chamada [homes] está incluída no arquivo de configuração, os serviços de conexão de clientes para seus diretórios podem ser criados em tempo real pelo servidor.
  - Quando a solicitação de conexão é feita, as seções existentes são verificadas. Se uma correspondência for encontrada, ela é usada. Se nenhuma correspondência for encontrada, o nome da seção solicitada é tratado como um nome de usuário e pesquisa no arquivo de senha local. Se o nome existe e a senha correta foi dada, um compartilhamento é criado por clonagem da seção [homes].
  - Algumas modificações são feitas então para o compartilhamento recém-criado:
    - O nome do compartilhamento é alterado de homes para o nome de usuário localizado
    - Se nenhum caminho foi dado, o caminho está definido para o diretório home do usuário
  - Se decide-se usar uma linha path = na sua seção [homes], pode ser útil usar a macro %S. Por exemplo:
    - path = /data/pchome/%S
  - é útil se você tiver diretórios home diferentes para seus PCs do que para o acesso Unix.
  - Essa é uma maneira rápida e simples para dar um grande número de clientes acesso a seus diretórios com um mínimo de barulho.
  - Um processo semelhante ocorre se o nome da seção solicitada é "homes", exceto que o nome do compartilhamento não é alterado para o usuário solicitante. Esse método de usar a seção [homes] funciona bem se diferentes usuários compartilham um PC cliente.
  - A seção [homes] pode especificar todos os parâmetros que uma seção normal de serviço pode especificar, embora alguns fazem mais sentido do que outros. O que se segue é uma seção [homes] típica e adequada:
    - [homes]
    - read only = no
  - Um ponto importante é que, se o acesso a visitantes é especificado na seção [homes], todos os diretórios home serão visíveis para todos os clientes sem uma senha. No caso muito improvável que isso é realmente desejável, é aconselhável também especificar acesso somente leitura.
  - A flag browseable para diretórios auto home será herdada da flag browseable global, não da bandeira browseable de [homes]. Isso é útil, uma vez que significa a criação browseable = no na seção [homes] irá esconder o [homes] share, mas faz quaisquer diretórios auto home visíveis.
- A seção [printers]
  - Essa seção funciona como [homes], mas para impressoras.

- Se uma seção [printers] ocorre no arquivo de configuração, os usuários são capazes de se conectar a qualquer impressora especificada no arquivo local printcap do servidor.
- Quando um pedido de conexão é feito, as seções existentes são verificadas. Se uma correspondência for encontrada, ela é usada. Se nenhuma correspondência é encontrada, mas uma seção [homes] existir, é utilizado tal como descrito acima. Caso contrário, o nome da seção solicitada é tratada como um nome de impressora e o arquivo printcap apropriado é varrido para ver se o nome da seção solicitada é um nome de compartilhamento de impressora válida. Se for encontrada uma correspondência, um novo compartilhamento de impressora é criado por clonagem da seção [printers].
- Algumas modificações são feitas então para o compartilhamento recém-criado:
    - O nome do compartilhamento é definido para o nome da impressora localizada
    - Se nenhum nome da impressora foi dada, o nome da impressora é definido para o nome da impressora localizada
    - Se o compartilhamento não permitir o acesso a visitantes e nenhum nome de usuário foi dado, o nome de usuário é definido para o nome da impressora localizada.
- O serviço [printers] deve ser printable - se você especificar o contrário, o servidor irá se recusar a carregar o arquivo de configuração.
- Normalmente o caminho especificado é o de um diretório de spool gravável com o sticky bit definido nele. Uma entrada típica [printers] se parece com isso:
    - [printers]
    - path = /usr/spool/public
    - guest ok = yes
    - printable = yes
- Todos os apelidos dados para uma impressora no arquivo printcap são nomes de impressoras legítimos, tanto quanto o interesse do servidor. Se o seu subsistema de impressão não trabalha assim, terá que se criar um pseudo-printcap. Esse é um arquivo que consiste em uma ou mais linhas como esta:
    - alias|alias|alias|alias…
- Cada acrônimo deve ser um nome de impressora aceitável para o seu subsistema de impressão. Na seção [global], especifique o novo arquivo como printcap. O servidor só irá reconhecer nomes encontrados no pseudo-printcap, o que naturalmente pode conter quaisquer aliases que se deseja. A mesma técnica pode ser utilizada simplesmente para limitar o acesso a um subconjunto de suas impressoras locais.
- Um alias, por sinal, é definido como qualquer componente da primeira entrada de um registro printcap. Os registros são separados por novas linhas, componentes (se houver mais do que um) são separados por símbolos de barras verticais (|).
- Nota
    - Em sistemas SYSV que utilizam lpstat para determinar quais as impressoras são definidas no sistema, pode-se ser capaz de usar o

nome printcap = lpstat para obter automaticamente uma lista de impressoras. Veja a opção printcap para mais detalhes.

- Exemplo
  - [global]
  - workgroup = LPI
  - [homes]
  - read only = no

Ferramentas e utilitários Samba

- smbstatus(1) - relata as conexões atuais do Samba
  - Exemplos:
    - smbstatus - relata as conexões atuais do Samba
    - smbstatus -b - da uma saída resumida
    - smbstatus -v - da uma saída verbose
    - smbstatus -L - faz o smbstatus listar apenas locks
    - smbstatus -p - imprime a lista de processos smbd e termina. Util para scripts
    - smbstatus -S - faz o smbstatus listar apenas compartilhamentos
    - smbstatus -u <usuário> - seleciona informação relevante ao usuário apenas
- testparm(1) - verifica por exatidão interna em um arquivo de configuração smb.conf
  - Exemplos:
    - testparm - verifica por exatidão interna em um arquivo de configuração smb.conf
    - testparm -s - sem essa opção, o testparm irá aguardar por um enter depois de imprimir os nomes de serviços e antes de despejar as definições de serviços
    - testparm -v - com essa opção especificada, o testparm irá exibir também todas as opções que não estão sendo usadas no smb.conf e seus valores padrão
    - testparm --section-name <seção> - despeja a seção especificada
- smbpasswd(8) - altera a senha SMB de um usuário
  - Exemplos:
    - smbpasswd <usuário> - altera a senha SMB de um usuário
    - smbpasswd -a - adiciona o usuário ao arquivo de senhas do samba
    - smbpasswd -x - exclui o usuário do arquivo de senhas do samba
    - smbpasswd -d - desabilita a conta do usuário do samba
    - smbpasswd -e - habilita a conta do usuário do samba
- nmblookup(1) - cliente NetBIOS sobre TCP/IP usado para pesquisar nomes NetBIOS
  - Exemplos:
    - nmblookup <nome> - pesquisa o nome NetBIOS e retorna o endereço IP
    - nmblookup -B <endereço> - envia a consulta para o endereço de broadcast especificado
    - nmblookup -U <endereço> - envia a consulta para o endereço de unicast especificado
    - nmblookup -T - faz com que qualquer endereço IP encontrado seja consultado de forma reversa no DNS
- smbclient(1) - um cliente estilo ftp para acessar recursos SMB/CIFS em servidores
  - Exemplos:
    - smbclient //<servidor>/<serviço> - conecta ao serviço com um cliente estilo ftp
    - smbclient -M <endereço> - envia uma mensagem para o servidor especificado
    - smbclient -L <endereço> - lista os recursos disponíveis no servidor especificado
    - smbclient -U <usuário> - especifica o usuário a ser usado

- net(8) - ferramenta para administração do Samba e servidores CIFS remotos
    - Exemplos:
        - net -U <usuário> - especifica o usuário a ser usado
        - net -S <endereço> - especifica o servidor a ser usado
        - net file - grupo de funções em arquivos remotos abertos
        - net share - grupo de funções de gerência em compartilhamentos
        - net session - grupo de funções de gerência de sessões
        - net user - grupo de funções de gerência de usuários
        - net group - grupo de funções de gerência de grupos
        - net password - altera senha do usuário no servidor alvo
        - net time - exibe/define hora
        - net lookup - verifica nome de hosts /endereços IP
        - net status - exibe o estado do servidor

Montando compartilhamentos Samba no Linux

- Descrição geral [4]
    - Usando um compartilhamento de um servidor Samba dentro de um sistema de arquivos Unix depende de uma série de componentes individuais. Precisa-se de pelo menos o smbfs (que já não é mantido) ou os modernos módulos CIFS do kernel. Embora a documentação mais velha diz para usar os smbfs, tem muitas restrições e o cifs deve ser favorecido.
    - Uma vez que o sistema de arquivos CIFS está incluído no kernel padrão do Linux, ele é simples para construí-lo, quer como módulo ou embutido. Se construir o módulo CIFS (por exemplo, obter uma versão mais recente, com correções de bugs) e não instalá-lo no local padrão , então pode-se precisar carregá-lo antes de montar um compartilhamento ou deixar o modprobe carregá-lo via modprobe.conf, a fim de obter o módulo cifs mais recente, em vez daquele fornecido com a distribuição.
    - O comando básico para montar é mount -t cifs //<servidor>/<serviço> <ponto de montagem>. Cada componente precisa de sua própria atenção.
    - O utilitário mount chama um ajudante de montagem, geralmente mount.cifs que chama para o kernel. O auxiliar de montagem mount.cifs é o ajudante do espaço de usuário e é necessário para analisar nomes TCP/IP e recuperar usuário e senha, e também fazer simples formatação das opções de montagem.
    - O servidor escuta as conexões de entrada do cliente através de TCP/IP, e portanto tem o endereço IP, e geralmente os nomes de host TCP configurados para eles, mas os usuários muitas vezes se referem ao servidor pelo seu "nome NetBIOS" (nome RFC1001). Para montar utilizando o cliente CIFS, um nome TCP (em vez de nome NetBIOS) deve ser especificado para o servidor. Para resolver o <servidor> para um endereço IP, precisa-se de um servidor DNS que conhece o endereço IP ou o cliente precisa do módulo NSS para WINS. É uma biblioteca compartilhada que deve estar no caminho do ldd. Normalmente em /usr/lib. Também tem que adicionar a opção de WINS para o hosts no /etc/nsswitch.conf. O utilitário smbclient também pode ser usado para identificar o nome TCP ou endereço IP de um servidor (identificado pelo seu nome NetBIOS).
    - O ponto de montagem deve ser um diretório em outra parte do sistema de arquivos, que deve existir.

Daemons Samba

- smbd [5]

    - smbd é o daemon do servidor que fornece serviços de compartilhamento de arquivos e de impressão para clientes Windows. O servidor fornece serviços de arquivo e impressora para clientes que usam o protocolo SMB (ou CIFS). Isso é compatível com o protocolo LanManager, e pode servir os clientes LanManager. Esses incluem MSCLIENT 3.0 para DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE para Macintosh, e smbfs para Linux.
    - Uma extensa descrição dos serviços que o servidor pode fornecer é dada na página do manual para o arquivo de configuração do controle dos atributos desses serviços (ver smb.conf(5)). Essa página de manual não vai descrever os serviços, mas vai concentrar-se na aspectos administrativos de funcionamento do servidor.
    - Por favor, note que existem implicações de segurança significativas para a execução desse servidor, e o manual smb.conf(5) deve ser considerado como leitura obrigatória antes de prosseguir com a instalação.
    - Uma sessão é criada sempre que um cliente solicita. Cada cliente recebe uma cópia do servidor para cada sessão. Essa cópia então serve todos as conexões feitas pelo cliente durante a sessão. Quando todas as conexões do cliente estão fechadas, a cópia do servidor para aquele cliente termina.
    - O arquivo de configuração, e todos os arquivos que ele inclui, são recarregados automaticamente a cada minuto, se eles mudam. Você pode forçar uma recarga através do envio de um SIGHUP para o servidor. Recarregar o arquivo de configuração não afetará as conexões com qualquer serviço que já estão estabelecidos. Ou o usuário terá que sair do serviço, ou o smbd morto e reiniciado.
- nmbd [6]
    - nmbd é um servidor que entende e pode responder solicitações de serviço de nome NetBIOS sobre IP, como os produzidos por clientes SMB/CIFS, como o Windows 95/98/ME, Windows NT, Windows 2000, Windows XP e os clientes LanManager. Participa também nos protocolos de navegação que compõem a visão de "Rede" do Windows.
    - Clientes SMB/CIFS, quando iniciam, podem desejar localizar um servidor SMB/CIFS. Ou seja, eles querem saber qual o número IP que um host especificado está usando.
    - Entre outros serviços, o nmbd vai ouvir tais pedidos, e se o seu próprio nome NetBIOS for especificado, ele irá responder com o número IP da máquina em que está sendo executado. O seu "próprio nome NetBIOS" é, por padrão, o nome de DNS primário do host em que está sendo executado, mas esse pode ser substituído pelo nome NetBIOS no smb.conf. Assim o nmbd irá responder ao broadcast as consultas para o(s) seu(s) próprio(s) nome(s). Nomes adicionais para o nmbd responder podem ser definidos através de parâmetros no arquivo de configuração smb.conf(5).
    - O nmbd também pode ser usado como um servidor WINS (Windows Internet Name Server). O que isso significa, basicamente, é que ele vai agir como um servidor de banco de dados WINS, criando um banco de dados de pedidos de registro de nomes que ele recebe e respondendo a consultas de clientes para esses nomes.
    - Além disso, o nmbd pode atuar como um proxy WINS, retransmitindo consultas de broadcast a partir de clientes que não entendem como falar o protocolo WINS com um servidor WINS.

Mapeando nomes de usuários Windows para nomes de usuários Linux

- Mapeamento de usuários [7] (Seção 13.4.4)
    - Em algumas situações, é inevitável que o nome de logon do Windows do usuário será diferente do ID de login que o usuário tem sobre o servidor Samba. É possível

criar um arquivo especial no servidor Samba, que permitirá que o nome do usuário Windows seja mapeado para um diferente nome de usuário Unix/Linux. O arquivo smb.conf também deve ser alterado para que seção [global] contenha o parâmetro:
    - username map = /etc/samba/smbusers
- O conteúdo do arquivo /etc/samba/smbusers é mostrado aqui:
    - parsonsw: "William Parsons"
    - marygee: geeringm
- Nesse exemplo, a conta de usuário do Windows “William Parsons” será mapeada para o usuário Unix “parsonsw”, e a conta de usuário Windows “geeringm” será mapeada para o usuário Unix “marygee”.

Segurança de Nível de Usuário e Nível de Compartilhamento

- Modos de Segurança Samba [8] (Seção 3.3)
    - A rede Microsoft Windows utiliza um protocolo que foi originalmente chamado o protocolo Server Message Block (SMB). Desde algum momento em torno de 1996, o protocolo foi mais conhecido como o protocolo Common Internet Filesystem (CIFS).
    - No mundo SMB/CIFS, existem apenas dois tipos de segurança: a nível de usuário e nível de compartilhamento. Refere-se a esses coletivamente como os níveis de segurança. Na execução desses dois níveis de segurança, o Samba fornece flexibilidades que não estão disponíveis com os servidores MS Windows NT4/200x. Na verdade, o Samba implementa a segurança em nível de compartilhamento apenas de uma forma, mas tem quatro formas de implementar a segurança a nível de usuário. Coletivamente, chama-se as implementações Samba dos modos de segurança de níveis de segurança. Eles são conhecidos como modos de compartilhamento, usuário, domínio, ADS e servidor.
    - Um servidor SMB informa o cliente, no momento da configuração de uma sessão, o nível de segurança que o servidor está sendo executado. Há duas opções: em nível de compartilhamento e em nível de usuário. Qual desses dois o cliente recebe afeta a maneira como o cliente, então, tenta se autenticar. Ele não afeta diretamente (em qualquer medida) a forma como o servidor Samba faz segurança. Isso pode soar estranho, mas ele se encaixa com a abordagem de cliente/servidor de SMB. No SMB tudo é iniciado e controlado pelo cliente, e o servidor só pode dizer ao cliente o que está disponível e se uma ação é permitida.
    - O termo cliente refere-se a todos os agentes se é uma estação de trabalho do Windows, um servidor Windows, um outro servidor Samba, ou qualquer sabor SMB ou aplicativo cliente CIFS (por exemplo, smbclient) que fazem uso dos serviços prestados por um servidor SMB/CIFS.
    - Segurança de Nível de Usuário
        - Descreve-se a segurança em nível de usuário em primeiro lugar porque é mais simples. Na segurança em nível de usuário, o cliente envia uma solicitação de configuração de sessão diretamente seguindo a negociação de protocolo. Esse pedido fornece um nome de usuário e senha. O servidor pode aceitar ou rejeitar aquela combinação de nome de usuário/senha. Nessa fase, o servidor não tem idéia de que compartilhamento o cliente acabará por tentar se conectar, por isso, não pode basear o aceitar/rejeitar em outra coisa senão:
            - o nome de usuário/senha.
            - o nome da máquina cliente.
        - Se o servidor aceita as credenciais de nome de usuário/senha, o cliente espera ser capaz de montar os compartilhamentos (usando uma conexão árvore) sem

especificar uma senha. Ele espera que todos os direitos de acesso serão como credenciais de usuário/senha definidas que foi especificado na configuração de sessão inicial.

- Também é possível que um cliente envie vários pedidos de configuração de sessão. Quando o servidor responde, dá ao cliente uma uid para usar como uma marca de autenticação para esse nome de usuário/senha. O cliente pode manter vários contextos de autenticação dessa forma (WinDD é um exemplo de uma aplicação que faz isso).
- Os nomes de contas de usuário de rede Windows são insensíveis ao caso, o que significa que caracteres maiúsculos e minúsculos na conta nome são consideradas equivalentes. Eles são ditos a serem case-preserving, mas não caso significativo. Sistemas Windows e LanManager anterior ao Windows NT versão 3.10 têm senhas de maiúsculas e minúsculas que não eram necessariamente caso de preservação. Todos os sistemas familiares Windows NT tratam senhas como case-preserving e case-sensitive.
- Exemplo de configuração:
  - No arquivo smb.conf, o parâmetro que define a segurança a nível de usuário é: security = user.
  - Isso é padrão desde o Samba 2.2.x.

- Segurança de Nível de Compartilhamento
  - Em segurança de nível de compartilhamento, o cliente autentica-se separadamente para cada ação. Ele envia uma senha junto com cada pedido de conexão árvore (montagem de compartilhamento), mas não explicitamente envia um nome de usuário com essa operação. O cliente espera que uma senha seja associado a cada compartilhamento, independente do utilizador. Isso significa que o Samba tem que descobrir qual o nome de usuário o cliente provavelmente quer usar, porque o usuário não é explicitamente enviado para o servidor SMB. Alguns servidores SMB comerciais, tais como NT realmente associa senhas diretamente com os compartilhamentos em segurança em nível de compartilhamento, mas Samba sempre usa o esquema de autenticação Unix onde é um par nome de usuário/senha que é autenticado, não um par compartilhamento/senha.
  - Para compreender os paralelos de rede do MS Windows, deve-se pensar em termos de MS Windows 9x/Me, onde pode-se criar uma pasta compartilhada que fornece somente leitura ou acesso total, com ou sem uma senha.
  - Muitos clientes enviam um pedido de definição de sessão, mesmo que o servidor esteja em segurança em nível de compartilhamento. Eles normalmente enviam um nome de usuário válido, mas nenhuma senha. O Samba registra esse nome em uma lista de possíveis nomes de usuário. Quando o cliente emite uma solicitação de conexão árvore, ele também acrescenta a essa lista o nome do compartilhamento que tenta se conectar (útil para diretórios home) e quaisquer usuários listados no parâmetro usuário no arquivo smb.conf. A senha é então verificada, por sua vez contra esses possíveis nomes de usuário. Se for encontrada uma correspondência, em seguida, o cliente é autenticado como esse usuário.
  - Se não for fornecida a lista de nomes de usuários possíveis, o Samba faz uma chamada de sistema Unix para encontrar a conta de usuário que tem um password que corresponde a fornecida a partir do banco de dados conta padrão. Em um sistema que não tem a facilidade de interruptor de serviço de nome (NSS), essas pesquisas serão a partir do banco de dados /etc/passwd. Em sistemas NSS

habilitados, a pesquisa irá para as bibliotecas que tenham sido especificadas no arquivo nsswitch.conf. As entradas nesse arquivo em que as bibliotecas são especificadas são:

- passwd: files nis ldap
- shadow: files nis ldap
- group: files nis ldap

- No exemplo mostrado aqui (não é susceptível de ser utilizado na prática) a pesquisa verificará /etc/passwd e /etc/group, se não
- encontrado irá verificar o NIS, e então o LDAP.

Termos e utilitários

- smbd, nmbd
  - smbd - (8) servidor para prover serviços SMB/CIFS aos clientes
  - nmbd - (8) servidor de nomes NetBIOS para prover serviço de resolução de nomes NetBIOS sobre TCP/IP para clientes
- smbstatus, testparm, smbpasswd, nmblookup
  - smbstatus - (1) relata as conexões atuais do Samba
  - testparm - (1) verifica por exatidão interna em um arquivo de configuração smb.conf
  - smbpasswd - (8) altera a senha SMB de um usuário
  - nmblookup - (1) cliente NetBIOS sobre TCP/IP usado para pesquisar nomes NetBIOS
- smbclient - (1) um cliente estilo ftp para acessar recursos SMB/CIFS em servidores
- net - (8) ferramenta para administração do Samba e servidores CIFS remotos
- /etc/smb/ - diretório dos arquivos de configuração do samba
- /var/log/samba/ - diretório dos arquivos de log do samba

Referências

1. http://en.wikipedia.org/wiki/Samba_(software)
2. https://www.samba.org/samba/docs/
3. smb.conf(5)
4. https://wiki.samba.org/index.php/Mounting_samba_shares_from_a_unix_client
5. smbd(8)
6. nmbd(8)
7. Samba-3-HOWTO

Exercícios práticos

1. Preparação para exercícios
   a. Instalar os pacotes cifs-utils, samba, samba-client e samba-doc (Ex.: yum install <pacote>)

   b. Definir o boleano do SELinux samba_enable_home_dirs como ligado, de forma permanente (Ex.: setsebool -P samba_enable_home_dirs on)

2. Documentação do Samba 3
   a. Varrer o conteúdo do diretório /usr/share/doc/samba-doc-<versão> (Ex.: find <diretório>)

b. Verificar o manual do arquivo smb.conf (Ex.: man <manual>)

3. Arquivos de configuração do Samba
   a. Listar o conteúdo do diretório /etc/samba (Ex.: ls <diretório>)

   b. Renomear o arquivo /etc/samba/smb.conf para /etc/samba/smb.conf-orig (Ex.: mv <origem> <destino>)

   c. Criar o arquivo vazio /etc/samba/smb.conf (Ex.: touch <arquivo>)

   d. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

   e. Verificar se o arquivo de configuração do samba está sintaticamente correto, exibindo todas as opções (incluindo as padrões) (Ex.: testparm <opções>)

   f. No arquivo de configuração principal do samba, definir: (Ex.: vi <arquivo>)
      i. Seção global
         1. Grupo de trabalho (workgroup): CURSOLPI

   g. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

   h. No arquivo de configuração principal do samba, definir: (Ex.: vi <arquivo>)
      i. Seção homes
         1. Navegável (browseable): não
         2. Somente leitura (read only): não

   i. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

4. Ferramentas e utilitários Samba
   a. Adicionar o usuário testesamba1 (Ex.: useradd <opções>)

   b. Definir uma senha para o usuário testesamba1 (Ex.: passwd <opções>)

   c. Para o serviço iptables (Ex.: service <serviço> stop)

   d. Iniciar o serviço smb (Ex.: service <serviço> start)

   e. Iniciar o serviço nmb (Ex.: service <serviço> start)

   f. Varrer o conteúdo do diretório /var/log/samba (Ex.: find <diretório>)

   g. Através de uma a estação de trabalho Windows remota, no Windows Explorer, tentar acessar o endereço \\<IP do servidor>, fornecendo as credenciais do usuário testesamba1 (ignore o erro)

   h. Adicionar o usuário testesamba1 no samba, definindo uma senha para o mesmo (Ex.: smbpasswd <opções>)

i. Através de uma a estação de trabalho Windows remota, no Windows Explorer, acessar o endereço \\<IP do servidor>, fornecendo as credenciais do usuário testesamba1

j. Através de uma a estação de trabalho Windows remota, no Windows Explorer, acessar o compartilhamento \\<IP do servidor>\testesamba1

k. Exibir o estado das conexões com o servidor samba (Ex.: smbstatus)

l. Resolva o endereço NetBIOS do nome do servidor local (Ex.: nmblookup <nome>)

5. Montando compartilhamentos Samba no Linux
   a. Através do comando smbclient, listar os compartilhamentos do servidor samba local, usando as credenciais do usuário testesamba1 (Ex.: smbclient <opções>)

   b. Através do comando smbclient, estabelecer conexão com o servidor samba local, usando as credenciais do usuário testesamba1, acessando o compartilhamento referente ao seu diretório home (Ex.: smbclient <opções>)

   c. Listar a relação de comandos disponíveis no smbclient conectado (Ex.: help)

   d. Listar o conteúdo do compartilhamento remoto (Ex.: ls)

   e. Desconectar do compartilhamento pelo smbclient (Ex.: quit)

   f. Criar o diretório /mnt/smb-teste (Ex.: mkdir <diretório>)

   g. Montar o compartilhamento testesamba1, do samba local, no diretório /mnt/smb-teste, usando as credenciais do usuário testesamba1 (Ex.: mount <opções>)

   h. Criar o diretório /mnt/smb-teste/temp01 (Ex.: mkdir <diretório>)

   i. Listar o conteúdo o diretório /home/testesamba1 (Ex.: ls <diretório>)

   j. Desmontar o diretório /mnt/smb-teste (Ex.: umount <diretório>)

6. Daemons Samba
   a. Verificar as últimas linhas do arquivo de log /var/log/samba/log.smbd (Ex.: tail <arquivo>)

   b. Verificar as últimas linhas do arquivo de log /var/log/samba/log.nmbd (Ex.: tail <arquivo>)

7. Mapeando nomes de usuários Windows para nomes de usuários Linux
   a. No arquivo de configuração principal do samba, definir: (Ex.: vi <arquivo>)
      i. Seção global
         1. Mapa de nome de usuários (username map): /etc/samba/smbusers

   b. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

   c. Recarregar o serviço smb (Ex.: service <serviço> reload)

d. Definir o apelido “Usuario de Teste 1” para o usuário testesamba1 no mapa de usuários (Ex.: vi <arquivo>)

e. Através do comando smbclient, listar os compartilhamentos do servidor samba local, usando o nome de usuário “Usuario de Teste 1” e as credenciais do usuário testesamba1 (Ex.: smbclient <opções>)

8. Segurança de Nível de Usuário e Nível de Compartilhamento
    a. No arquivo de configuração principal do samba, definir: (Ex.: vi <arquivo>)
        i. Seção global
            1. Segurança (security): share

    b. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

    c. Recarregar o serviço smb (Ex.: service <serviço> reload)

    d. Montar o compartilhamento testesamba1, do samba local, no diretório /mnt/smb-teste, usando apenas a senha do usuário testesamba1 (Ex.: mount <opções>)

    e. Desmontar o diretório /mnt/smb-teste (Ex.: umount <diretório>)

    f. No arquivo de configuração principal do samba, definir: (Ex.: vi <arquivo>)
        i. Seção global
            1. Segurança (security): user

    g. Verificar se o arquivo de configuração do samba está sintaticamente correto (Ex.: testparm)

    h. Recarregar o serviço smb (Ex.: service <serviço> reload)

    i. Iniciar o serviço iptables (Ex.: service <serviço> start)

Simulado

1. O Samba traz em seu empacotamento de código fonte, páginas de manuais e vasta documentação, incluindo um documento de como fazer teórico e um documento de como fazer por exemplos.
    a. V
    b. F

2. …, … e … são os nomes de documentos pdfs distribuídos com o código do Samba 3.

3. Em distribuições RH e derivados, a documentação extra pode ser obtida através do pacote samba-doc.
    a. V
    b. F

4. Apesar da documentação extra, ela não é necessária para a maioria das funcionalidades que podem ser implementadas com o samba, que podem ser consultadas diretamente no manual do arquivo de configuração.
    a. V
    b. F

5. …, … e … são respectivamente o arquivo de configuração global do samba, o arquivo padrão de mapeamento de usuários e o arquivo padrão de resolução estática de nomes NetBIOS.

6. São seções especiais do arquivo de configuração do samba:
    a. global
    b. local
    c. homes
    d. netlogon

7. Excluindo a seção global, todas as outras seções declaradas no arquivo de configuração do samba, são consideradas como compartilhamento, seja de arquivos ou de impressoras.
    a. V
    b. F

8. … é o parâmetro que define o grupo de trabalho/domínio ao qual o servidor samba faz parte.

9. Através do utilitário … é possível gerenciar quase todos os recursos do servidor samba.

10. A ferramenta … serve para validar os arquivos de configuração do samba.

11. Para verificar o endereço de um servidor com o nome NetBIOS “servidor”, o comando …, distribuído com o samba, deve ser usado.

12. Para alterar a senha de um usuário samba, os comandos … e ... podem ser usados.

13. O comando … é usado para verificar o estado das conexões no servidor samba.

14. Através do comando … e da opção -t cifs, é possível montar compartilhamentos Samba/Windows em sistemas operacionais Linux.

15. O módulo smbfs passou a fazer parte do kernel do Linux, não sendo necessário mais o uso do comando smbmount, para realizar a montagem de compartilhamentos Samba/Windows.
    a. V
    b. F

16. Os daemons … e … são respectivamente responsáveis por compartilhar recursos de arquivos e impressão e trabalhar com resolução de nomes NetBIOS.

17. Após a alteração dos arquivos de configuração do samba, para trocar o nome do servidor, o daemon … deve ser recarregado.

18. Após a mudança de propriedades de compartilhamento de um diretório, basta aguardar a releitura automática de configuração, que ocorre a cada 300 segundos, para a mudança entrar em vigor para todas as conexões.
    a. V
    b. F

19. Para mapear o usuário de Windows “John Doe” para a conta Linux “guest”, qual é a definição necessária no arquivo de mapeamento de usuários.

20. … é o arquivo padrão de mapeamento de usuários no samba.

21. O modelo de mapeamento de contas de Windows para contas Linux no samba é N para 1.
    a. V
    b. F

22. Ao se usar a segurança a nível de compartilhamento no samba, o usuário deve informar login e senha para acessar um compartilhamento.
    a. V
    b. F

23. Ao se usar a segurança a nível de usuário no samba, não é possível usar recursos anonimamente.
    a. V
    b. F

24. A segurança a nível de usuário em servidores SMB possui mais de uma equivalência no Samba.
    a. V
    b. F

25. … é o parâmetro que define o nível de segurança e ele deve ser especificado na seção … do arquivo smb.conf.

209.2 Configuração do servidor NFS

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de exportar sistemas de arquivos usando NFS. Esse objetivo inclui restrição de acesso, montando um sistema de arquivos NFS em um cliente e protegendo o NFS.
Áreas de conhecimentos chave:

- Arquivos de configuração do NFSv3
- Ferramentas e utilitários NFS
- Restrições de acesso para certos hosts e/ou subredes
- Opções de montagem no servidor e cliente
- TCP Wrappers
- Consciência do NFSv4

Termos e utilitários:

- /etc/exports
- exportfs
- showmount
- nfsstat
- /proc/mounts
- /etc/fstab
- rpcinfo
- mountd
- portmapper

Áreas de conhecimentos chave

Arquivos de configuração do NFSv3

- NFS [1]
    - Network File System (NFS) é um protocolo de sistema de arquivos distribuídos originalmente desenvolvido pela Sun Microsystems em 1984, permitindo um usuário em um computador cliente acessar arquivos através de uma rede bem como o armazenamento local é acessado. O NFS, como muitos outros protocolos, baseia-se no sistema de Open Network Computing Remote Procedure Call (ONC RPC). O Network File System é um padrão aberto definido em RFCs, permitindo qualquer pessoa implementar o protocolo.
    - Versões e variações
        - A Sun utilizou a versão 1 apenas para fins experimentais na casa. Quando a equipe de desenvolvimento acrescentou alterações substanciais para o NFS versão 1 e disponibilizaram fora da Sun, eles decidiram lançar a nova versão como v2, de modo que a interoperação de versão e versão de emergência RPC poderiam ser testadas.
        - NFSv2
            - A versão 2 do protocolo (NFSv2) foi implementada pela primeira vez no SunOS versão 2.0, que foi lançado em Maio de 1985. As pessoas envolvidas na criação do NFS versão 2 incluem Russel Sandberg, Bob Lyon, Bill Joy, Steve Kleiman, e outros. O protocolo é definido na RFC 1094, publicado em Março de 1989.
            - O NFSv2 operava inicialmente apenas sobre UDP. Seus projetistas pretendiam a manter o lado do servidor sem estado (stateless), com bloqueio (por exemplo) realizado fora do protocolo central. A decisão de fazer o sistema de arquivos sem estado foi uma decisão importante, uma vez que

fizeram a recuperação de falhas do servidor triviais (toda os clientes de rede se congelam quando um servidor cai, mas uma vez que o servidor reparou o sistema de arquivos e reiniciou, todo o estado para repetir cada transação foi contido em cada RPC, foi repetido pelo(s) stub(s) cliente.) Essa decisão de projeto permitiu aplicações Unix (que não podiam tolerar falhas do servidor de arquivos) ignorar o problema.
    - A interface do sistema de arquivos virtual permitiu uma aplicação modular, refletida em um protocolo simples. Em Fevereiro de 1986, as implementações foram demonstradas para os sistemas operacionais como o System V lançamento 2, Microsoft DOS, e VAX/VMS usando Eunice.
    - Devido a limitações de 32 bits, o NFSv2 permitia apenas os primeiros de 2 GB de um arquivo serem lidos.
- NFSv3
    - Versão 3 (RFC 1813, Junho de 1995), acrescentou:
        - suporte para tamanhos de arquivo de 64 bits e compensações, para lidar com arquivos maiores que 2 gigabytes (GB);
        - suporte para escrita assíncrona no servidor, para melhorar o desempenho de gravação;
        - atributos de arquivos adicionais em muitas respostas, para evitar a necessidade de voltar a buscá-los;
        - uma operação READDIRPLUS, para obter identificadores de arquivo e atributos, juntamente com os nomes de arquivos na varredura de um diretório;
        - outras melhorias sortidas.
    - No momento da introdução da versão 3, o suporte do fornecedor para o TCP como um protocolo de camada de transporte começou a aumentar. Enquanto vários fornecedores já tinham adicionado suporte para NFS versão 2 com o TCP como um transporte, a Sun Microsystems adicionou suporte para TCP como um transporte para NFS, ao mesmo tempo que adicionou suporte para a versão 3. Usando o TCP como um transporte fez o uso do NFS através de uma WAN mais viável.
- NFSv4
    - Versão 4. (RFC 3010, Dezembro de 2000; revisto pela última vez na RFC 7530, Março de 2015), influenciado pelo AFS e CIFS, inclui melhorias de desempenho, exige uma forte segurança, e introduz um protocolo com estado (stateful). A versão 4 se tornou a primeira versão desenvolvida com a Internet Engineering Task Force (IETF), após a Sun Microsystems entregar o desenvolvimento dos protocolos NFS.
    - O NFS versão 4.1 (RFC 5661, Janeiro de 2010) tem como objetivo fornecer suporte ao protocolo para tirar proveito de implementações de servidores em cluster, incluindo a capacidade de fornecer acesso paralelo escalável de arquivos distribuídos entre vários servidores (extensão pNFS). O NFS versão 4.2 está sendo desenvolvido atualmente.
- Outras extensões
    - WebNFS, uma extensão para a versão 2 e a versão 3, permite o NFS integrar mais facilmente em navegadores Web e habilita a operação através de firewalls. Em 2007, a Sun Microsystems abriu o código fonte de sua implementação WebNFS do lado do cliente.

- Vários protocolos de lado a banda se tornaram associados com NFS, incluindo:
  - O protocolo de byte-range consultivo Network Lock Manager (NLM) (adicionado para suportar APIs UNIX System V de bloqueio de arquivo).
  - O protocolo de relatório de quota remota (RQUOTAD) (para permitir os usuários do NFS ver suas cotas de armazenamento de dados em servidores NFS).
- O NFS sobre RDMA é uma adaptação do NFS que usa RDMA como um transporte.

- Portmap [2]
  - O mapeador de porta (rpc.portmap ou apenas portmap, ou rpcbind) é um serviço Open Network Computing Remote Procedure Call (ONC RPC) que roda em nós de rede que fornecem outros serviços ONC RPC.
  - A versão 2 do protocolo de mapeador de porta mapeia pares de número de programa/número de versão ONC RPC para o número da porta de rede para essa versão do programa. Quando um servidor ONC RPC é iniciado, ele vai dizer o mapeador de porta, para cada par de número de programa/número de versão determinado, que ele suporta para um protocolo de transporte particular (TCP ou UDP), o número da porta que ele está usando para aquele determinado par número de programa/número de versão naquele protocolo de transporte. Os clientes que desejam fazer uma chamada ONC RPC a uma versão específica de um determinado serviço ONC RPC devem primeiro contatar o mapeador de porta na máquina do servidor para determinar a real porta TCP ou UDP para usar.
  - As versões 3 e 4 do protocolo, chamadas protocolo rpcbind, mapeiam um par número de programa/número de versão, e um indicador que especifica um protocolo de transporte, para um endereço de destino da camada de transporte, para esse par número de programa/número de versão, naquele protocolo de transporte.
  - O serviço mapeador de porta sempre usa TCP ou UDP 111; uma porta fixa é necessária para isso, como um cliente não seria capaz de obter o número da porta para o serviço de mapeador de porta, a partir do próprio mapeador de porta.
  - O mapeador de porta deve ser iniciado antes de quaisquer outros servidores RPC serem iniciados.
  - O serviço mapeador de porta apareceu pela primeira vez no SunOS 2.0.
- Arquivos de configuração
  - /etc/exports [3]
    - O arquivo /etc/exports contém uma tabela de sistemas de arquivos físicos locais em um servidor NFS, que são acessíveis para clientes NFS. O conteúdo do arquivo é mantido pelo administrador do sistema do servidor.
    - Cada sistema de arquivos nessa tabela tem uma lista de opções e uma lista de controle de acesso. A tabela é utilizada pelo exportfs(8) para dar informação ao mountd(8).
    - O formato do arquivo é semelhante ao arquivo de exports do SunOS. Cada linha contém um ponto de exportação e uma lista separada por espaços em branco de clientes permitidos para montar o sistema de arquivos nesse ponto. Cada cliente listado pode ser imediatamente seguido por um lista entre parêntesis, de opções de exportação, separada por vírgulas para esse cliente. Nenhum espaço em branco é permitido entre um cliente e sua lista de opções.

        - Além disso, cada linha pode ter uma ou mais especificações para opções padrão após o nome do caminho, sob a forma de um traço ("-"), seguido por uma lista de opção. A lista de opções é usada para todas as exportações subsequentes sobre essa linha única.
        - As linhas em branco são ignoradas. Um sinal de hash ("#") introduz um comentário no final da linha. As inscrições podem ser continuadas através de novas linhas usando uma barra invertida. Se um nome de exportação contiver espaços, deve ser citado o uso de aspas duplas. Pode-se especificar espaços ou outros caracteres incomuns no nome de exportação usando uma barra invertida seguida pelo código do caractere como três dígitos octais.
        - Para aplicar as alterações nesse arquivo, execute exportfs -r ou reinicie o servidor NFS.
        - Exemplos:
            - / master(rw) trusty(rw,no_root_squash)
            - /projects proj*.local.domain(rw)
            - /usr *.local.domain(ro) @trusted(rw)
            - /home/joe pc001(rw,all_squash,anonuid=150,anongid=100)
            - /pub *(ro,insecure,all_squash)
            - /srv/www -sync,rw server @trusted @external(ro)
            - /foo 2001:db8:9:e54::/64(rw) 192.0.2.0/24(rw)
            - /build buildhost[0-9].local.domain(rw)
    - /etc/fstab [4]
        - O comando mount(8) atribui um sistema de arquivos a hierarquia espaço de nome do sistema em um determinado ponto de montagem. O arquivo /etc/fstab descreve como o mount(8) deve reunir a hierarquia de nome de arquivo de um sistema a partir de vários sistemas de arquivos independentes (incluindo sistemas de arquivos exportados por servidores NFS). Cada linha no arquivo /etc/fstab descreve um único sistema de arquivos, seu ponto de montagem, e um conjunto de opções padrão de montagem para aquele ponto de montagem.
        - Para montagens de sistema de arquivo NFS, uma linha no arquivo /etc/fstab especifica o nome do servidor, o nome do caminho do diretório do servidor exportado para montar, o diretório local, que é o ponto de montagem, o tipo de sistema de arquivos que está sendo montado, e uma lista de opções de montagem que controlam a forma como o sistema de arquivos está montado e como o cliente NFS se comporta quando acessar arquivos sobre esse ponto de montagem. O quinto e o sexto campos em cada linha não são utilizados pelo NFS, assim convencionalmente cada contêm o dígito zero. Por exemplo:
            - <servidor>:<caminho> <ponto de montagem> <tipo de sistema de arquivos> <opção>,<opção>,... 0 0
        - O hostname e caminho de exportação do servidor são separados por dois pontos, enquanto as opções de montagem são separados por vírgulas. Os demais campos são separados por espaços em branco ou tabulações.

Ferramentas e utilitários NFS

- exportfs(8) - matem a tabela de sistema de arquivos exportados por NFS
    - Exemplos:
        - exportfs - exibe a lista de diretórios exportados
        - exportfs -a - exporta ou para de exportar todos os diretórios

        - exportfs -r - reexporta todos os diretórios, sincronizando o /var/lib/nfs/etab com o /etc/exports
        - exportfs -u - para de exportar um ou mais diretórios
        - exportfs -v - modo verbose
- showmount(8) - mostra informações de montagem para um servidor NFS
    - Exemplos:
        - showmount - mostra informações de montagem para um servidor NFS
        - showmount -a - lista ambos o endereço do cliente e o diretório montado no formato host:dir
        - showmount -d - lista apenas os diretórios montados por algum cliente
        - showmount -e - mostra a lista de exportação do servidor NFS
- nfsstat(8) - lista estatísticas NFS
    - Exemplos:
        - nfsstat - lista estatísticas NFS
        - nfsstat -s - imprime apenas estatísticas do lado do servidor
        - nfsstat -c - imprime apenas estatísticas do lado do cliente
        - nfsstat -n - imprime apenas estatísticas do NFS (o padrão é exibir NFS e RPC)
        - nfsstat -2 - imprime apenas estatísticas do NFSv2
        - nfsstat -3 - imprime apenas estatísticas do NFSv3
        - nfsstat -4 - imprime apenas estatísticas do NFSv4
        - nfsstat -m - imprime informação sobre cada um dos sistemas de arquivos NFS montados
        - nfsstat -r - imprime apenas estatísticas do RPC
        - nfsstat -v - modo verbose
        - nfsstat -l - imprime na forma de lista
- rpcinfo(8) - relata informações do RPC
    - Exemplos:
        - rpcinfo - relata informações do RPC
        - rpcinfo <endereço> - relata informações do RPC no endereço especificado
        - rpcinfo -p - sonda o rpcbind no host usando a versão 2 do protocolo
        - rpcinfo -s - exibe uma lista concisa

Restrições de acesso para certos hosts e/ou subredes

- exports [3]
    - Clientes NFS podem ser especificado em uma série de maneiras:
        - único host
            - Pode-se especificar um host ou por um nome abreviado reconhecido pelo resolvedor, o nome de domínio totalmente qualificado, um endereço IPv4 ou um endereço IPv6. Os endereços IPv6 não devem estar dentro de colchetes em /etc/exports para que não sejam confundidos com classe de caracteres curinga de correspondências.
        - netgroups
            - Netgroups NIS podem ser dados como @grupo. Somente a parte do host de cada um dos membros de netgroups é considerada na verificação de adesão. Partes de hosts vazios ou aqueles que contêm um único traço (-) são ignoradas.
        - wildcards

            - Nomes de máquinas podeem conter os caracteres curinga * e ?, ou podem conter listas de classes de caracteres dentro de [colchetes]. Isso pode ser usado para fazer o arquivo exports mais compacto; por exemplo, *.cs.foo.edu corresponde a todos os hosts no domínio cs.foo.edu. Como esses caracteres também combinam os pontos em um nome de domínio, o modelo dado também irá corresponder todos os hosts dentro de qualquer subdomínio de cs.foo.edu.
        - Redes IP
            - Também pode-se exportar os diretórios para todos os hosts em uma sub rede IP simultaneamente. Isso é feito especificando um par endereço IP e máscara de rede como endereço/máscara de rede, onde a máscara pode ser especificada no formato decimal com pontos, ou como um comprimento de máscara contígua. Por exemplo, tanto '/255.255.252.0' ou '/22' anexado à base de rede de endereços IPv4 resultada em sub-redes idênticas com 10 bits de host.
            - Os endereços IPv6 devem usar um comprimento de máscara contígua e não devem estar dentro de colchetes para evitar confusão com classe de caracteres curingas. Os caracteres curinga geralmente não funcionam em endereços IP, embora possam trabalhar por acidente quando pesquisas reversas de DNS falhar.

Opções de montagem no servidor e cliente

- Servidor [3]
    - Opções gerais
        - secure
        - rw
        - async
        - sync
        - no_wdelay
        - nohide
        - crossmnt
        - no_subtree_check
        - insecure_locks
        - no_auth_nlm
        - mountpoint=path
        - mp
        - fsid=num|root|uuid
        - nordirplus
        - refer=path@host[+host][:path@host[+host]]
        - replicas=path@host[+host][:path@host[+host]]
    - Mapeamento de ID de usuário
        - nfsd baseia seu controle de acesso a arquivos na máquina servidor no uid e GID fornecidos em cada pedido NFS RPC. O comportamento normal que um usuário esperaria é que ela pode acessar seus arquivos no servidor da mesma maneira que ela faria em um sistema de arquivos normal. Isso exige que os mesmos uids e gids são usados no cliente e o computador servidor. Isso nem sempre é verdade, nem sempre é desejável.

        - Muitas vezes, não é desejável que o usuário root em uma máquina cliente também é tratado como root ao acessar arquivos no servidor NFS. Para este fim, uid 0 é normalmente mapeado para uma ID de diferente: o chamado uid anônimo ou ninguém. Esse modo de operação (chamado de "root squashing") é o padrão, e pode ser desligado com no_root_squash.
        - Por padrão, exportfs escolhe um uid e gid de 65534 para acesso esmagado. Esses valores também podem ser substituídos pelas opções anonuid e anongid. Finalmente, Pode-se mapear todas as solicitações de usuários para o uid anônimo, especificando a opção all_squash.
        - Aqui está a lista completa das opções de mapeamento:
            - root_squash
            - no_root_squash
            - all_squash
            - anonuid and anongid
- Cliente [4]
    - Opções para todas as versões
        - soft / hard
        - timeo=<n>
        - retrans=<n>
        - resize=<n>
        - wsize=<n>
        - ac / noac
        - acregmin=<n>
        - acregmax=<n>
        - acdirmin=<n>
        - acdirmax=<n>
        - actimeo=<n>
        - bg / fg
        - retry=<n>
        - sec=<modo>
        - sharecache / nosharecache
        - resvport / noresvport
        - lookupcache=<modo>
    - Opções para uso nas versões 2 e 3
        - proto=<netid>
        - udp
        - tcp
        - rdma
        - port=<n>
        - mountport=<n>
        - mountproto=<netid>
        - mounthost=<nome>
        - mountvers=<n>
        - namelen=<n>
        - nfsvers=<n>
        - vers=<n>
        - lock / nolock
        - intr / nointr

        - cto / noct
        - acl / noacl
        - rdirplus / nordirplus
        - local_lock=<mecanismo>

TCP Wrappers
- Serviço
    - rpcbind

Consciência do NFSv4
- NFSv4 [5]
    - NFS versões 2 e 3 são protocolos sem estado, mas o NFS versão 4 introduz estado. Um cliente NFS versão 4 usa o estado para notificar um servidor NFS versão 4 de suas intenções em um arquivo: bloqueio, leitura, escrita, e assim por diante. Um servidor NFS versão 4 pode retornar informações para um cliente sobre quais outros clientes têm intenções em um arquivo, para permitir que um cliente faça cache de dados do arquivo de forma mais agressiva via delegação. Para ajudar a manter o estado consistente, mecanismos mais sofisticados de recuperação do cliente e reinicialização do servidor são construídos para o protocolo NFS versão 4.
    - O NFS versão 4 introduz suporte para o bloqueio de intervalo de bytes e reserva partes. Bloqueio em NFS versão 4 é baseado em contrato de locação, de modo que um cliente NFS versão 4 deve manter contato com um servidor NFS versão 4 para continuar ampliando suas locações de abertura e bloqueio.
    - O NFS versão 4 introduz delegação de arquivo. Um servidor NFS versão 4 pode permitir um cliente NFS versão 4 acessar e modificar um arquivo em seu próprio cache sem enviar quaisquer pedidos de rede para o servidor, até que o servidor indique através de uma chamada de retorno que um outro cliente deseja acessar um arquivo. Isso reduz a quantidade de tráfego entre cliente e servidor NFS versão 4 consideravelmente nos casos em que não há outros clientes que desejam acessar um conjunto de arquivos simultaneamente.
    - O NFS versão 4 usa RPCs compostos. Um cliente NFS versão 4 pode combinar várias operações tradicionais NFS (LOOKUP, OPEN, e READ, por exemplo) em uma única solicitação RPC para realizar uma operação complexa em uma ida e volta de rede.
    - O NFS versão 4 especifica uma série de mecanismos de segurança sofisticados, e reforça a sua aplicação por todos os clientes conformes. Esses mecanismos incluem Kerberos 5 e SPKM3, além da segurança AUTH_SYS tradicional. Uma nova API é fornecida para permitir a fácil adição de novos mecanismos de segurança no futuro.
    - O NFS versão 4 padroniza o uso e interpretação de ACLs em ambientes POSIX e Windows. Ele também suporta atributos nomeados. Informação do utilizador e do grupo é armazenado sob a forma de strings, não como valores numéricos. ACLs, nomes de usuário, nomes de grupos e nomes de atributos são armazenados com codificação UTF-8.
    - O NFS versão 4 combina os protocolos NFS diferentes (estatísticas, NLM, de montagem, ACL, e NFS) em uma única especificação de protocolo para permitir uma melhor compatibilidade com firewalls de rede.
    - O NFS versão 4 introduz o suporte de protocolo para migração de arquivos e replicação.
    - O NFS versão 4 requer suporte de RPC sobre protocolos de transporte de rede de streaming, como o TCP. Embora muitos clientes NFS versão 4 continuem a suportar RPC

via datagramas, esse suporte pode ser aplicado gradualmente ao longo do tempo a favor de protocolos de transporte de fluxo mais confiáveis.

Termos e utilitários

- /etc/exports - (5) tabela de exportação do servidor NFS
- exportfs - (8) matem a tabela de sistema de arquivos exportados por NFS
- showmount - (8) mostra informações de montagem para um servidor NFS
- nfsstat - (8) lista estatísticas NFS
- /proc/mounts - lista de sistemas de arquivos montados
- /etc/fstab - (5) informação estática sobre sistema de arquivos
- rpcinfo - (8) relata informações do RPC
- mountd - rpc.mountd(8) daemon de montagem NFS
- portmapper - rpcbind(8) mapeador universal de endereço para número de programa RPC

Referências

1. http://en.wikipedia.org/wiki/Network_File_System
2. http://en.wikipedia.org/wiki/Portmap
3. exports(5)
4. nfs(5)
5. http://nfs.sourceforge.net/

Exercícios práticos

1. Preparação para exercícios
    a. Criar o diretório /mnt/nfs (Ex.: mkdir <diretório>)

    b. Iniciar o serviço nfslock (se estiver parado) (Ex.: service <serviço> start)

    c. Iniciar o serviço nfs (Ex.: service <serviço> start)

2. Arquivos de configuração do NFSv3
    a. Verificar o conteúdo do arquivo de exportação do servidor NFS (Ex.: cat <arquivo>)

    b. Definir no arquivo de exportação do servidor NFS (Ex.: vi <arquivo>)
        i. Diretório /tmp
            1. Endereço: 127.0.0.1
            2. Opções: síncrono

3. Termos e utilitários NFS
    a. Recarregar a tabela de exportações do NFS utilizando o comando exportfs (Ex.: exportfs <opções>)

    b. Verificar os diretórios exportados pelo servidor NFS, através do comando exportfs (Ex.: exportfs)

c. Verificar os diretórios exportados pelo servidor NFS, através do comando exportfs, exibindo as opções de exportação (Ex.: exportfs <opções>)

d. Verificar o mapeamento de número de programas RPC para portas (Ex.: rpcinfo)

e. Verificar o mapeamento de número de programas RPC para portas, no endereço IP da segunda interface (Ex.: rpcinfo <opções>)

f. Verificar quais sistemas de arquivos exportados no servidor local estão montados em clientes NFS (Ex.: showmount <opções>)

g. Exibir somente as estatísticas do servidor NFS (Ex.: nfsstat <opções>)

4. Restrições de acesso para certos hosts e subredes
   a. Tentar montar o diretório /tmp exportado no servidor NFS local, no diretório /mnt/nfs, usando o protocolo NFSv3, através do IP da segunda interface (Ex.: mount <opções>)

   b. Adicionar a definição no arquivo de exportação do servidor NFS (Ex.: vi <arquivo>)
      i. Diretório /tmp
         1. Endereço: <IP da segunda interface>
         2. Opções: síncrono

   c. Verificar os diretórios exportados pelo servidor NFS, através do comando exportfs, exibindo as opções de exportação (Ex.: exportfs <opções>)

   d. Recarregar a tabela de exportações do NFS utilizando o comando exportfs (Ex.: exportfs <opções>)

   e. Verificar os diretórios exportados pelo servidor NFS, através do comando exportfs, exibindo as opções de exportação (Ex.: exportfs <opções>)

   f. Montar o diretório /tmp exportado no servidor NFS local, no diretório /mnt/nfs, usando o protocolo NFSv3, através do IP da segunda interface (Ex.: mount <opções>)

5. Opções de montagem no servidor e cliente
   a. Varrer o conteúdo do diretório /mnt/nfs (Ex.: find <diretório>)

   b. Alterar a definição no arquivo de exportação do servidor NFS (Ex.: vi <arquivo>)
      i. Diretório /tmp
         1. Endereço: <IP da segunda interface>
         2. Opções: síncrono, sem esmagar root

   c. Recarregar a tabela de exportações do NFS utilizando o comando exportfs (Ex.: exportfs <opções>)

   d. Varrer o conteúdo do diretório /mnt/nfs (Ex.: find <diretório>)

   e. Tentar criar o arquivo vazio /mnt/nfs/temp (Ex.: touch <arquivo>)

f. Alterar a definição no arquivo de exportação do servidor NFS (Ex.: vi <arquivo>)
    i. Diretório /tmp
        1. Endereço: <IP da segunda interface>
        2. Opções: síncrono, sem esmagar root, leitura e escrita

g. Recarregar a tabela de exportações do NFS utilizando o comando exportfs (Ex.: exportfs <opções>)

h. Criar o arquivo vazio /mnt/nfs/temp (Ex.: touch <arquivo>)

i. Desmontar o diretório /mnt/nfs (Ex.: umount <diretório>)

j. Montar o diretório /tmp exportado no servidor NFS 127.0.0.1, no diretório /mnt/nfs, usando o protocolo NFSv3, com a opção sem acl (Ex.: mount <opções>)

6. TCP Wrappers
    a. Desmontar o diretório /mnt/nfs (Ex.: umount <diretório>)

    b. No arquivo de negação do TCP Wrappers, negar acesso ao serviço rpcbind, para todos os endereços (Ex.: vi <arquivo>)

    c. Tentar montar o diretório /tmp exportado no servidor NFS local, no diretório /mnt/nfs, usando o protocolo NFSv3, através do IP da segunda interface (Ex.: mount <opções>)

Simulado

1. … é o arquivo que define quais são os diretórios exportados por NFS em um sistema.

2. No arquivo de exportação do NFS, para um mesmo diretório exportado, é suportado diferentes opções para diferentes hosts.
    a. V
    b. F

3. No arquivo fstab, o formato montagem de um diretório exportado por NFS é <servidor>:<caminho> <ponto de montagem> <tipo de sistema de arquivos> <opções> <dump> <fsck>.
    a. V
    b. F

4. O comando … é usado para verificar o mapeamento de portas por número de programa RPC.

5. O serviço portmapper, também chamado de portmap ou rpcbind, é responsável por mapear o número de programa RPC para uma porta de rede.
    a. V
    b. F

6. O comando exportfs deve ser usado em um servidor NFS para visualizar a tabela de exportações, enquanto o comando showmount deve ser usado em um cliente NFS para visualizar os sistemas de arquivos importados.
    a. V
    b. F

7. No arquivo de exportações NFS, a string “*.example.com” declara que todos os hosts do domínio e subdomínios do example.com, podem acessar aquele diretório exportado.
    a. V
    b. F

8. No arquivo de exportações NFS, a string “192.168.” declara que todos os endereços começados por “192.168.”, podem acessar aquele diretório exportado.
    a. V
    b. F

9. Para garantir que o usuário root do cliente NFS tenha acesso aos arquivos exportados do usuário root do servidor NFS, o parâmetro … deve ser usado.

10. A opção rw é padrão para sistemas de arquivos exportados por NFS.
    a. V
    b. F

11. A opção de montagem … define a versão 3 do protocolo NFS.

12. A opção … faz com que não haja atrasados durante a comissão de dados no dispositivo de armazenamento.

13. O serviço … pode ser usado através do TCP Wrappers, para controlar acesso ao NFS v3.

14. O NFSv4 possui integração com autenticação Kerberos.
    a. V
    b. F

15. O NFSv4 possui servidor orientado a estado.
    a. V
    b. F

Tópico 210: Gerenciamento de Cliente de Rede

210.1 Configuração DHCP

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de configurar um servidor DHCP. Esse objetivo inclui definindo opções padrão e por cliente, adicionando hosts estáticos e hosts BOOTP. Também está incluído configurando um agente de relay DHCP e mantendo um servidor DHCP.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários DHCP
- Subredes e configuração de alocação dinâmica de faixa

Termos e utilitários:

- dhcpd.conf
- /var/log/daemon.log e /var/log/messages
- dhcpd.leases
- arp
- dhcpd

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários DHCP

- DHCP [1]
    - O Dynamic Host Configuration Protocol (DHCP) é um protocolo de rede padronizado usado em redes IP (Internet Protocol) para distribuir dinamicamente os parâmetros de configuração de rede, como endereços IP para interfaces e serviços. Com o DHCP, computadores solicitam endereços IP e parâmetros de rede automaticamente de um servidor DHCP, reduzindo a necessidade de um administrador de rede ou um usuário, para definir essas configurações manualmente.
    - Visão geral
        - Os computadores usam o Dynamic Host Configuration Protocol para o pedido de parâmetros de protocolo de Internet, a partir de um servidor de rede, como um endereço IP. O protocolo opera com base no modelo cliente-servidor. O DHCP é muito comum em todas as redes modernas que variam em tamanho, de redes domésticas à grandes redes de campus e redes regionais de provedores de serviços de Internet. A maioria dos roteadores de rede residenciais recebem um endereço IP global exclusivo dentro da rede do provedor. No interior de uma rede local, o DHCP atribui um endereço de IP local para dispositivos ligados à rede local.
        - Quando um computador ou outro dispositivo de rede, se liga a uma rede, o software de cliente DHCP no seu sistema operativo, envia uma transmissão de consulta solicitando informação necessária. Qualquer servidor DHCP na rede pode atender à solicitação. O servidor DHCP gerencia um pool de endereços IP e informações sobre os parâmetros de configuração do cliente, tais como gateway padrão, nome de domínio, os servidores de nomes, e servidores de tempo. Ao receber um pedido, o servidor pode responder com informações específicas para cada cliente, conforme configurado anteriormente por um administrador, ou com um endereço específico e

qualquer outra informação válida para toda a rede, e o período de tempo durante o qual a atribuição (concessão) é válida. Um host tipicamente consulta para obter essas informações imediatamente após a inicialização, e depois periodicamente antes da expiração da informação. Quando uma tarefa é atualizada pelo computador do cliente, ele inicialmente solicita os mesmos valores de parâmetro, mas pode ser atribuído um novo endereço a partir do servidor, com base nas políticas de atribuição definidos pelos administradores.

- Em grandes redes que consistem em múltiplos links, um único servidor DHCP pode atender toda a rede quando auxiliados por agentes de retransmissão DHCP localizados nos roteadores comunicantes. Tais agentes retransmitem mensagens entre clientes e servidores DHCP localizados em diferentes sub-redes.
- Dependendo da implementação, o servidor DHCP pode ter três métodos de alocação de endereços IP:
  - Alocação dinâmica: um administrador de rede reserva um intervalo de endereços IP para DHCP, e cada computador cliente na rede local está configurado para solicitar um endereço IP a partir do servidor DHCP durante a inicialização de rede. O processo de solicitação e permissão usa um conceito de concessão com um período de tempo controlável, permitindo o servidor DHCP recuperar (e, em seguida, realocar) endereços IP que não são renovados
  - Alocação automática: o servidor DHCP atribui um endereço IP permanente para um cliente solicitando a partir do intervalo definido pelo administrador. Isso é como a atribuição dinâmica, mas o servidor DHCP mantém uma tabela das últimas de atribuições de endereços IP, de modo que ele pode preferencialmente atribuir a um cliente o mesmo endereço de IP que o cliente tinha anteriormente.
  - Alocação estática: o servidor DHCP atribui um endereço IP com base em um mapeamento pré-configurado para o endereço MAC de cada cliente. Esse recurso é variadamente chamado de atribuição DHCP estático pelo DD-WRT, endereço fixo pela documentação do dhcpd, reserva de endereço pelo Netgear, reserva de DHCP ou DHCP estático pela Cisco e Linksys, e reserva de endereço IP ou endereço MAC/IP de ligação por vários outros fabricantes de roteadores.
- O DHCP é usado para o Internet Protocol versão 4 (IPv4), bem como IPv6. Enquanto ambas as versões têm a mesma finalidade, os detalhes do protocolo para IPv4 e IPv6 são suficientemente diferentes para que possam ser considerados protocolos separados. Para a operação IPv6, dispositivos podem, alternativamente, usar autoconfiguração de endereço sem estado. Hosts IPv4 também podem usar endereçamento local vinculado para atingir o funcionamento restrito à ligação de rede local.

- História
  - Em 1984, o Reverse Address Resolution Protocol (RARP), definido na RFC 903, foi introduzido para permitir que dispositivos simples, como estações de trabalho sem disco, obtessem dinamicamente um endereço IP adequado. No entanto, porque ele agiu na camada de enlace de dados, dificultou a implementação em diversas plataformas de servidor, e também requeriu que um servidor estivesse presente em cada link de rede individual. Logo depois, ele foi substituído pelo "Protocolo Bootstrap" (BOOTP) definido na RFC 951. Essa introduziu o conceito de um agente

de retransmissão, o que permitiu o encaminhamento de pacotes BOOTP em redes, permitindo um servidor BOOTP central servir hosts em muitas sub-redes IP.
        - O DHCP é baseado no BOOTP, mas pode alocar dinamicamente endereços IP a partir de um pool e recuperá-los quando eles não estão mais em uso. Ele também pode ser usado para fornecer uma ampla variedade de parâmetros de configuração adicionais para clientes IP, incluindo os parâmetros específicos da plataforma. Foi definido primeiro na RFC 1531 em Outubro de 1993; mas devido a erros no processo editorial foi quase que imediatamente reeditado como RFC 1541.
        - Quatro anos mais tarde o tipo de mensagem DHCPINFORM e outras pequenas mudanças foram adicionadas pela RFC 2131; que a partir de 2014 continua a ser o padrão para redes IPv4.
        - O DHCPv6 foi inicialmente descrito pela RFC 3315 em 2003, mas essa foi atualizada por muitas RFCs subseqüentes. A RFC 3633 acrescentou um mecanismo para prefixo de delegação DHCPv6, e a autoconfiguração de endereço sem estado foi adicionado pela RFC 3736.
- dhcpd [2] [3]
    - dhcpd (uma abreviação para "daemon DHCP") é o nome de um programa que funciona como um daemon em um servidor para fornecer o serviço Dynamic Host Configuration Protocol (DHCP) para uma rede.
    - Os clientes podem solicitar um endereço IP (IP) a partir de um servidor DHCP quando eles precisam de um. O servidor DHCP, em seguida, oferece o "leasing" (concessão) de um endereço IP para o cliente, o qual é livre para solicitar ou ignorar. Se o cliente o solicite e o servidor reconhece, então o cliente está autorizado a utilizar esse endereço IP para o "tempo de concessão" especificado pelo servidor. Em algum momento antes de expirar o contrato de locação, o cliente deve re-pedir o mesmo endereço IP se ele quer continuar a usá-lo.
    - Endereços IP emitidos são rastreados pelo dhcpd através de um registro no arquivo dhcpd.leases. Isso permite o servidor manter o estado sobre reinícios do serviço de dhcp, que ao contrário, poderiam levar à duplicação de endereços IP a serem emitidos quando o servidor emitir o mesmo IP novamente enquanto outro cliente ainda tem o direito de usá-lo.
    - Essa implementação de referência de DHCP é desenvolvido pelo Internet Systems Consortium e é suportada em Linux, Mac OS X, FreeBSD, Solaris, AIX e HP-UX.
    - O acesso remoto a uma instância em execução do dhcpd é fornecida pelo Application Programming Object Management Interface (OMAPI). No lado do servidor, essa interface permite a edição de informações de registro para nós gerenciados. Usos no cliente incluem buscar informações de configuração, liberar e renovar contratos de concessão, e mudando as interfaces que são gerenciados pelo cliente DHCP.
    - Operação
        - Na inicialização, o dhcpd lê o arquivo dhcpd.conf e armazena uma lista de endereços disponíveis em cada subrede na memória. Quando um cliente solicita um endereço usando o protocolo DHCP, o dhcpd atribui um endereço para ele. Cada cliente é atribuído uma concessão, que expira depois de um período de tempo escolhido pelo administrador (por padrão, um dia). Antes das concessões expirarem, os clientes em as concessões são atribuídas, são esperados para renová-las, a fim de continuar a usarem os endereços. Uma vez que uma concessão expirou, o cliente a que essa concessão foi atribuída, já não está autorizado a utilizar o endereço IP concedido.

- A fim de manter o controle das concessões em toda a reinicialização do sistema e reinicio do servidor, o dhcpd mantém uma lista de concessões que atribuiu, no arquivo dhcpd.leases(5). Antes do dhcpd conceder uma concessão a um host, ele registra a concessão nesse arquivo e garante que o conteúdo do arquivo é gravado para o disco. Isso garante que mesmo em caso de uma falha no sistema, o dhcpd não irá esquecer-se sobre uma concessão que tenha atribuído. Na inicialização, depois de ler o arquivo dhcpd.conf, o dhcpd lê o arquivo dhcpd.leases para refrescar sua memória sobre que concessões foram atribuídas.
- Novas concessões são acrescentadas ao final do arquivo dhcpd.leases. A fim de impedir que o arquivo torne-se arbitrariamente grande, de tempos em tempos o dhcpd cria um novo arquivo dhcpd.leases, a partir do banco de dados de concessão, interno do seu núcleo. Uma vez que esse arquivo foi gravado no disco, o arquivo antigo é renomeado dhcpd.leases~, e o novo arquivo é renomeado dhcpd.leases. Se o sistema falhar no meio deste processo, qualquer arquivo dhcpd.leases que restar, conterá todas as informações de concessão, e por isso, não há necessidade de um processo de recuperação de falhas especial.
- O suporte BOOTP também é fornecido por esse servidor. Ao contrário do DHCP, o protocolo BOOTP não fornece um protocolo para a recuperação de endereços atribuídos dinamicamente uma vez que eles não são mais necessários. Ainda é possível atribuir dinamicamente endereços a clientes BOOTP, mas algum processo administrativo para recuperar endereços é necessário. Por padrão, as concessões são concedidas a clientes BOOTP em perpetuidade, embora o administrador da rede pode definir uma data de corte mais cedo ou um comprimento mais curto para concessão BOOTP, se isso faz sentido.
- Clientes BOOTP também pode ser servidos no modo padrão antigo, que é simplesmente apresentar uma declaração no arquivo dhcpd.conf para cada cliente BOOTP, permanentemente atribuindo um endereço para cada cliente.
- Sempre que as alterações são feitas no arquivo dhcpd.conf, o dhcpd deve ser reiniciado. Para reiniciar dhcpd, envie um SIGTERM (sinal 15) para a identificação do processo contido no /var/run/dhcpd.pid, e então re-invoque o dhcpd. Porque o banco de dados do servidor DHCP não é tão leve quanto um banco de dados BOOTP, o dhcpd não reinicia-se automaticamente quando se vê uma alteração no arquivo dhcpd.conf.

- Subredes
  - O dhcpd precisa saber os números de sub-rede e máscaras de rede de todas as sub-redes para o qual ele estará fornecendo serviço. Além disso, a fim de alocar dinamicamente endereços, deve ser atribuído um ou mais intervalos de endereços em cada subrede, que por sua vez pode atribuir ao hosts clientes ao iniciarem. Assim, uma configuração muito simples prestando suporte DHCP, pode ter essa aparência:
    - subnet 239.252.197.0 netmask 255.255.255.0 {
    - range 239.252.197.10 239.252.197.250;
    - }
  - Várias faixas de endereços podem ser especificadas desse jeito:
    - subnet 239.252.197.0 netmask 255.255.255.0 {
    - range 239.252.197.10 239.252.197.107;
    - range 239.252.197.113 239.252.197.250;
    - }

        - Se uma subrede só será provida com o serviço BOOTP e sem atribuição de endereços dinâmicos, a cláusula de faixa pode ser deixada de fora por completo, mas a declaração de subrede deve aparecer.
    - Comprimentos de Concessões
        - Concessões de DHCP podem ser atribuídas com quase qualquer comprimento, de zero segundos para o infinito. O comprimento de concessão que faz sentido para uma determinada subrede, ou para qualquer instalação dada, irá variar dependendo dos tipos de hosts sendo servidos.
        - Por exemplo, em um ambiente de escritório, onde os sistemas são adicionados de tempos em tempos e removidos de tempos em tempos, mas movem relativamente com pouca freqüência, pode fazer sentido permitir tempos de concessão de um mês de mais. Em um ambiente de teste final em um chão de fábrica, pode fazer mais sentido atribuir um comprimento de concessão máximo de 30 minutos - tempo suficiente para passar por um procedimento de teste simples em um dispositivo de rede, antes de o embalar para a entrega.
        - É possível especificar dois comprimentos de concessão : o comprimento padrão que será atribuído se o cliente não requerer qualquer período de concessão particular, e um comprimento máximo de concessão. Esses são especificados como cláusulas para o comando de subrede:
            - subnet 239.252.197.0 netmask 255.255.255.0 {
            - range 239.252.197.10 239.252.197.107;
            - default-lease-time 600;
            - max-lease-time 7200;
            - }
        - Essa declaração de subrede particular, especifica um tempo de concessão padrão de 600 segundos (dez minutos) e um tempo máximo de concessão de 7200 segundos (duas horas). Outros valores comuns seria 86400 (um dia), 604800 (uma semana) e 2592000 (30 dias).
        - Cada subrede não precisa ter a mesma concessão - no caso de um ambiente de escritório e um ambiente de produção, servidos pelo mesmo servidor DHCP, pode fazer sentido ter valores demasiadamente discrepantes para tempos padrão e máximos de concessão em cada subrede.
    - Suporte BOOTP
        - Cada cliente BOOTP deve ser expressamente declarado no arquivo dhcpd.conf. Uma declaração de cliente muito básica irá especificar o endereço de hardware da interface de rede do cliente, e o endereço IP para atribuir a esse cliente. Se o cliente precisa ser capaz de carregar um arquivo de inicialização do servidor, o nome do arquivo deve ser especificado. A declaração cliente bootp simples pode ter esta aparência:
            - host haagen {
            - hardware ethernet 08:00:2b:4c:59:23;
            - fixed-address 239.252.197.9;
            - filename "/tftpboot/haagen.boot";
            - }
    - Opções
        - O DHCP (e também BOOTP com Vendor Extensions) fornecem um mecanismo pelo qual o servidor pode fornecer ao cliente informações sobre como configurar sua interface de rede (por exemplo, a máscara de subrede), e também como o cliente

pode acessar vários serviços de rede (por exemplo, DNS, roteadores IP, e assim por diante).
- Essas opções podem ser especificadas em uma base por subrede e, para os clientes BOOTP, também em uma base por cliente. No caso em que uma declaração de cliente BOOTP especifica as opções que também são especificadas na sua declaração de subrede, as opções especificadas na declaração do cliente têm precedência. Uma configuração DHCP razoavelmente completa poderia ser algo como isso:
  - subnet 239.252.197.0 netmask 255.255.255.0 {
  - range 239.252.197.10 239.252.197.250;
  - default-lease-time 600 max-lease-time 7200;
  - option subnet-mask 255.255.255.0;
  - option broadcast-address 239.252.197.255;
  - option routers 239.252.197.1;
  - option domain-name-servers 239.252.197.2, 239.252.197.3;
  - option domain-name "isc.org";
  - }
- Um host bootp nessa subrede que precisa estar em um domínio diferente e usar um servidor de nome diferente, pode ser declarado como segue:
  - host haagen {
  - hardware ethernet 08:00:2b:4c:59:23;
  - fixed-address 239.252.197.9;
  - filename "/tftpboot/haagen.boot";
  - option domain-name-servers 192.5.5.1;
  - option domain-name "vix.com";
  - }
- Uma descrição mais completa da sintaxe do arquivo dhcpd.conf é fornecido em dhcpd.conf(5).

- OMAPI
  - O servidor DHCP fornece a capacidade de modificar algumas das sua configuração enquanto estiver em execução, sem interrompê-lo, modificando seus arquivos de banco de dados, e o reiniciando. Essa capacidade é atualmente assegurada usando OMAPI - uma API para manipular objetos remotos. Clientes OMAPI se conectam ao servidor usando TCP/IP, autenticam e então podem examinar o status atual do servidor e fazer alterações nele.
  - Em vez de implementar o protocolo OMAPI subjacente diretamente, programas de usuário devem usar a API do dhcpctl ou o próprio OMAPI. O dhcpctl é um wrapper que lida com algumas das tarefas domésticas que OMAPI não faz automaticamente. O dhcpctl e OMAPI estão documentados no dhcpctl(3) e omapi(3).
  - O OMAPI exporta objetos que, podem então, ser examinados e modificados. O servidor DHCP exporta os seguintes objetos: concessão, host, estado de failover e grupo. Cada objeto tem um certo número de métodos que são fornecidos: pesquisar, criar e destruir. Além disso, é possível olhar para os atributos que estão armazenados nos objetos, e em alguns casos, modificar esses atributos.

- Arquivos de configuração
  - dhcpd.conf(5) - arquivo de configuração do dhcpd
- Utilitários
  - dhcpd(8) - Servidor de Protocolo de Configuração de Host Dinâmico

        - Exemplos:
            - dhcpd - inicia o servidor dhcp em IPv6
            - dhcpd -4 - inicia o servidor dhcp em IPv4
            - dhcpd -6 - inicia o servidor dhcp em IPv6
            - dhcpd -p <porta> - especifica a porta que o serviço irá usar
            - dhcpd -f - executa o servidor em primeiro plano
            - dhcpd -d - exibe o log na saída de erro padrão
            - dhcpd -cf <arquivo> - especifica o arquivo de configuração
            - dhcpd -lf <arquivo> - especifica o arquivo de concessão
            - dhcpd -t - valida a configuração
            - dhcpd -T - valida o arquivo de concessão
            - dhcpd -user <usuário> - especifica o usuário que executará o servidor
            - dhcpd -group <grupo> - especifica o grupo que executará o servidor
            - dhcpd -chroot <diretório> - especifica o diretório para ser usado como jaula
    - arp(8) - manipula o cache ARP de sistema
        - Exemplos:
            - arp - exibe a tabela de cache de ARP do sistema
            - arp -v - modo verbose
            - arp -n - exibe endereços numéricos
            - arp -a <host> - exibe todas as entradas para o host especificado
            - arp -d <host> - remove todas as entradas do host especificado
            - arp -s <host> <mac> - define o endereço mac para o host especificado

Subredes e configuração de alocação dinâmica de faixa

- Configuração de rede [3]
    - Através do bloco subnet
    - Formato:
        - subnet <endereço de rede> netmask <máscara de rede> {
        - <parâmetros específicos da subrede>
        - }
- Configuração de faixa [3]
    - Através dos parâmetros range e range6
    - Para qualquer subrede onde endereços serão distribuídos dinamicamente, deve existir pelo menos uma declaração de faixa
    - A opção dynamic-bootp é usada para atribuir endereço a clientes bootp reconhecidos (bloco host)
    - Formato:
        - range <endereço inicial> <endereço final>
        - range dynamic-bootp <endereço inicial> <endereço final>
- Exemplos:
    - subnet 204.254.239.0 netmask 255.255.255.224 {
    - <parâmetros específicos da subrede>
    - range 204.254.239.10 204.254.239.30;
    - }
    - subnet 204.254.239.32 netmask 255.255.255.224 {
    - <parâmetros específicos da subrede>
    - range dynamic-bootp 204.254.239.42 204.254.239.62;
    - }

    - subnet 204.254.239.64 netmask 255.255.255.224 {
    - <parâmetros específicos da subrede>
    - range 204.254.239.74 204.254.239.94;
    - }

Hosts e configuração de alocação estática

- Configuração por clientes [3]
    - Através do bloco host e do parâmetro hardware
    - A opção hardware é necessária para identificar o host
    - Formato:
        - host <nome> {
        - hardware ethernet <endereço MAC>
        - <parâmetros específicos do cliente>
        - }
- Alocação estática por cliente [3]
    - Através dos parâmetros fixed-address e fixed-address6
    - Clientes com endereços fixos são capazes de iniciar o sistema usando bootp ou dhcp
    - Formato:
        - fixed-address <endereço>
- Exemplos:
    - host cl01 {
    - hardware ethernet 07:01:07:26:c1:b7;
    - <parâmetros específicos do cliente>
    - fixed-address 192.168.10.10;
    - }
    - host cl02 {
    - hardware ethernet 07:01:07:26:a6:41;
    - <parâmetros específicos do cliente>
    - fixed-address 192.168.10.11;
    - }
    - host cl03 {
    - hardware ethernet 07:01:07:26:b5:c7;
    - <parâmetros específicos do cliente>
    - fixed-address 192.168.10.12;
    - }

Uso do serviço como agente de encaminhamento

- Através do utilitário dhcrelay
    - dhcrelay(8) - Agente de Encaminhamento de Protocolo de Configuração de Host Dinâmico
    - Exemplos:
        - dhcrelay <servidor> - encaminha as requisições da subrede local para o servidor especificado
        - dhcrelay -4 - usa o agente em IPv4 (padrão)
        - dhcrelay -6 - usa o agente em IPv6
        - dhcrelay -c <número> - máximo número de saltos
        - dhcrelay -d - força o agente a executar como processo em primeiro plano
        - dhcrelay -q - modo silencioso

Termos e utilitários

- dhcpd.conf - (5) arquivo de configuração do dhcpd
- dhcpd.leases - (5) banco de dados de concessão de cliente DHCP
- /var/log/daemon.log e /var/log/messages
    - /var/log/daemon.log - arquivo de log para daemons (Debian e derivados)
    - /var/log/messages - arquivo de log geral do sistema (RH e derivados)
- arp - (8) manipula o cache ARP de sistema
- dhcpd - (8) Servidor de Protocolo de Configuração de Host Dinâmico

Referências

1. http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
2. https://en.wikipedia.org/wiki/DHCPD
3. dhcpd.conf(5)

Simulado

1. … é o arquivo de configuração do servidor dhcp.

2. O serviço DHCP permite que um dispositivo de rede obtenha configurações de forma dinâmica, sem a necessidade de um administrador ou usuário definir manualmente essas configurações.
    a. V
    b. F

3. A diferença entre entrega de endereços dinâmico e automático é que o segundo sempre entrega novos endereços para os clientes DHCP.
    a. V
    b. F

4. O protocolo DHCP é baseado no protocolo ….

5. … é o arquivo padrão usado como base de dados do servidor dhcp.

6. O serviço dhcp pode contar com agentes de retransmissão, que encaminham o pedido original de um segmento de rede, a um servidor de dhcp remoto.
    a. V
    b. F

7. Através do comando …, é possível manipular a tabela cache de endereços mac do sistema operacional.

8. A diretiva subnet <endereço> { … } é usada para definir um bloco de definições para uma subrede.
    a. V
    b. F

9. A diretiva … é usada para definir um intervalo de endereços IP.

10. O comando … é usado para iniciar o agente de encaminhamento de requisições DHCP e BOOTP.

210.2 Autenticação por PAM

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de configurar o PAM para suportar autenticação usando vários métodos disponíveis.
Áreas de conhecimentos chave:

- Arquivos de configuração, termos e utilitários PAM
- Senhas passwd e shadow

Termos e utilitários:

- /etc/pam.d/
- pam.conf
- nsswitch.conf
- pam_unix, pam_cracklib, pam_limits, pam_listfile

Áreas de conhecimentos chave

Arquivos de configuração, termos e utilitários PAM

- PAM [1]
    - Um módulo de autenticação conectável (PAM) é um mecanismo para integrar múltiplos sistemas de autenticação de baixo nível em uma interface de programação de aplicativo de alto nível (API). Ele permite que programas que dependem de autenticação, sejam escritos de forma independente do esquema de autenticação subjacente. O PAM foi proposto pela primeira vez pela Sun Microsystems na Requisição para Comentários (RFC) 86.0 da Open Software Foundation, datada de Outubro de 1995. Foi adotada como a estrutura de autenticação do Ambiente de Desktop Comum. Como uma infra-estrutura de código aberto stand-alone, o PAM apareceu pela primeira vez no Red Hat Linux 3.0.4 em Agosto de 1996. O PAM é atualmente suportado no sistema operacional AIX, DragonFly BSD, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD e Solaris.
    - Uma vez que não exite nenhuma norma central do comportamento PAM, houve uma tentativa posterior de padronizar o PAM como parte do processo de normalização X/Open UNIX, resultando no padrão X/Open Sign-on (XSSO). Essa norma não foi ratificada, mas o projeto de norma tem servido como um ponto de referência para implementações PAM posteriores (por exemplo, OpenPAM).
    - Críticas ao PAM
        - Como a maioria das implementações do PAM não fazem interface com clientes remotos por si mesmo, o PAM por si só não pode implementar o Kerberos, o tipo mais comum de SSO usado em ambientes Unix. Isso levou à incorporação do SSO como a porção "de autenticação primária" da norma que pretendia ser o XSSO e o advento de tecnologias como SPNEGO e SASL. Essa falta de funcionalidade é também a razão que o SSH faz a sua própria negociação de mecanismo de autenticação.
        - Na maioria das implementações do PAM, o pam_krb5 apenas busca TGTs, que envolve solicitar as credenciais do usuário e são usados somente para login inicial em um ambiente SSO. Para buscar uma permissão de serviço para uma

determinada aplicação, e não solicitar ao usuário que digite as credenciais de novo, aquela aplicação deve ser especificamente codificada para suportar Kerberos, como o pam_krb5 por si só não pode obter tickets de serviço, embora existam versões do PAM-KRB5 que estão tentando contornar o problema.

- Linux PAM [2] [3]
    - Linux Pluggable Authentication Modules (PAM) provê autenticação dinâmica para aplicações e serviços em um sistema Linux. O Linux PAM é evoluido a partir da arquitetura Unix PAM.
    - Configuração
        - Quando um aplicativo de concessão de privilégio ciente PAM é iniciado, ele ativa a sua ligação ao PAM-API. Essa ativação executa uma série de tarefas, sendo a mais importante a leitura do arquivo de configuração: /etc/pam.conf. Alternativamente, esse pode ser o conteúdo do diretório /etc/pam.d/. A presença desse diretório fará o Linux-PAM ignorar /etc/pam.conf.
        - Esses arquivos listam os PAMs que irão fazer as tarefas de autenticação exigidas por esse serviço, bem como o comportamento adequado do PAM-API no evento que PAMs individuais falham.
        - A sintaxe do arquivo de configuração /etc/pam.conf é a seguinte. O arquivo é composto de uma lista de regras, cada regra é normalmente colocado em uma única linha, mas pode ser estendida com um fim de linha escapado: '\<LF>'. Os comentários são precedidos com marcas '#' e se estendem até o próximo fim de linha.
        - O formato de cada regra é uma coleção de tokens separados por espaço, os três primeiros sendo case-insensitive:
            - <tipo> <serviço> <controle> <caminho do módulo> <argumentos do módulo>
        - A sintaxe de arquivos contidos no diretório /etc/pam.d/, são idênticas, exceto pela ausência de qualquer campo de serviço. Neste caso, o serviço é o nome do arquivo no diretório /etc/pam.d/. Esse nome do arquivo deve estar em minúsculas.
        - Uma característica importante do PAM, é que um número de regras podem ser empilhadas, para combinar os serviços de um certo número de PAMs, para uma dada tarefa de autenticação.
        - O <serviço> é normalmente o nome familiar do respectivo pedido: login e su são bons exemplos. O nome do serviço, “other”, está reservado para dar regras padrão. Somente as linhas que mencionam o serviço atual (ou na ausência de tal, as entradas “other”) vão ser associadas com o serviço da aplicação.
        - O <tipo> é o grupo de gerenciamento que a regra corresponde. É usado para especificar quais dos grupos de gestão do módulo subsequente estão para ser associados. As entradas válidas são:
            - account - esse tipo de módulo executa gerenciamento de contas baseado em não-autenticação. Ele é geralmente usado para restringir/permitir o acesso a um serviço com base no tempo do dia, os recursos do sistema atualmente disponíveis (número máximo de usuários) ou talvez a localização do usuário recorrente - 'root' loga somente no console.
            - auth - esse tipo de módulo fornece dois aspectos da autenticação do usuário. Em primeiro lugar, estabelece que o usuário é quem diz ser, instruindo o pedido para solicitar ao usuário uma senha ou outros meios de identificação. Em segundo lugar, o módulo pode conceder a associação de grupo ou outros privilégios por meio de suas propriedades de concessão de credenciais.

- password - esse tipo de módulo é necessário para atualizar o token de autenticação associada ao usuário. Normalmente, há um módulo para cada tipo de autenticação (auth) baseado em 'desafio/resposta'.
- session - esse tipo de módulo está associado com fazer as coisas que precisam ser feitas para o usuário antes/depois que ele possa receber determinado serviço. Essas coisas incluem o registro de informações relativas à abertura/fechamento de alguma troca de dados com um usuário, os diretórios de montagem, etc.

- Se o valor do tipo da lista acima é prefixado com um caractere '-', a biblioteca PAM não irá registrar no log do sistema se não é possível carregar o módulo porque está faltando no sistema. Isso pode ser útil especialmente para módulos que nem sempre são instalados no sistema e não são necessários para autenticação e autorização correta da sessão de login.
- O terceiro campo, o <controle>, indica o comportamento do PAM-API que deve o módulo deixar de ter na sua função de sucesso de autenticação. Há dois tipos de sintaxe para esse campo de controle: um simples tem uma única palavra-chave simples; o mais complicado envolve uma seleção entre colchetes de pares valor=ação.
- Para a sintaxe simples (histórica) os valores de controle válidos são:
  - required - falha de tal PAM acabará por levar o PAM-API a retornar uma falha, mas somente depois que os módulos empilhados restantes (para este serviço e tipo) forem invocados.
  - requisite - como necessário, no entanto, no caso que um tal módulo retorna uma falha, o controle é devolvido diretamente para a aplicação. O valor de retorno é aquele associado com o primeiro módulo requerido ou requisitado a falhar. Note, esse sinalizador pode ser usado para proteger contra a possibilidade de um usuário ter a oportunidade de digitar uma senha através de um meio inseguro. É concebível que tal comportamento pode informar um atacante de contas válidas em um sistema. Essa possibilidade deve ser pesada contra as preocupações não insignificantes de expor uma senha sensível em um ambiente hostil.
  - sufficient - o sucesso de um tal módulo é suficiente para satisfazer os requisitos de autenticação da pilha de módulos (se um módulo requerido antes falhou o sucesso de um módulo presente é ignorado). Uma falha deste módulo não é considerado como fatal para satisfazer o pedido que esse tipo foi bem sucedido. Se o módulo for bem-sucedido o framework PAM retorna sucesso ao aplicativo imediatamente sem tentar quaisquer outros módulos.
  - optional - o sucesso ou o fracasso deste módulo só é importante se ele é o único módulo na pilha associado a esse tipo + serviço.
  - include - incluem todas as linhas de determinado tipo do arquivo de configuração especificado como um argumento para esse controle.
  - substack - incluem todas as linhas de determinado tipo do arquivo de configuração especificada como um argumento para esse controle. Isso é diferente de incluir, em que a avaliação das ações done e die em um subpilha não causam pular o resto da pilha do módulo completo, mas apenas da subpilha. Saltos em um subpilha também não podem fazer salto de avaliação fora dela, e toda a subpilha é contado como um módulo quando o salto é feito em uma pilha pai. A ação de reset irá repor o estado de uma

pilha de módulo para o estado em que estava, como de início da avaliação da subpilha.

- Para os valores de controle válidos de sintaxe mais complicada tem a seguinte forma:
  - [valor1 = ação1 valor2 = ação2 ...]
  - Onde valorN corresponde ao código de retorno da função chamada no módulo para o qual a linha é definida. Ele é selecionado a partir de um destes: success, open_err, symbol_err, service_err, system_err, buf_err, perm_denied, auth_err, cred_insufficient, authinfo_unavail, user_unknown, maxtries, new_authtok_reqd, acct_expired, session_err, cred_unavail, cred_expired, cred_err, no_module_data, conv_err, authtok_err, authtok_recover_err, authtok_lock_busy, authtok_disable_aging, try_again, ignore, abort, authtok_expired, module_unknown, bad_item, conv_again, incomplete, e default.
  - O último deles, padrão, implica em todos os valoresN não mencionados de forma explícita. Note-se, a lista completa dos erros PAM está disponível em /usr/include/security/_pam_types.h. A açãoN pode ser: um inteiro sem sinal, n, significando uma ação de pular ao longo dos próximos n módulos na pilha; ou tomar uma das seguintes formas:
    - ignore - quando usado com uma pilha de módulos, o estado de retorno do módulo não irá contribuir para o código de retorno que o aplicativo obtém.
    - bad - essa ação indica que o código de retorno deve ser pensado como um indicativo de falha do módulo. Se esse módulo é o primeiro na pilha a falhar, o seu valor de estado será utilizado para a toda a pilha.
    - die - o equivalente a ruim com o efeito colateral de pilha que encerra o módulo PAM e imediatamente retorna para o aplicativo.
    - ok - isso diz PAM que o administrador pensa que esse código de retorno deve contribuir diretamente para o código de retorno da pilha completa de módulos. Em outras palavras, se o estado anterior da pilha levaria a um retorno de PAM_SUCCESS, o código de retorno do módulo irá substituir esse valor. Observe que, se o estado anterior da pilha contém algum valor que é indicativo de uma falha de módulos, esse valor 'ok' não irá ser utilizado para substituir esse valor.
    - done - o equivalente a ok com o efeito colateral de pilha que encerra o módulo PAM e imediatamente retornar para o aplicativo.
    - reset - limpar toda a memória do estado da pilha de módulo e começa de novo com o próximo módulo empilhado.
  - Cada uma das quatro palavras-chave: necessário; requisito; suficiente; e opcional, tem uma expressão equivalente em termos da sintaxe [...]. Eles são como se segue:
    - required - [success=ok new_authtok_reqd=ok ignore=ignore default=bad]
    - requisite - [success=ok new_authtok_reqd=ok ignore=ignore default=die]
    - sufficient - [success=done new_authtok_reqd=done default=ignore]
    - optional - [success=ok new_authtok_reqd=ok default=ignore]

- <caminho do módulo> é tanto o nome do arquivo completo do PAM para ser usado pelo aplicativo (ele começa com uma '/'), ou um caminho relativo a partir da localização de módulo padrão: /lib/security/ ou /lib64/security/, dependendo na arquitetura.
- <argumentos do módulo> são uma lista separada por espaço de tokens que podem ser usados para modificar o comportamento específico do PAM dado. Tais argumentos serão documentados para cada módulo individual. Note que, se se deseja incluir espaços em um argumento, deve-se cercar esse argumento com colchetes.
    - squid auth required pam_mysql.so user=passwd_query passwd=mada \
    - db=eminence [query=select user_name from internet_service \
    - where user_name=´%u´ and password=PASSWORD(´%p´) and \
    - service=´web_proxy´]
- Ao usar essa convenção, pode-se incluir caracteres '[' dentro da cadeia, e se deseja incluir um caractere ']' dentro da cadeia que vai sobreviver o argumento de análise, deve-se usar "\] '. Em outras palavras:
    - [..[..\]..] --> ..[..]..
- Qualquer linha em (um dos) arquivo(s) de configuração, que não está formatado corretamente, em geral, tendem (errando no lado do cuidado) fazer o processo de autenticação falhar. Um erro correspondente é gravado nos arquivos de log do sistema com uma chamada para o syslog(3).
- Mais flexível do que o arquivo de configuração único é configurar o libpam via o conteúdo do diretório /etc/pam.d/. Nesse caso, o diretório é preenchido com arquivos que cada uma deles tem um nome igual a um nome do serviço (em letras minúsculas): é o arquivo de configuração pessoal para o serviço nomeado.
- A sintaxe de cada arquivo em /etc/pam.d/ é semelhante ao do /etc/pam.conf e é composta de linhas da seguinte forma:
    - <tipo> <controle> <caminho do módulo> <argumentos do módulo>
- A única diferença é que o nome do serviço não está presente. O nome do serviço é, naturalmente, o nome do arquivo de configuração dado. Por exemplo, /etc/pam.d/login contém a configuração para o serviço login.

- Name Service Switch [4]
    - O Name Service Switch (NSS) é uma facilidade em sistemas operacionais Unix-like que fornece uma variedade de fontes para bancos de dados de configuração comuns e mecanismos de resolução de nomes. Essas fontes incluem arquivos do sistema operacional local (tais como /etc/passwd, /etc/group e /etc/hosts), o Domain Name System (DNS), o Network Information Service (NIS), e LDAP.
    - nsswitch.conf
        - Um administrador de sistema geralmente configura serviços de nome do sistema operacional usando o arquivo /etc/nsswitch.conf. Esse lista bancos de dados (tais como passwd, shadow e grupo) e uma ou mais fontes de obtenção dessa informação. Exemplos de fontes são files para arquivos locais, ldap para o Lightweight Directory Access Protocol, nis para o Serviço de Informação de Rede, nisplus para NIS+, e wins para o Windows Internet Name Service.
        - O arquivo nsswitch.conf tem entradas de linha para cada serviço que consiste em um nome de banco de dados no primeiro campo, terminado por dois pontos, e uma

lista de possíveis mecanismos de bancos de dados de origem no segundo campo. Um arquivo típico pode parecer:

- passwd: files ldap
- shadow: files
- group: files ldap
- hosts: dns nis files
- ethers: files nis
- netmasks: files nis
- networks: files nis
- protocols: files nis
- rpc: files nis
- services: files nis
- automount: files
- aliases: files

    - A ordem dos serviços listados determina em que ordem o NSS tentará usar esses serviços para resolver consultas no banco de dados especificado.

- Arquivos de configuração
    - /etc/pam.d/ - diretório de configuração do PAM
    - pam.conf(5) - arquivo de configuração do PAM
- Exemplo de configuração
    - /etc/pam.d/system-auth
        - auth required pam_env.so
        - auth sufficient pam_unix.so nullok try_first_pass
        - auth requisite pam_succeed_if.so uid >= 500 quiet
        - auth required pam_deny.so
        - account required pam_unix.so
        - account sufficient pam_localuser.so
        - account sufficient pam_succeed_if.so uid < 500 quiet
        - account required pam_permit.so
        - password requisite pam_cracklib.so try_first_pass retry=3 type=
        - password sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
        - password required pam_deny.so
        - session optional pam_keyinit.so revoke
        - session required pam_limits.so
        - session [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
        - session required pam_unix.so
- Alguns módulos
    - pam_unix(8) - módulo para autenticação de senha tradicional
    - pam_cracklib(8) - módulo PAM para verificar a senha contra dicionário de palavras
    - pam_limits(8) - módulo PAM para limitar recursos
    - pam_listfile(8) - nega ou permite serviços baseados em um arquivo arbitrário

Senhas passwd e shadow

- Arquivo passwd [5]

    - O arquivo /etc/passwd é um banco de dados baseado em texto de informações sobre os usuários que podem logar no sistema ou outras identidades de usuários do sistema operacional que possuem processos em execução.
    - Em muitos sistemas operacionais esse arquivo é apenas um dos muitos possíveis backends para o serviço de nome passwd mais geral.
    - O nome do arquivo se origina de uma de suas funções iniciais, uma vez que continha os dados usados para verificar as senhas de contas de usuário. No entanto, em sistemas Unix modernos as informações de senha de segurança sensível, em vez disso, é muitas vezes armazenadas em um arquivo diferente usando senhas sombra (shadow passwords), ou outras implementações de banco de dados.
    - O arquivo /etc/passwd normalmente tem permissões de sistema de arquivos que permitem que ele seja lido por todos os usuários do sistema (de leitura para todos), embora só pode ser modificado pelo superusuário ou usando alguns comandos privilegiados de propósito especial.
    - O arquivo /etc/passwd é um arquivo de texto com um registro por linha, cada um descrevendo uma conta de usuário. Cada registro consiste em sete campos separados por dois pontos. A ordenação dos registros dentro do arquivo é geralmente sem importância.
- Arquivo shadow [5]
    - /etc/shadow é utilizado para aumentar o nível de segurança de senhas, restringindo todos, mas o acesso de usuários altamente privilegiados aos dados hash de senha. Normalmente, os dados são mantidos em arquivos de propriedade de e acessível somente pelo super usuário.
    - Os administradores de sistemas podem reduzir a probabilidade de ataques de força bruta, fazendo a lista de senhas com hash ilegíveis por usuários sem privilégios. A maneira óbvia de fazer isso é fazer com que o próprio banco de dados passwd legível somente pelo usuário root. No entanto, isso pode restringir o acesso a outros dados no arquivo, como mapeamentos de nome de usuário para id de usuário, que iria quebrar muitas utilidades e disposições existentes. Uma solução é um arquivo de senhas "sombra" para manter os hashes de senha separado dos outros dados no arquivo passwd legível por todos. Para arquivos locais, esse é normalmente /etc/shadow em sistemas Linux e Unix, ou /etc/master.passwd nos sistemas BSD; cada um é lido somente pelo root. (Acesso aos dados pelo root é considerado aceitável uma vez em sistemas com o tradicional modelo de segurança "root todo-poderoso", o usuário root seria capaz de obter as informações por outros meios, em qualquer caso). Praticamente todos os últimos sistemas operacionais Unix-like usam senhas em shadow.
    - O arquivo de senhas sombra não resolve completamente o problema do atacante acessar o hash de senhas, como alguns esquemas de autenticação de rede operam transmitindo o hash de senha através da rede (por vezes em texto puro, por exemplo, Telnet), tornando-o vulnerável à interceptação. Cópias de dados do sistema, como backups do sistema escritos em fita ou mídia óptica, também pode se tornar um meio de obter ilicitamente hash de senhas. Além disso, as funções utilizadas por programas de verificação de senha legítimos precisam ser escritas de tal maneira que os programas mal-intencionados não podem fazer um grande número de verificações de autenticação em altas taxas de velocidade.
    - O formato do arquivo shadow é simples, e basicamente idêntico ao do arquivo de senhas, a saber, uma linha por usuário, campos ordenados em cada linha, e os campos separados por dois pontos. Muitos sistemas requerem a ordem de linhas de usuários no arquivo shadow ser idêntica à ordem dos usuários correspondentes no arquivo de senha.

Termos e utilitários

- /etc/pam.d/ - diretório de configuração do PAM
- pam.conf - (5) arquivo de configuração do PAM
- nsswitch.conf - (5) arquivo de configuração do Name Service Switch
- pam_unix, pam_cracklib, pam_limits, pam_listfile
    - pam_unix - (8) módulo para autenticação de senha tradicional
    - pam_cracklib - (8) módulo PAM para verificar a senha contra dicionário de palavras
    - pam_limits - (8) módulo PAM para limitar recursos
    - pam_listfile - (8) nega ou permite serviços baseados em um arquivo arbitrário

Referências

1. http://en.wikipedia.org/wiki/Pluggable_authentication_module
2. http://en.wikipedia.org/wiki/Linux_PAM
3. pam(8)
4. http://en.wikipedia.org/wiki/Name_Service_Switch
5. http://en.wikipedia.org/wiki/Passwd

Simulado

1. O arquivo /etc/pam.conf tem precedência sobre o diretório /etc/pam.d.
    a. V
    b. F

2. O arquivo de configuração /etc/pam.d/sudo é usado para configurar o processo de autenticação do serviço sudo.
    a. V
    b. F

3. Somente no arquivo de configuração /etc/pam.conf, o campo serviço é usado.
    a. V
    b. F

4. Se um módulo que foi declarado como requerido falha, o processamento PAM encerra e retorna imediatamente para a aplicação.
    a. V
    b. F

5. Módulos que são suficientes como controle, não impactam no caso de falha.
    a. V
    b. F

6. Módulos opcionais podem falhar, sem afetar o processo de autenticação.
    a. V
    b. F

7. Os tipos …, …, ... e … dizem respeito respectivamente a sessão, troca de senha, autorização e autenticação.

8. Cada módulo pode ter seu próprio conjunto de argumentos, que definem o seu comportamento durante o processo de autenticação de um serviço.
    a. V
    b. F

9. A diferença entre os controles requerido e requisito é a mesma entre a açãoN bad e die.
    a. V
    b. F

10. Os módulos PAM …, …, … e … são usados respectivamente para limitar acesso a recursos, negar ou permitir serviços baseados em um arquivo arbitrário, autenticar usando senha de forma tradicional e testar a senha contra um dicionário de palavras.

11. A permissão no formato numérico do arquivo /etc/shadow é ….

12. O arquivo de senhas shadow tem como principal objetivo, negar acesso de leitura a outros usuários que não sejam o superusuário.
    a. V
    b. F

13. O arquivo de senhas (/etc/passwd) pode conter senhas em texto claro.
    a. V
    b. F

14. O arquivo de senhas sombreadas possuem outras informações sobre as contas dos usuários como data de criação da conta.
    a. V
    b. F

15. O arquivo de senhas (/etc/passwd) pode substituir por completo o arquivo de senhas sombreadas.
    a. V
    b. F

210.3 Uso de cliente LDAP

Visão geral

Peso: 2
Descrição:

- Os candidatos devem ser capazes de executar consultas e atualizações em um servidor LDAP. Também está incluído importando e adicionando itens, bem como adicionando e mantendo usuários.

Áreas de conhecimentos chave:

- Utilitários LDAP parar gerenciamento de dados e consultas
- Alterar senhas de usuários
- Consultas ao diretório LDAP

Termos e utilitários:

- ldapsearch
- ldapadd
- ldappasswd
- ldapdelete

Áreas de conhecimentos chave

Utilitários LDAP parar gerenciamento de dados e consultas

- LDAP [1]
    - O Lightweight Directory Access Protocol (LDAP) é um, protocolo de aplicação aberto, de fabricante neutro e padrão de indústria, para acessar e manter serviços de informação de diretório, distribuídos através de uma rede IP (Internet Protocol). Os serviços de diretório desempenham um papel importante no desenvolvimento de aplicações intranet e Internet, permitindo a partilha de informações sobre os usuários, sistemas, redes, serviços e aplicações em toda a rede. Como exemplos, serviços de diretório podem fornecer qualquer conjunto organizado de registros, muitas vezes com uma estrutura hierárquica, tais como um diretório de e-mail corporativo. Da mesma forma, uma lista telefónica é uma lista de assinantes com um endereço e um número de telefone.
    - O LDAP é especificado em uma série de publicações Trilha Padrão do Internet Engineering Task Force (IETF) chamadas Request for Comments (RFC), usando a linguagem de descrição ASN.1. A especificação mais recente é a versão 3, publicada como RFC 4511. Por exemplo, aqui está uma pesquisa LDAP traduzida em Inglês simples: "Search in the company email directory for all people located in Nashville whose name contains 'Jesse' that have an email address. Please return their full name, email, title, and description” ... "Pesquisar no diretório de e-mail da empresa por todas as pessoas, localizadas em Nashville das quais o nome contém ‘Jesse’, que tenha um endereço de e-mail. Por favor, retorne o nome completo, e-mail, título e descrição ".
    - Um uso comum do LDAP é fornecer um "single sign on", onde uma senha de um usuário é compartilhada entre diversos serviços, tais como a aplicação de um código de login da empresa para páginas web (para que a equipe logue apenas uma vez nos computadores da empresa, e, em seguida, são automaticamente registrados na intranet da empresa).
    - O LDAP é baseado em um subconjunto mais simples dos padrões contidos no padrão X.500. Devido a essa relação, o LDAP é às vezes chamado X.500-lite.
    - Visão geral do protocolo

- Um cliente inicia uma sessão LDAP conectando a um servidor LDAP, chamado de Agente de Sistema de Diretório (DSA), por padrão na porta 389 TCP e UDP, ou na porta 636 para LDAPS. O catálogo global está disponível por padrão nas portas 3268 e 3269 para LDAPS. O cliente então envia um pedido de operação para o servidor, e o servidor envia respostas em troca. Com algumas exceções, o cliente não precisa esperar por uma resposta antes de enviar a próxima solicitação, e o servidor pode enviar as respostas em qualquer ordem. Toda a informação é transmitida usando Basic Encoding Rules (BER).
- O cliente pode solicitar as seguintes operações:
  - StartTLS - usar a extensão Transport Layer Security (TLS) do LDAPv3 para uma conexão segura
  - Vincular (Bind) - autenticar e especificar a versão do protocolo LDAP
  - Pesquisar (Search) - procurar e/ou recuperar entradas de diretório
  - Comparar (Compare) - testar se uma entrada chamada contém um determinado valor de atributo
  - Adicionar (Add) - adicionar uma nova entrada
  - Deletar (Delete) - remover uma entrada
  - Modificar (Modify) - modificar uma entrada
  - Modificar nome distinto (DN) (Modrdn) - mover ou renomear uma entrada
  - Abandonar (Abandon) - abortar um pedido anterior
  - Operação Extendida - operação genérica usada para definir outras operações
  - Desvincular (Unbind) - fechar a conexão (não é o inverso do Bind)
- Além disso, o servidor pode enviar notificações "não solicitadas" que não são respostas a qualquer pedido, por exemplo, antes que a conexão é excedida.
- Um método alternativo comum de proteger a comunicação LDAP é usando um túnel SSL. Esse é denotado nas URLs LDAP usando o esquema de URL "ldaps". A porta padrão para o LDAP sobre SSL é 636. O uso de LDAP sobre SSL era comum no LDAP versão 2 (LDAPv2) mas nunca foi padronizado em qualquer especificação formal. Esse uso foi depreciado juntamente com LDAPv2, que foi oficialmente aposentado em 2003.

- Estrutura de diretório
  - O protocolo fornece uma interface com os diretórios que seguem a edição 1993 do modelo X.500:
    - Uma entrada é constituída por um conjunto de atributos.
    - Um atributo possui um nome (um atributo tipo ou atributo descrição) e um ou mais valores. Os atributos são definidos em um esquema.
    - Cada entrada tem um identificador único: o seu nome distinto (DN). Esse consiste em seu nome distinto relativo (RDN), construído a partir de algum(s) atributo(s) na(s) entrada(s), seguido pelo DN da entrada pai. Pense no DN como o caminho de arquivo completo e o RDN como seu nome de arquivo relativo em sua pasta pai (por exemplo, se /foo/bar/myfile.txt fosse o DN, então, myfile.txt seria o RDN).
  - Um DN pode mudar ao longo do tempo de vida da entrada, por exemplo, quando as entradas são movidas dentro de uma árvore. Para identificar com segurança e de forma inequívoca as entradas, um UUID pode ser fornecido no conjunto de atributos operacionais da entrada.

- Uma entrada pode ficar assim quando representada no LDAP Data Interchange Format (LDIF) (LDAP em si é um protocolo binário):
  - dn: cn=John Doe,dc=example,dc=com
  - cn: John Doe
  - givenName: John
  - sn: Doe
  - telephoneNumber: +1 888 555 6789
  - telephoneNumber: +1 888 555 1232
  - mail: john@example.com
  - manager: cn=Barbara Doe,dc=example,dc=com
  - objectClass: inetOrgPerson
  - objectClass: organizationalPerson
  - objectClass: person
  - objectClass: top
- "dn" é o nome distinto da entrada; não é nem um atributo nem uma parte da entrada. "cn=John Doe" é RDN da entrada (Relative Distinguished Name), e "dc=example,dc=com" é o DN da entrada pai, onde "dc" denota "componente de domínio" (Domain Component). As outras linhas mostram os atributos na entrada. Os nomes de atributos são tipicamente strings mnemônicas, como "cn" para nome comum, "dc" para o componente de domínio, "mail" para endereço de e-mail, e "sn" para sobrenome.
- Um servidor mantém uma subárvore a partir de uma entrada específica, por exemplo, "dc=example,dc=com" e seus filhos. Os servidores também podem conter referências a outros servidores, portanto, uma tentativa de acesso "ou=departament,dc=example,dc=com" poderia retornar uma referência (referral) ou referência de continuação (continuation reference) a um servidor que contém a parte da árvore de diretórios. O cliente pode, então, entrar em contato com o outro servidor. Alguns servidores também suportam encadeamento, o que significa que o servidor contata o outro servidor e retorna os resultados para o cliente.
- O LDAP raramente define qualquer ordenação: o servidor pode retornar os valores de um atributo, os atributos em uma entrada e as entradas encontradas por uma operação de busca em qualquer ordem. Isso segue a partir das definições formais - uma entrada é definida como um conjunto de atributos, e um atributo é um conjunto de valores, e conjuntos não necessitam de serem ordenados.

- Operações
  - Adicionar
    - A operação ADD insere uma nova entrada no banco de dados do servidor de diretório. Se o nome distinto na requisição add já existe no diretório, em seguida, o servidor não irá adicionar uma entrada duplicada, mas irá definir o código de resultado no resultado add para decimal 68, "entryAlreadyExists".
      - Servidores compatíveis com LDAP nunca excluem a referência do nome distinto transmitido na requisição de adição ao tentar localizar a entrada, ou seja, nomes distintos nunca são “desapelidados”.
      - Servidores compatíveis com LDAP irão garantir que o nome diferenciado e todos os atributos estão em conformidade com os padrões de nomenclatura.
      - A entrada a ser adicionada não deve existir, e o superior imediato deve existir.

        - O comando usado para pesquisar as entradas é:
            - ldapsearch -h localhost -x -LLL -W -D "cn=admin,dc=example,dc=com" -b "dc=example,dc=com"
    - Exemplo LDIF de adicionar
        - dn: uid=user,ou=people,dc=example,dc=com
        - changetype: add
        - objectClass: top
        - objectClass: person
        - uid: user
        - sn: last-name
        - cn: common-name
        - userPassword: password
    - No exemplo acima, uid=user,ou=people,dc=example,dc=com não deve existir, e ou=people,dc=example,dc=com tem de existir.
- Bind (autenticar)
    - Quando uma sessão LDAP é criada, ou seja, quando um cliente LDAP se conecta ao servidor, o estado de autenticação da sessão é definido como anônimo. A operação BIND estabelece o estado de autenticação para uma sessão.
    - O BIND simples e SASL PLAIN pode enviar o DN do usuário e senha em texto simples, então as conexões que utilizam simples ou SASL PLAIN deve ser criptografadas usando Transport Layer Security (TLS). O servidor normalmente verifica a senha contra o atributo userPassword na entrada chamada. Anonymous BIND (com DN e senha vazios) redefine a conexão para o estado anônimo.
    - O BIND SASL (Simple Authentication and Security Layer) fornece serviços de autenticação através de uma ampla gama de mecanismos, por exemplo, Kerberos ou o certificado de cliente enviado com TLS.
    - O BIND também define a versão do protocolo LDAP enviando um número de versão na forma de um número inteiro. Se o cliente solicitar uma versão que o servidor não suporta, o servidor deve definir o código de resultado na resposta BIND, para um erro de protocolo. Normalmente, os clientes devem usar LDAPv3, que é o padrão no protocolo, mas nem sempre em bibliotecas LDAP.
    - O BIND tinha que ser a primeira operação em uma sessão no LDAPv2, mas não é necessário como do LDAPv3. No LDAPv3, cada solicitação BIND bem sucedida altera o estado de autenticação da sessão e cada solicitação BIND vencida redefine o estado de autenticação da sessão.
- Deletar
    - Para deletar uma entrada, um cliente LDAP transmite uma requisição de deleção devidamente formada para o servidor.
        - A requisição de deleção deve conter o nome distinto da entrada a ser excluída.
        - Requisição de controles podem também ser ligadas a requisições de deleção.
        - Servidores não excluem referências de apelidos ao processar uma solicitação de deleção.

        - Somente as entradas de folha (entradas sem subordinados) podem ser deletadas por uma requisição de deleção. Alguns servidores suportam um atributo operacional “hasSubordinates” cujo valor indica se uma entrada tem quaisquer entradas subordinadas, e alguns servidores suportam um atributo operacional “numSubordinates” indicando o número de entradas subordinadas à entrada contendo o atributo numSubordinates.
        - Alguns servidores suportam o controle de requisição de deleção de subárvores permitindo eliminação do DN e todos os objetos subordinados ao DN, sujeitos a controles de acesso. Requisições de deleção são sujeitas a controles de acesso, ou seja, se uma conexão com um determinado estado de autenticação terá permissão para excluir uma determinada entrada, é regido por mecanismos de controle de acesso específicos do servidor.
- Pesquisar e comparar
    - A operação de pesquisa é usada para tanto procurar e ler as entradas. Seus parâmetros são:
        - baseObject - O nome da entrada de objeto de base (ou possivelmente a raiz) em relação ao qual a pesquisa está a ser executada.
        - scope - quais elementos abaixo do baseObject para busca. Isso pode ser BaseObject (procure apenas a entrada chamada, normalmente usado para ler uma entrada), singleLevel (entradas imediatamente abaixo da base DN), ou wholeSubtree (toda a subárvore começando na base DN).
        - filter - critérios para usar na seleção de elementos dentro do escopo. Por exemplo, o filtro (&(objectClass=person)(|(givenName=John)(mail=john*))) irá selecionar "pessoas" (elementos de objectClass person) onde as regras de correspondência para givenName e mail determinam se os valores para esses atributos correspondem a afirmação de filtro. Note-se que um equívoco comum é que os dados LDAP são case-insensitive, nas regras de correspondência de fatos e regras de ordenação que determinam correspondência, comparações e relações de valor relativo. Se os exemplos de filtros foram necessários para corresponder o caso do valor de atributo, um filtro de correspondência extensível tem de ser usado, por exemplo, (&(objectClass=person)(|(givenName:caseExactMatch:=John)(mail:caseExactSubstringsMatch:=John*))).
        - derefAliases - se e como seguir as entradas de alias (entradas que se referem a outras entradas).
        - attributes - quais atributos retornar nas entradas de resultados.
        - sizeLimit, timeLimit - o número máximo de entradas para retornar, e tempo máximo para permitir a pesquisa executar. Esses valores, no entanto, não podem substituir quaisquer restrições que o servidor coloca no limite de tamanho e limite de tempo.
        - typesOnly - retorna atributos de tipos apenas, não atributos de valores.

- O servidor retorna as entradas correspondentes e, potencialmente, as referências de continuação. Essas podem ser devolvidas em qualquer ordem. O resultado final irá incluir o código de resultado.
- A operação de comparação tem uma DN, um nome de atributo e um valor de atributo, e verifica se a entrada chamada contém esse atributo com esse valor.

■ Modificar
- A operação MODIFY é usada por clientes LDAP para requisitar que o servidor LDAP faça alterações para entradas existentes. As tentativas de modificar as entradas que não existem falhará. Pedidos MODIFY estão sujeitos a controles de acesso conforme implementado pelo servidor.
- A operação de modificação requer que o nome distinto (DN) da entrada seja especificado, e uma sequência de mudanças. Cada mudança na seqüência deve ser um dos seguintes:
  - add (adicionar um novo valor, que não deve existir no atributo)
  - delete (deletar um valor existente)
  - replace (substituir um valor existente por um novo valor)
- Exemplo LDIF de adicionar um valor a um atributo:
  - dn: dc=example,dc=com
  - changetype: modify
  - add: cn
  - cn: the-new-cn-value-to-be-added
  - -
- Para substituir o valor de um atributo existente, use a palavra-chave replace. Se o atributo é multi-valorizado, o cliente deve especificar o valor do atributo a ser excluído.
- Para excluir um atributo de uma entrada, use a palavra-chave delete e o designador changetype modify. Se o atributo é multi-valorizado, o cliente deve especificar o valor do atributo a ser excluído.
- Há também uma extensão de incremento de modificação que permite que um valor de atributo incrementável seja incrementado em uma quantidade especificada. A extensão de modificação de incremento usa o identificador de objeto 1.3.6.1.1.14. O exemplo a seguir usando LDIF incrementa employeeNumber por 5:
  - dn: uid=user.0,ou=people,dc=example,dc=com
  - changetype: modify
  - increment: employeeNumber
  - employeeNumber: 5
  - -
- Quando servidores LDAP estão em uma topologia replicada, os clientes LDAP devem considerar usando o controle pós-leitura para verificar atualizações em vez de uma pesquisa depois de uma atualização. O controle pós-ler é projetado para que os aplicativos não precisam emitir uma solicitação de pesquisa após uma atualização - é má forma para recuperar uma entrada para o único propósito de verificar se uma atualização funcionou por causa do modelo de consistência eventual de replicação. Um cliente de LDAP não deve presumir que ele se conecta ao mesmo servidor de diretório

para cada solicitação porque arquitetos podem ter colocado balanceadores de carga ou proxies LDAP ou ambos entre clientes e servidores LDAP.

- Modificar DN
  - Modificar DN (mover/renomear entrada) leva o novo RDN (Relative Distinguished Name), opcionalmente o DN do novo pai, e uma flag que diga se deseja excluir o(s) valor(es) na entrada que coincide com o velho RDN. O servidor pode suportar a renomeação de subárvores inteiras do diretório.
  - Uma operação de atualização é atômica: outras operações verão a nova entrada ou a antiga. Por outro lado, o LDAP não define operações de múltiplas operações: se você ler uma entrada e, em seguida, modificá-la, um outro cliente pode ter atualizado a entrada nesse meio tempo. Os servidores podem implementar extensões que suportam isso, porém.
- Operações estendidas
  - A operação extendida é uma operação LDAP genérica que pode definir novas operações que não faziam parte da especificação do protocolo original. O StartTLS é uma das extensões mais significativas. Outros exemplos incluem o Anular (Cancel) e Modificar Senha (Password modify).
- StartTLS
  - A operação StartTLS estabelece Transport Layer Security (o descendente de SSL) na conexão. Ele pode fornecer confidencialidade de dados (para proteger os dados de serem observados por terceiros) e/ou proteção da integridade dos dados (que protege os dados contra adulteração). Durante a negociação TLS o servidor envia seu certificado X.509 para provar sua identidade. O cliente também pode enviar um certificado para provar sua identidade. Após fazer isso, o cliente pode então usar SASL/EXTERNAL. Ao utilizar o SASL/EXTERNAL, o cliente solicita que o servidor deriva sua identidade a partir de credenciais fornecidas em um nível inferior (como TLS). Embora tecnicamente o servidor pode utilizar quaisquer informações de identidade estabelecidas em qualquer nível mais baixo, geralmente o servidor irá utilizar as informações de identidade estabelecidas por TLS.
  - Os servidores também muitas vezes suportam o protocolo não-padrão "LDAPS" ("LDAP seguro", vulgarmente conhecido como "LDAP sobre SSL") em uma porta separada, por padrão 636. O LDAPS difere de LDAP de duas formas: 1) em cima de conexão, o cliente e servidor estabelecem o TLS antes que quaisquer mensagens LDAP são transferidas (sem uma operação StartTLS) e 2) a conexão LDAPS deve ser fechada após o fechamento TLS.
  - Note-se que algumas bibliotecas do cliente "LDAPS" apenas criptografam a comunicação; elas não verificar o nome do host contra o nome no certificado fornecido.
- Abandonar
  - A operação Abandon requisita que o servidor aborte uma operação chamada por um ID de mensagem. O servidor não precisa honrar o pedido. Nem o Abandon nem uma operação abandonada com êxito enviam uma resposta. Uma operação extendida Cancelar semelhante envia as respostas, mas nem todas as implementações suportam isso.
- Unbind

- A operação Unbind abandona quaisquer operações pendentes e fecha a conexão. Não tem qualquer resposta. O nome é de origem histórica, e não é o oposto da operação de ligação.
- Os clientes podem cancelar uma sessão, basta fechar a conexão, mas eles devem usar o Unbind. O Unbind permite que o servidor feche graciosamente a conexão e libere recursos que, de outra forma mantém por algum tempo até descobrir que o cliente tinha abandonado a conexão. Ele também instrui o servidor para cancelar as operações que podem ser canceladas, e para não enviar as respostas para as operações que não podem ser cancelados.

  - URLs LDAP
    - Um formato de URL LDAP existe, que clientes suportam em diferentes graus, e servidores retornam em referências e menções complementares (ver RFC 4516):
      - ldap://host:port/DN?attributes?scope?filter?extensions
    - A maioria dos componentes descritos abaixo são opcionais.
      - host é o endereço IP ou FQDN do servidor LDAP para busca.
      - port é a porta de rede (porta padrão 389) do servidor LDAP.
      - DN é o nome distinto a ser usado como base de pesquisa.
      - attributes é uma lista de atributos separados por vírgulas para recuperar.
      - scope especifica o escopo da pesquisa e pode ser "base" (o padrão), "one" ou "sub".
      - filter é um filtro de pesquisa. Por exemplo (objectClass=*), tal como definido na RFC 4515.
      - extensions são extensões para o formato URL LDAP.
    - Por exemplo, "ldap://ldap.example.com/cn=John%20Doe,dc=example,dc=com" refere-se a todos os atributos do usuário na entrada de John Doe em ldap.example.com, enquanto "ldap:///dc=example,dc=com??sub?(givenName=John)" pesquisa a entrada no servidor padrão (observe a barra tripla, omitindo o host, e o ponto de interrogação duplo, omitindo os atributos). Como em outras URLs, os caracteres especiais devem ser codificado por cento.
    - Há semelhantes ldaps não-padrão: esquema de URL para LDAP sobre SSL. Isso não deve ser confundido com LDAP com TLS, o que é conseguido usando a operação StartTLS usando o esquema ldap padrão.

- OpenLDAP [2]
  - OpenLDAP é uma implementação de código aberto do Lightweight Directory Access Protocol (LDAP) desenvolvido pelo Projeto OpenLDAP. Ele é liberado sob a sua própria licença BSD chamada o OpenLDAP Public License.
  - O LDAP é um protocolo independente de plataforma. Várias distribuições Linux incluem o Software OpenLDAP para suporte LDAP. O software também roda em variantes BSD, bem como AIX, Android, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT e derivados, por exemplo, 2000, XP, Vista, Windows 7, etc.), e z/OS.
- Utilitários
  - ldapsearch(1) - ferramenta de pesquisa LDAP
    - Exemplos:
      - ldapsearch <opções> <filtro> [atributos] - pesquisa usando o filtro especificado
      - -b <base> - especifica a base para consulta
      - -s <escopo> - especifica o escopo da consulta

            - ldapsearch -L - exibe o resultado em formato LDIF
            - ldapsearch -LL - exibe o resultado em formato LDIF, desabilitando comentários
            - ldapsearch -LLL - exibe o resultado em formato LDIF, desabilitando comentários e desabilitando a impressão da versão do LDIF
    - ldappasswd(1) - altera a senha de uma entrada LDAP
        - Exemplos:
            - ldappasswd <opções> <dn> - altera a senha do usuário especificado pelo dn
            - ldappasswd -A - exibe um prompt para a senha antiga
            - ldappasswd -a - especifica a senha antiga como argumento
            - ldappasswd -S - exibe um prompt para a nova senha
            - ldappasswd -s - especifica a nova senha como argumento
    - ldapadd(ldapmodify(1)) - ferramentas para modificar uma entrada LDAP e adicionar uma entrada LDAP
        - Exemplos:
            - ldapadd <opções> - adiciona entradas no LDAP recebidas da entrada padrão, no formato LDIF
            - ldapadd -f <arquivo> - especifica um arquivo LDIF para a operação
    - ldapdelete(1) - ferramenta para deletar uma entrada LDAP
        - Exemplos:
            - ldapdelete <opções> <dn> - remove o dn especificado
            - ldapdelete -r - remove de forma recursiva
    - Opções comuns aos utilitários:
        - -n - mostra o que seria feito
        - -v - modo verbose
        - -c - modo de operação continua (erros são reportados e ignorados)
        - -x - usa autenticação simples ao invés de SASL
        - -D <dn> - especifica um dn para autenticação
        - -W - exibe um prompt para a senha
        - -w <senha> - especifica a senha como argumento
        - -H <URI> - especifica a URI de acesso ao servidor
        - -h <endereço> - especifica o endereço do servidor
        - -p <porta> - especifica a porta do servidor
        - -P <versão> - especifica a versão do protocolo
        - -Z - solicita tentativa de conexão por TLS
        - -ZZ - exige conexão por TLS

Alterar senhas de usuários

- Através do ldappasswd

Consultas ao diretório LDAP

- Através do ldapsearch

Termos e utilitários

- ldapsearch - (1) ferramenta de pesquisa LDAP
- ldappasswd - (1) altera a senha de uma entrada LDAP

- ldapadd - ldapmodify(1) ferramentas para modificar uma entrada LDAP e adicionar uma entrada LDAP
- ldapdelete - (1) ferramenta para deletar uma entrada LDAP

Referências

1. http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
2. http://en.wikipedia.org/wiki/OpenLDAP

Exercícios práticos

1. Preparação para exercícios
    a. Instalar o pacote openldap-clients (Ex.: yum install <pacote>)

    b. Instalar o pacote openldap-servers (necessário apenas para os exercícios práticos) (Ex.: yum install <pacote>)

    c. Iniciar o serviço slapd (Ex.: service <serviço> start)

    d. Criar o arquivo vazio secret.ldif com o seguinte conteúdo: (Ex.: vi <arquivo>)
        i. dn: olcDatabase={2}bdb,cn=config
        ii. add: olcRootPW
        iii. olcRootPW: secret

    e. Alterar a entrada “olcDatabase={2}bdb,cn=config”, através do comando ldapmodify, usando como URI de acesso “ldapi:///” (ldapmodify -H ldapi:/// -f secret.ldif)

2. Utilitários LDAP parar gerenciamento de dados e consultas
    a. Criar o arquivo my-domain.com.ldif, declarando (Ex.: vi <arquivo>)
        i. Entrada
            1. Nome distinto (dn): dc=my-domain,dc=com
            2. Componente de domínio (dc): my-domain
            3. Classe de objeto (objectClass): dcObject
            4. Classe de objeto: organization
            5. Organização (organization): My Company
        ii. Entrada
            1. Nome distinto: ou=people,dc=my-domain,dc=com
            2. Unidade organizacional (ou): people
            3. Classe de objeto: organizationalUnit
            4. Classe de objeto: top
        iii. Entrada
            1. Nome distinto: ou=group,dc=my-domain,dc=com
            2. Unidade organizacional: group
            3. Classe de objeto: organizationalUnit
            4. Classe de objeto: top

b. Adicionar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapadd <opções>)

c. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

d. Criar o arquivo users.ldif, declarando (Ex.: vi <arquivo>)
   i. Entrada
      1. Nome distinto: cn=user1,ou=people,dc=my-domain,dc=com
      2. Nome comum (cn): user1
      3. Classe de objeto: person
      4. Classe de objeto: top
      5. Sobrenome (sn): test
   ii. Entrada
      1. Nome distinto: cn=user2,ou=people,dc=my-domain,dc=com
      2. Nome comum: user2
      3. Classe de objeto: person
      4. Classe de objeto: top
      5. Sobrenome: test
   iii. Entrada
      1. Nome distinto: cn=user3,ou=people,dc=my-domain,dc=com
      2. Nome comum: user3
      3. Classe de objeto: person
      4. Classe de objeto: top
      5. Sobrenome: test
   iv. Entrada
      1. Nome distinto: cn=u4,ou=people,dc=my-domain,dc=com
      2. Nome comum: u4
      3. Classe de objeto: person
      4. Classe de objeto: top
      5. Sobrenome: Test

e. Adicionar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapadd <opções>)

f. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

g. Criar o arquivo groups.ldif, declarando (Ex.: vi <arquivo>)
   i. Entrada
      1. Nome distinto: cn=group1,ou=group,dc=my-domain,dc=com
      2. Nome comum: group1
      3. Classe de objeto: groupOfNames
      4. Membro (member): cn=user1,ou=people,dc=my-domain,dc=com
   ii. Entrada
      1. Nome distinto: cn=group2,ou=group,dc=my-domain,dc=com

2. Nome comum: group2
3. Classe de objeto: groupOfNames
4. Membro (member): cn=user2,ou=people,dc=my-domain,dc=com

iii. Entrada

1. Nome distinto: cn=group3,ou=group,dc=my-domain,dc=com
2. Nome comum: group3
3. Classe de objeto: groupOfNames
4. Membro (member): cn=user3,ou=people,dc=my-domain,dc=com

h. Adicionar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapadd <opções>)

i. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

j. Criar o arquivo users-mod.ldif, declarando (Ex.: vi <arquivo>)

i. Entrada

1. Nome distinto: cn=user1,ou=people,dc=my-domain,dc=com
2. Adicionar (add): description
3. Descrição (description): Test user 1
4. Substituir (replace): sn
5. Sobrenome: Test

ii. Entrada

1. Nome distinto: cn=user2,ou=people,dc=my-domain,dc=com
2. Adicionar (add): description
3. Descrição (description): Test user 2
4. Substituir (replace): sn
5. Sobrenome: Test

iii. Entrada

1. Nome distinto: cn=user3,ou=people,dc=my-domain,dc=com
2. Adicionar (add): description
3. Descrição (description): Test user 3
4. Substituir (replace): sn
5. Sobrenome: Test

k. Modificar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapmodify <opções>)

l. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

m. Apagar a entrada “cn=user3,ou=people,dc=my-domain,dc=com” do LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret” (Ex.: ldapdelete <opções>

n. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

o. Criar o arquivo group3.ldif, declarando (Ex.: vi <arquivo>)
    i. Entrada
        1. Nome distinto: cn=group3,ou=group,dc=my-domain,dc=com
        2. Adicionar: member
        3. Membro: cn=user1,ou=people,dc=my-domain,dc=com
        4. Adicionar: member
        5. Membro: cn=user2,ou=people,dc=my-domain,dc=com
        6. Remover (delete): member
        7. Membro: cn=user3,ou=people,dc=my-domain,dc=com

p. Modificar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapmodify <opções>)

q. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

r. Renomear a entrada do LDAP “cn=u4,ou=people,dc=my-domain,dc=com” para “cn=user4”, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret” (Ex.: ldapmodrdn <opções>)

s. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

t. Criar o arquivo user4.ldif, declarando (Ex.: vi <arquivo>)
    i. Entrada
        1. Nome distinto: cn=user4,ou=people,dc=my-domain,dc=com
        2. Remover: cn
        3. Nome comum: u4

u. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

3. Alterar senhas de usuários
    a. Tentar pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=user1,ou=people,dc=my-domain,dc=com” e a senha “secretUser1” e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

    b. Definir a senha do usuário user1 do LDAP para “secretUser1”, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret” (Ex.: ldappasswd <oções>)

    c. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

d. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=user1,ou=people,dc=my-domain,dc=com” e a senha “secretUser1” e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

e. Definir a senha do usuário user2 do LDAP para “secretUser2”, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=Manager,dc=my-domain,dc=com” e a senha “secret” (Ex.: ldappasswd <oções>)

f. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

4. Consultas ao diretório LDAP
   a. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima, a base de pesquisa dc=my-domain,dc=com e o escopo “base” (Ex.: ldapsearch <opções>)

   b. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima, a base de pesquisa dc=my-domain,dc=com e o escopo “one” (Ex.: ldapsearch <opções>)

   c. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima, a base de pesquisa dc=my-domain,dc=com e o escopo “sub” (Ex.: ldapsearch <opções>)

   d. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com, por pessoas (classe de objeto person) (Ex.: ldapsearch <opções>)

   e. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com, por pessoas (classe de objeto person) que tenham senhas definidas (Ex.: ldapsearch <opções>)

   f. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com, por pessoas (classe de objeto person) que tenham senhas definidas, retornando apenas o nome comum da pessoa (Ex.: ldapsearch <opções>)

   g. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com, por grupos (classe de objeto groupOfNames) que tenham como membro o usuário cn=user2,ou=people,dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

Simulado

1. LDAP é um protocolo leve de acesso a diretórios, baseado em um subconjunto do padrão X.500.
   a. V

b. F

2. Apesar de muitos softwares implementarem o protocolo LDAPS, o mesmo não é padronizado formalmente.
   a. V
   b. F

3. … é o formato de troca de informações para o protocolo LDAP.

4. O método StartTLS utiliza o protocolo LDAP na porta ….

5. Após estabelecer uma conexão LDAP, o cliente pode prover um estado autenticação usando o método ….

6. São comandos que possibilitam a troca de senha de um usuário no LDAP:
   a. ldapadd
   b. ldapbind
   c. ldappasswd
   d. ldapmodify

7. O comando ... , usado com o argumento … possibilita a exclusão de objetos no LDAP, de forma recursiva.

8. O comando … é usado para pesquisar informações em uma base LDAP.

9. Os parâmetros comuns …, …, … e … aos utilitários do OpenLDAP, são usados respectivamente para definir uma autenticação simples, exibir um prompt de senha para a autenticação, especificar um usuário para autenticação e especificar modo de operação contínua.

10. O comando ldapadd é um link físico para o comando ldapmodify, e o que corresponde ao comando ldapmodify -a.
    a. V
    b. F

210.4 Configurar um servidor OpenLDAP

Visão geral

Peso: 4
Descrição: Os candidatos devem ser capazes de configurar um servidor OpenLDAP básico incluindo conhecimento do formato LDIF e controle de acesso essencial. Um entendimento do papel do SSSD na autenticação e gerenciamento de identidade é incluído.
Áreas de conhecimentos chave:

- OpenLDAP
- Controle de acesso
- Nomes distintos
- Operações de mudanças de tipo
- Esquemas e páginas brancas
- Diretórios
- IDs de objetos, atributos e classes
- Consciência do Daemon de Serviços de Segurança de Sistemas (SSSD)

Termos e utilitários:

- slapd
- slapd.conf
- LDIF
- slapdadd
- slapcat
- slapindex
- /var/li/ldap/
- loglevel

Áreas de conhecimentos chave

OpenLDAP

- OpenLDAP [1]
    - OpenLDAP é uma implementação de código aberto do Lightweight Directory Access Protocol (LDAP) desenvolvido pelo Projeto OpenLDAP. Ele é liberado sob a sua própria licença BSD chamada o OpenLDAP Public License.
    - O LDAP é um protocolo independente de plataforma. Várias distribuições Linux incluem o Software OpenLDAP para suporte LDAP. O software também roda em variantes BSD, bem como AIX, Android, HP-UX, Mac OS X, Solaris, Microsoft Windows (NT e derivados, por exemplo, 2000, XP, Vista, Windows 7, etc.), e z/OS.
    - História
        - O projeto OpenLDAP foi iniciado em 1998 por Kurt Zeilenga. O projeto começou por clonagem da fonte de referência LDAP da Universidade de Michigan, onde um projeto de longa duração tinha suportado o desenvolvimento e evolução do protocolo LDAP até o lançamento final daquele projeto em 1996.
        - Em de Abril de 2006, o projeto OpenLDAP tem três membros do núcleo: Howard Chu (arquiteto-chefe), Pierangelo Masarati, e Kurt Zeilenga. Existem inúmeros outros contribuidores importantes e ativos, incluindo Luke Howard, Hallvard Furuseth, Quanah Gibson-Mount, e Gavin Henry.
    - Componentes
        - OpenLDAP tem três componentes principais:

- slapd - stand-alone LDAP daemon e módulos e ferramentas associadas
- bibliotecas de implementação do protocolo LDAP e Regras Codificação Básicas (BER) ASN.1
- software de cliente: ldapsearch, ldapadd, ldapdelete, e outros

■ Além disso, o Projeto OpenLDAP é o lar de uma série de subprojetos:

- JLDAP - bibliotecas de classes Java para LDAP
- JDBC-LDAP - Java JDBC - LDAP Bridge Driver
- ldapc++ - bibliotecas de classe LDAP para C++
- Fortress - gerenciamento de acesso de identidade com base em papéis Java SDK
- LMDB - biblioteca de banco de dados mapeados na memória

○ Backends

■ Conceito geral

- Historicamente, a arquitetura do servidor OpenLDAP (slapd, o Standalone LDAP Daemon) foi dividida entre um frontend que lida com acesso à rede e processamento do protocolo, e um backend que trata estritamente com o armazenamento de dados. Esse projeto da separação foi uma característica do código original escrito em 1996, da Universidade de Michigan, e desenvolvido em todos os subseqüentes lançamentos do OpenLDAP. O código original incluía um backend de banco de dados principal e dois backends experimental/demo. A arquitetura é modular e muitos backends diferentes estão agora disponíveis para interface com outras tecnologias, não apenas bancos de dados tradicionais.
- Nota: nas versões (1.x) mais antigas, os termos "backend" e "banco de dados" foram freqüentemente usados como sinônimos. Para ser preciso, um "backend" é uma classe de interface de armazenamento, e um "banco de dados" é uma instância de um backend. O servidor slapd pode usar arbitrariamente muitos backends de uma vez, e pode ter arbitrariamente muitas instâncias de cada backend (ou seja, arbitrariamente muitos bancos de dados) ativos ao mesmo tempo.

■ Backends disponíveis

- Atualmente 16 diferentes backends são fornecidos na distribuição OpenLDAP e vários terceiros são conhecidos por manter outros backends de forma independente. Os backends padrão são livremente organizados em três categorias diferentes:
- Backends de armazenamento de dados - esses realmente armazenam dados
  - back-bdb: o primeiro backend transacional para o OpenLDAP, construído em Berkeley DB
  - back-hdb: uma variante de back-bdb que é totalmente hierárquica e suporta renomeações de subárvore
  - back-ldif: construído em arquivos texto simples LDIF
  - back-mdb: um backend de banco de dados transacional construído sobre mapeamento de memória relâmpago do OpenLDAP (MDB)
  - back-ndb: um backend transacional construído sobre o motor de cluster NDB do MySQL
- Backends de proxy - esses atuam como gateway para outros sistemas de armazenamento de dados
  - back-ldap: proxy simples a outros servidores LDAP

        - back-meta: proxy com características meta-diretório
        - back-passwd: utiliza os dados do passwd e group de um sistema Unix
        - back-relay: redireciona internamente para outros servidores slapd
        - back-sql: fala com bancos de dados SQL arbitrários
      - Backends dinâmicos - esses geram dados em tempo real
        - back-config: configuração slapd via LDAP
        - back-dnssrv: localiza servidores LDAP via DNS
        - back-monitor: estatísticas do slapd via LDAP
        - back-null: um backend sink/no-op, análogo ao Unix /dev/null
        - back-perl: invoca módulos perl arbitrários em resposta a requisições LDAP
        - back-shell: invoca scripts shell para os requisições LDAP
        - back-sock: encaminha solicitações LDAP sobre IPC, para daemons arbitrários
      - Alguns backends disponíveis em versões mais antigas do OpenLDAP foram retirados de uso, principalmente o back-ldbm que foi herdado do código UMich original, e o back-tcl que era semelhante ao back-perl e back-shell.
      - Suporte para outros backends em breve serão retirado também. O back-ndb é obsoleto agora desde que a parceria com o MySQL que levou ao seu desenvolvimento foi encerrado pela Oracle após a Oracle adquirir o MySQL. O back-bdb e back-hdb serão depreciados em favor do back-mdb logo desde que ele é superior em todos os aspectos de desempenho, confiabilidade e capacidade de gerenciamento.
      - Na prática, backends como -perl, -shell, e -sock permitem interface com qualquer linguagem de programação arbitrária, assim, fornecendo capacidades ilimitadas para personalização e expansão. Em efeito, o servidor slapd se torna um motor RPC com uma API compacta, bem definida e ubíqua.
  - Overlays
    - Conceito geral
      - Normalmente uma solicitação LDAP é recebida pela interface, descodificada, e, em seguida, passada para o backend para o processamento. Quando o backend completa um pedido, ele retorna um resultado para o frontend, que, em seguida, envia o resultado para o cliente LDAP. Uma sobreposição (overlay) é um pedaço de código que pode ser inserido entre o frontend e o backend. Assim, é capaz de interceptar solicitações e desencadear outras ações sobre eles antes do backend recebê-los, e também pode agir da mesma forma sobre os resultados do backend antes que eles atinjam o frontend. Sobreposições tem acesso completo às APIs internas do slapd, e por isso podem invocar qualquer coisa que o frontend ou outros backends poderiam realizar. Várias sobreposições podem ser utilizadas de uma só vez, formando uma pilha de módulos entre a interface e a infra-estrutura.
      - Sobreposições fornecem um meio simples para aumentar a funcionalidade de um banco de dados sem exigir que um backend totalmente novo seja escrito, e permitir que novas funcionalidades sejam adicionadas em compatibilidade, facilmente depuráveis e sustentável em módulos. Desde a introdução do recurso de sobreposição no OpenLDAP 2.2 muitas novas sobreposições foram contribuídas por parte da comunidade OpenLDAP.

- Overlays disponíveis
  - Atualmente existem 21 overlays na distribuição principal do OpenLDAP, com outros 15 overlays na sessão de código de contribuição de usuários, e mais aguardando aprovação para inclusão.
  - Os overlays principais incluem:
    - accesslog: loga a atividade do servidor em outro banco de dados LDAP, para logging LDAP acessível
    - auditlog: loga a atividade do servidor em um arquivo de texto simples
    - chain: intercepta referências (referrals) e encadeia elas em vez; o código é parte do back-ldap
    - collect: implementa atributos coletivos de estilo X.500 (conhecido também como Classe de Serviço Netscape)
    - constraint: restringe os valores aceitáveis para atributos particulares
    - dds: serviço de dados dinâmico - entradas de vida curta e auto-expiração
    - deref: retorna informações sobre entradas referenciadas em um determinado resultado de pesquisa
    - dyngroup: simples suporte a grupo dinâmico
    - dynlist: suporte mais sofisticado a grupo dinâmico e mais
    - memberof: suporte para memberOf e atributos backlink semelhantes
    - pcache: cache de resultados de pesquisa, principalmente para melhorar o desempenho para servidores com proxy
    - ppolicy: Política de Senha LDAP - qualidade de senha, expiração, etc
    - refint: integridade referencial
    - retcode: define pré-determinados códigos de retorno para várias operações; usado para depuração de cliente
    - rwm: módulo de reescrita, para várias alterações de dados LDAP
    - seqmod: serializa escrita para entradas individuais
    - sssvlv: Ordenação do Lado do Servidor e Visualizações de Lista Virtual
    - syncprov: Provedor Syncrepl, implementa o lado do mestre de um acordo de replicação
    - translucent: passagem semitransparente, para aumentar localmente dados em um servidor proxy
    - unique: para forçar singularidade de valores de atributos dentro de uma árvore
    - valsort: mantém várias ordens de classificação para os valores de um atributo
  - Os overlays contribuídos incluem:
    - addpartial: recebe requisições Add e as transforma em Modify se a entrada de destino já existe
    - allop: retorna todos os atributos operacionais, para os clientes que não sabem como os pedir
    - autogroup: grupos estáticos geridos de forma dinâmica
    - cloack: esconde atributos, a menos que tenha sido expressamente solicitado em uma busca
    - denyop: rejeita pedidos arbitrariamente configurados

        - dupent: retorna resultados de valor múltiplos como entradas separadas
        - lastbind: grava o timestamp da última autenticação bem-sucedida de um usuário
        - lastmod: mantém o timestamp da última mudança dentro de uma árvore
        - nops: filtrar modificações redundantes
        - noopsrch: conta as entradas que seriam retornadas por uma pesquisa
        - nssov: responde requisições NSS e PAM diretamente no slapd, substitui o nss-ldap e pam-ldap
        - proxyOld: suporta uma codificação obsoleta de ProxyAuthz usado pela Sun et al.
        - smbk5pwd: mantém senhas Samba e Kerberos
        - trace: loga cada solicitação LDAP e resposta
        - usn: Números de Atualização de Seqüência (como no Microsoft AD, ainda não lançado)
- Outros módulos
    - Backends e overlays são os dois tipos mais usados de módulos. Backends foram tipicamente incorporados ao binário slapd, mas também podem ser construídos como módulos carregados dinamicamente, e overlays são geralmente construídos como módulos dinâmicos. Além disso, o slapd suporta módulos dinâmicos para implementação de novas sintaxes LDAP, regras correspondentes, controles e operações estendidas, bem como para a implementação de mecanismos de controle de acesso personalizados e mecanismos de hash de senha.
    - O OpenLDAP também suporta SLAPI, a arquitetura de plugin usada pela Sun e Netscape/Fedora/Red Hat. Nas versões atuais, o framework SLAPI é implementado dentro de um overlay slapd. Enquanto muitos plugins escritos para o Sun/Netscape/Fedora/Red Hat são compatíveis com o OpenLDAP, muito poucos membros da comunidade OpenLDAP usam o SLAPI.
    - Módulos disponíveis
        - Módulos slapd nativos
            - acl/posixgroup - suporte a filiação PosixGroup em controles de acesso
            - comp_match - suporte a correspondência baseada em componente
            - kinit - mantém/atualiza um Kerberos TGT para slapd
            - passwd/ - mecanismos de hash de senha adicionais. Atualmente inclui Kerberos, Netscape, RADIUS e SHA2.
        - Plugins SLAPI
            - addrdnvalue - adiciona valor RDN para uma entrada se ele foi omitido em uma solicitação Add
- Sumário de lançamento
    - Os principais (funcionais) lançamentos do OpenLDAP Software incluem:
        - OpenLDAP versão 1 foi um clean-up geral do último lançamento do projeto da Universidade de Michigan (versão 3.3), e consolidação de alterações adicionais.
        - OpenLDAP versão 2.0, lançado em Agosto de 2000, incluiu grandes melhorias, incluindo suporte ao LDAP versão 3 (LDAPv3), suporte ao Internet Protocol versão 6 (IPv6), e inúmeras outras melhorias.

- OpenLDAP versão 2.1, lançado em Junho de 2002, incluiu o backend de banco de dados transacional (com base no banco de dados Berkeley ou BDB), suporte ao Simple Authentication and Security Layer (SASL), e Meta, Monitor, e backends virtuais experimentais.
- OpenLDAP versão 2.2, lançado em Dezembro de 2003, incluiu o motor LDAP "sync" com suporte a replicação (syncrepl), a interface de overlay, e numerosos banco de dados e melhorias funcionais relacionadas com a RFC.
- OpenLDAP versão 2.3, lançado em Junho de 2005, incluiu o backend de configuração (configuração dinâmica), sobreposições adicionais, incluindo o software Política de Senha conforme com a RFC, e inúmeras melhorias adicionais.
- OpenLDAP versão 2.4, lançado em Outubro de 2007, introduziu replicação N-way MultiMaster, mestre Stand-by, e a capacidade de apagar e modificar elementos de esquema em tempo real, além de muitos outros.

    - Replicação
        - O OpenLDAP suporta replicação usando a sincronização de conteúdo, conforme especificado na RFC 4533. Essa especificação é referenciada depois como "syncrepl". Em adição à especificação de base, uma melhoria conhecida como delta-syncrepl também é suportada. Melhorias adicionais foram implementadas para suportar a replicação multi-mestre.
        - syncrepl
            - A operação básica de sincronização é descrita na RFC 4533. O protocolo é definido de tal forma que uma base de dados persistente de mudanças não é necessária. Pelo contrário, o conjunto de alterações está implícito através da informação do número de seqüência mudança (CSN), armazenada em cada entrada e otimizada, através de um log de sessão opcional que é particularmente útil para rastrear exclusões recentes.
        - delta-syncrepl
            - Esse protocolo mantém um banco de dados persistente dos acessos de gravação (mudanças) e podem representar cada modificar precisamente (que significa apenas os atributos que foram alterados). Ele ainda é construído sobre a especificação syncrepl padrão, que sempre envia mudanças como entradas completas. Mas no delta-syncrepl, as entradas transmitidas são realmente enviadas a partir de um banco de dados de log, onde cada mudança na base de dados principal é registrada como uma entrada de registro. As entradas de log são gravadas através do Esquema de Log LDAP.

- Configuração
    - /etc/openldap/
        - slapd.conf(5) - arquivo de configuração para o slapd, o stand-alone LDAP daemon
        - slapd.d/(slapd-config(5)) - diretório de configuração usado pelo backend slapd-config
- Utilitários
    - slapd(8) - Stand-alone LDAP Daemon
        - Exemplos
            - slapd - incia o daemon LDAP stand-alone
            - slapd -4 - escuta em endereços IPv4 apenas
            - slapd -6 - escuta em endereços IPv6 apenas

- slapd -T <ferramenta> - executa no modo ferramenta (slapadd, slapcat, slapindex …)
- slapd -d <nível de depuração> - habilita a depuração como definido pelo nível de depuração
- slapd -f <arquivo> - especifica o arquivo de configuração (padrão: /etc/openldap/slapd.conf)
- slapd -F <diretório> - especifica o diretório de configuração (padrão: /etc/openldap/slapd.d/). Se ambas opções são especificadas, o arquivo de configuração será lido e convertido para formato de configuração de diretório, e gravado no diretório especificado
- slapd -h <lista de URL> - especifica a lista de URL que será servida pelo daemon (padrão: ldap:///)
- slapd -r <diretório> - especifica o diretório para ser usado como uma jaula chroot
- slapd -u <usuário> - especifica o usuário com o qual o daemon será executado
- slapd -g <grupo> - especifica o grupo com o qual o daemon será executado
- slapd -c <cookie> - provê um cookie para o consumidor de replicação syncrepl

- slapadd(8) - Adiciona entradas para um banco de dados SLAPD
  - Exemplos
    - slapadd - adiciona entradas para um banco de dados SLAPD, lendo da entrada padrão
    - slapadd -l <ldif> - especifica o arquivo ldif a ser lido
    - slapadd -b <sufixo> - especifica a base pelo sufixo
    - slapadd -n <número> - especifica a base pelo número (base 0 identifica o backend slapd-config)
    - slapadd -c - modo de operação contínua (ignora erros)
    - slapadd -j <número da linha> - salta para a linha especificada no LDIF antes de processar qualquer entrada
    - slapadd -q - habilita o modo rápido (poucas verificações de integridade)
    - slapadd -s - desabilita a verificação de esquema
    - slapadd -u - habilita o modo de teste de execução (não escreve no backend)
    - slapadd -v - habilita o modo verbose
- slapcat(8) - Utilitário banco de dados SLAPD para LDIF
  - Exemplos
    - slapcat - escreve na saída padrão, entradas do banco de dados SLAPD
    - slapcat -l <ldif> - especifica o arquivo ldif a ser escrito
    - slapcat -b <sufixo> - especifica a base pelo sufixo
    - slapcat -n <número> - especifica a base pelo número (base 0 identifica o backend slapd-config)
    - slapcat -a <filtro> - especifica o filtro para a pesquisa de entradas
    - slapcat -c - modo de operação contínua (ignora erros)
    - slapcat -H <URI> - usa dn, escopo e filtro da URI especificada para manusear apenas as entradas correspondentes
    - slapcat -s <dn> - apenas despeja entradas na subárvore especificada pelo dn (depreciado - use ldap:///<dn> no lugar)
    - slapcat -v - habilita o modo verbose

    - slapindex(8) - reindexa entradas em um banco de dados SLAPD
        - Exemplos
            - slapindex - reindexa entradas em um banco de dados SLAPD
            - slapindex -b <sufixo> - especifica a base pelo sufixo
            - slapindex -n <número> - especifica a base pelo número
            - slapindex -q - habilita o modo rápido (poucas verificações de integridade)
            - slapindex -t - habilita o modo truncado (trunca um índice antes de indexar qualquer entrada)
            - slapindex -v - habilita o modo verbose
- Diretório de dados
    - /var/lib/ldap/(slapd(8)) - diretório dos bancos de dados SLAPD

Controle de acesso

- slapd access [2]
    - O arquivo slapd.conf consiste em uma série de opções de configurações globais que se aplicam ao slapd como um todo (incluindo todos os backends), seguido de zero ou mais definições de back-end de banco de dados que contêm informações específicas para uma instância de back-end.
    - O formato geral da slapd.conf é como se segue:
        - # comment - these options apply to every database
        - <global configuration options>
        - # first database definition & configuration options
        - database <backend 1 type>
        - <configuration options specific to backend 1>
        - # subsequent database definitions & configuration options
        - …
    - Tanto a configuração global e cada seção específica do backend pode conter informações de acesso. Diretivas de controle de acesso específicas de back-end são usadas para as entradas que pertencem ao backend, de acordo com o seu contexto de nomeação. No caso em que diretivas de controle de acesso não são definidas para um backend, ou as que estão definidas não são aplicáveis, as diretivas da seção de configuração global são então utilizadas.
    - Se não há controles de acesso presentes, a política padrão permite que qualquer um e todos poderão ler qualquer coisa, mas restringe as atualizações para o rootdn. (Por exemplo, "access to * by * read").
    - Ao lidar com uma lista de acesso, porque a lista de acesso global é efetivamente anexada a cada lista por banco de dados, se a lista resultante é não-vazia, então a lista de acesso vai terminar com uma diretiva de acesso implícito “access to * by * none”. Se não houver diretivas de acesso aplicáveis a um backend, então uma leitura padrão é usada.
    - Esteja avisado: o rootdn sempre pode ler e gravar tudo!
    - Para entradas não realizadas em qualquer backend (como uma raiz DSE), as diretivas globais são usadas.
    - Argumentos que devem ser substituídos por um texto real são mostrados entre colchetes <>.
    - A diretiva access
        - A estrutura da diretiva de controle de acesso é
            - access to <what> [by <who> [<access>] [<control>]]+

- Concede acesso (especificado por <access>) para um conjunto de entradas e/ou atributos (especificado por <what>) por um ou mais requisitantes (especificado por <who>).

- Listas de diretivas de acesso são avaliadas na ordem em que aparecem no slapd.conf. Quando uma cláusula <what> corresponde ao dado cujo acesso está sendo avaliado, a sua lista de cláusula <who> é verificada. Quando uma cláusula <who> corresponde propriedades do assessor, as suas cláusulas <access> e <control> são avaliadas. A verificação de controle de acesso para na primeira correspondência das cláusulas <what> e <who>, a menos que ditada de outra forma pela cláusula <controle>. Cada lista de cláusula <who> é implicitamente terminada por uma cláusula
  - by * none stop
- que resulta em deter o controle de acesso, sem privilégios de acesso concedidos. Cada lista de cláusula <what> é implicitamente terminado por uma cláusula
  - access to * by * none
- que resulta em concessão de sem privilégios de acesso a um dado de outra forma não especificada.

- Cláusula <what>
  - O campo <what> especifica a entidade ao qual o controle de acesso se aplica. Ela pode ter as formas:
    - dn[.<dnstyle>]=<dnpattern>
    - filter=<ldapfilter>
    - attrs=<attrlist>[val[/matchingRule][.<attrstyle>]=<attrval>]
  - com
    - <dnstyle>={{exact|base(object)}|regex|one(level)|sub(tree)|children}
    - <attrlist>={<attr>|[{!|@}]<objectClass>}[,<attrlist>]
    - <attrstyle>={{exact|base(object)}|regex|one(level)|sub(tree)|children}
  - A declaração dn=<dnpattern> seleciona as entradas com base em seu contexto de nomeação. O <dnpattern> é uma string de representação do DN da entrada. O curinga * representa todas as entradas, e está implícito se nenhuma forma dn é dada.
  - O <dnstyle> é opcional; no entanto, recomenda-se especificar para evitar ambiguidades. Base (sinônimo de baseObject), o padrão, ou exact (um apelido para base) indica a entrada cujo DN é igual ao <dnpattern>; one (sinônimo de onelevel) indica todas as entradas imediatamente abaixo do <dnpattern>, sub (sinônimo de subárvore) indica todas as entradas na subárvore no <dnpattern>, children indica todas as entradas abaixo (subordinadas ao) do <dnpattern>.
  - Se o qualificador <dnstyle> é regex, então <dnpattern> é um padrão de expressão regular POSIX ("estendido"), conforme detalhado em regex(7) e/ou re_format(7), correspondendo uma string de representação normalizada do DN de entrada. A forma regex do padrão (ainda) não permite UTF-8.
  - A declaração filter=<ldapfilter> seleciona as entradas com base em um filtro LDAP válido, como descrito na RFC 4515. Um filtro (objectClass=*) está implícito se nenhuma forma de filtro é dada.
  - A declaração attrs=<attrlist> seleciona os atributos aos quais a regra de controle de acesso se aplica. É uma lista de tipos de atributos, além da entrada de nomes especial separados por vírgula, indicando o acesso à próprio entrada e, em crianças, indicando o acesso às crianças da entrada. Nomes objectClass também podem ser

especificados na lista, o que afetará todos os atributos que são necessários e/ou permitidos pelos objectClass. Na verdade, nomes <attrlist> que são prefixados por @, são tratados diretamente como nomes objectClass. Um nome prefixado por !, também é tratado como um objectClass, mas neste caso a regra de acesso afeta os atributos que não são necessários, e nem permitidos, por esse objectClass. Se nenhuma forma attrs é dada, attrs=@extensibleObject está implícito, ou seja, todos os atributos são abordados.

- Usando as formas attrs=<attr> val[/matchingRule][.<attrstyle>]=<attrval> especifica o acesso a um determinado valor de um único atributo. Neste caso, apenas um único tipo de atributo pode ser dado. O <attrstyle> exact (o padrão) usa a regra de correspondência de igualdade do atributo para comparar o valor, a menos que uma regra diferente (e compatível) correspondendo é especificada. Se o <attrstyle> é regex, o valor fornecido é utilizado como um padrão de expressão regular POSIX ("estendido"). Se o atributo tiver a sintaxe DN, o <attrstyle> pode ser qualquer de base, onelevel, sub ou children, resultando em correspondência de base, onelevel, sub ou children, respectivamente.
- As declarações dn, filter e attrs são aditivas; eles podem ser usadas em sequência para selecionar as entidades as quais a regra de acesso se aplica, com base no contexto de nomeação, o valor e o tipo de atributo simultaneamente. Subcorrespondências resultantes da correspondência regex podem ser desreferenciadas no campo <who> usando a sintaxe ${v<n>}, onde <n> é o número da subcorrespondência. A sintaxe padrão, $<n>, é realmente um apelido para ${d<n>}, que corresponde a desreferenciando subcorrespondências a partir da porção dnpattern do campo <what>.

- Cláusula <who>
  - O campo <who> indica a quem a regra de acesso se aplica. Múltiplas declarações <who> podem aparecer em uma declaração de controle de acesso, indicando os diferentes privilégios de acesso ao mesmo recurso, que aplica a diferentes acessos. Ele pode ter uma das formas:
    - *
    - anonymous
    - users
    - self[.<selfstyle>]
    - dn[.<dnstyle>[,<modifier>]]=<DN>
    - dnattr=<attrname>
    - realanonymous
    - realusers
    - realself[.<selfstyle>]
    - realdn[.<dnstyle>[,<modifier>]]=<DN>
    - realdnattr=<attrname>
    - group[/<objectclass>[/<attrname>]][.<groupstyle>]=<group>
    - peername[.<peernamestyle>]=<peername>
    - sockname[.<style>]=<sockname>
    - domain[.<domainstyle>[,<modifier>]]=<domain>
    - sockurl[.<style>]=<sockurl>
    - set[.<setstyle>]=<pattern>
    - ssf=<n>
    - transport_ssf=<n>

        - tls_ssf=<n>
        - sasl_ssf=<n>
        - dynacl/<name>[/<options>][.<dynstyle>][=<pattern>]
    - com
        - <style>={exact|regex|expand}
        - <selfstyle>={level{<n>}}
        - <dnstyle>={{exact|base(object)}|regex|one(level)|sub(tree)|children|level{<n>}}
        - <groupstyle>={exact|expand}
        - <peernamestyle>={<style>|ip|ipv6|path}
        - <domainstyle>={exact|regex|sub(tree)}
        - <setstyle>={exact|expand}
        - <modifier>={expand}
        - <name>=aci <pattern>=<attrname>]
    - Elas podem ser especificados em combinação.
    - O curinga * refere-se a todos
    - As palavras-chave prefixados pelo verdadeiro ato como os seus homólogos sem prefixo; a verificação ocorre, respectivamente, com a autenticação DN e a autorização DN.
    - A palavra-chave anonymous significa que o acesso é concedido a clientes não autenticados; ela é usado principalmente para limitar o acesso aos recursos de autenticação (por exemplo, o atributo userPassword) para clientes não autenticados, para fins de autenticação.
    - A palavra-chave users significa que o acesso é concedido a clientes autenticados.
    - A palavra-chave self significa que acesso a uma entrada é permitida para a entrada em si (por exemplo, a entrada que está sendo acessada e solicitando a entrada, devem ser a mesma). Ela permite que o estilo level{<n>}, em que <n> indica o antepassado do DN, é para ser utilizado em correspondências. Um valor positivo indica que o ancestral <n>-th do DN do usuário deve ser considerado; um valor negativo indica que o <n>-ésimo antepassado do alvo deve ser considerado. Por exemplo, uma cláusula "by self.level{1} ..." iria corresponder quando o objeto "dc=example,dc=com" é acessado por "cn=User,dc=example,dc=com". A cláusula "by self.level{-1} ..." corresponde quando o mesmo usuário acessa o objeto "ou=Address Book,cn=User,dc=example,dc=com".
    - A declaração dn=<DN> significa que o acesso é concedido ao DN correspondente. O qualificador de estilo opcional dnstyle permite as mesmas escolhas da forma dn do campo <what>. Além disso, o estilo regex pode explorar substituição de substring das subcorrespondências na cláusula <what> dn.regex, usando a forma $<dígito>, com dígitos que variam de 0 a 9 (onde 0 corresponde à string inteira), ou no formato ${<dígito>+}, para subcorrespondências superiores a 9. Substituição de substring do valor do atributo pode ser feito usando a forma ${v<dígito>+}. Desde que o caractere cifrão é usado para indicar uma substituição de substring, o caractere cifrão que é usado para indicar a correspondência até o fim da string, deve ser precedido por um segundo caractere cifrão, por exemplo,
        - access to dn.regex="^(.+,)?uid=([^,]+),dc=[^,]+,dc=com$" by dn.regex="^uid=$2,dc=[^,]+,dc=com$$" write
    - O qualificador de estilo permite um modificador opcional. Atualmente, o único tipo permitido é expand, que faz com que a substituição de substring de das

subcorrespondências ocorra mesmo se o dnstyle não é regex. Note que a expressão regular dnstyle no exemplo acima pode ser de uso apenas se a cláusula <by> precisa ser um regex; caso contrário, se o valor da segunda porção (a partir da direita) dc= do DN no exemplo acima foram fixadas, a forma

- access to dn.regex="^(.+,)?uid=([^,]+),dc=example,dc=com$" by dn.exact,expand="uid=$2,dc=example,dc=com" write

- pode ser usada; se tivesse de corresponder ao valor na cláusula <what>, a forma
  - access to dn.regex="^(.+,)?uid=([^,]+),dc=([^,]+),dc=com$" by dn.exact,expand="uid=$2,dc=$3,dc=com" write
- pode ser usada.
- Formas da cláusula <what>, que não sejam regex, podem fornecer subcorrespondências também. As formas base(object), sub(tree), one(level), e children fornecem $0 como correspondência de toda a string. As formas sub(tree), one(level), e children também fornecer $1 como correspondência da parte mais à direita do DN, tal como definido na cláusula <what>. Isso pode ser útil, por exemplo, para proporcionar o acesso a todos os antepassados de um utilizador através da definição
  - access to dn.subtree="dc=com" by dn.subtree,expand="$1" read
- o que significa que somente o acesso a entradas que aparecem no DN da cláusula <by> é permitida.
- A forma level{<n>} é uma extensão e uma generalização da forma onelevel, o que corresponde a todos os DNs cujos ancestrais <n>-th é o padrão de correspondência. Assim, o level{1} é equivalente a onelevel, e level{0} é equivalente a base.
- É perfeitamente inútil dar quaisquer privilégios de acesso a um DN que corresponde exatamente ao rootdn do banco de dados que as ACLs se aplicam, porque implicitamente possui privilégios de escrita para toda a árvore de banco de dados. Na verdade, o controle de acesso é ignorado para o rootdn, para resolver o problema intrínseco da galinha e do ovo.
- A declaração dnattr=<attrname> significa que o acesso é concedido as requisições cujo o DN está listado na entrada que está sendo acessada sob o atributo <attrname>.
- A declaração group=<group> significa que o acesso é concedido as requisições cujo o DN está listado na entrada de grupo cujo o DN é dado por <group>. Os parâmetros opcionais <objectclass> e <attrname> definem a classe de objeto e o tipo de atributo membro, da entrada de grupo. Os padrões são groupOfNames e member, respectivamente. O qualificador de estilo opcional <style> pode ser expand, o que significa que <group> será expandido como uma string de substituição (mas não como uma expressão regular) de acordo com regex(7) e/ou re_format(7), e exact, que significa que a correspondência exata será usada. Se o estilo da porção DN da cláusula <what> é regex, as subcorrespondências são disponibilizadas de acordo com regex(7) e/ou re_format(7); outros estilos fornecem subcorrespondências limitadas como discutido acima, sobre a forma DN da cláusula <by>.
- Para grupos estáticos, o attributeType especificado deve ter a sintaxe DistinguishedName ou NameAndOptionalUID. Para grupos dinâmicos, o attributeType deve ser um subtipo do tipo de atributo labeledURI. Somente URIs LDAP de forma ldap:///<base>??<scope>?<filter> serão avaliadas em um grupo dinâmico, procurando apenas o servidor local.

- As declarações peerName=<peername>, sockname=<sockname>, domain=<domain>, e sockurl=<sockurl> significam que IP do host contactante (no formato IP=<ip>:<porta> para IPv4, ou IP=[<ipv6>]:<porta> para IPv6) ou o nome do arquivo de pipe nomeado do host contactante (na forma PATH=<path> se conectando através de um pipe nomeado) para peername, o nome do arquivo de pipe nomeado para sockname, o nome do host para contactante para domain, e a URL do contatante com sockurl, são comparados contra o padrão de correspondência para determinar o acesso. As mesmas regras de estilo para padrão de correspondência descritas para o caso de grupo aplicam-se, além do estilo regex, o que implica subcorrespondência expand e correspondência de regex dos parâmetros de conexão correspondentes. O estilo exact da cláusula <peername> (o padrão) implica uma correspondência case-exact no IP do cliente, incluindo o prefixo IP=e o restante :<port>, ou caminho do cliente, incluindo o prefixo PATH= se conectando através de um pipe nomeado. O estilo especial ip interpreta o padrão como <peername>=<ip>[%<mask>][{<n>}], onde <ip> e <mask> são representações numéricas pontilhadas do IP e da máscara, enquanto <n>, delimitado por chaves, é uma porta opcional. O mesmo se aplica aos endereços IPv6 quando o estilo especial ipv6 é usado. Ao verificar os privilégios de acesso, a parte IP do peername é extraída, eliminando o prefixo IP= e a parte :<port>, e é comparado com a porção <ip> do padrão após o mascaramento com <mask>: ((peername & <mask>) == <ip>). Como exemplo, peername.ip=127.0.0.1 e peername.ipv6=::1 permite conexões somente de localhost, peername.ip=192.168.1.0%255.255.255.0 permite conexões de qualquer IP no domínio classe C 192.168.1, e peername.ip=192.168.1.16%255.255.255.240{9009} permite conexões de qualquer IP na faixa 192.168.1.[16-31] do mesmo domínio, somente se a porta 9009 é usada. O estilo especial path elimina o prefixo PATH= do peername quando conectando através de um pipe nomeado, e realiza uma correspondência exata no padrão de correspondência dado. A cláusula <domain> também permite que o estilo subtree, que sucede quando um nome totalmente qualificado corresponde exatamente ao padrão domain, ou a sua parte à direita, após um ponto, corresponde exatamente ao padrão domain. O estilo expand é permitido, o que implica uma correspondência exata com a expansão de subcorrespondência; o uso do expand como um modificador de estilo é considerado mais apropriado. Como exemplo, domain.subtree=example.com irá corresponder www.example.com, mas não corresponderá www.anotherexample.com. O domínio do host contactante é determinado através da realização de uma pesquisa reversa no DNS. Como essa pesquisa pode ser facilmente falsificada, o uso da declaração de domínio é fortemente desencorajado. Por padrão, as pesquisas reversas estão desativadas. O qualificador domainstyle, opcional da cláusula <domain>, permite uma opção modificadora; o único valor suportado atualmente é expand, o que faz com que a substituição de substring de subcorrespondências ocorra mesmo se o domainstyle não é regex, bem como o uso de análogos na cláusula <dn>.
- A declaração set=<pattern> não está documentada ainda.
- A declaração dynacl/<name>[/<options>] [.<dynstyle>] [=<pattern>] significa que a verificação de acesso é delegada ao método definido pelo administrador indicado pelo <name>, que pode ser registrado em tempo de execução, por meio da declaração moduleload. Os campos <options>, <dynstyle> e <pattern> são

opcionais, e são diretamente passados para a rotina de análise registrada. Dynacl é experimental; ele deve ser ativado em tempo de compilação.

- A declaração dynacl/aci[=<attrname>] significa que o controle de acesso é determinado pelos valores no attrname da própria entrada. O opcional <attrname> indica que tipo de atributo contém as informações ACI na entrada. Por padrão, o atributo operacional OpenLDAPaci é usado. ACIs são experimentais; elas devem ser ativadas em tempo de compilação.
- As declarações ssf=<n>, transport_ssf=<n>, tls_ssf=<n>, e sasl_ssf=<n> definem o mínimo exigido Fator de Força de Segurança (SSF) necessários para conceder acesso. O valor deve ser inteiro positivo.

- Cláusula <access>
  - O campo opcional <access> ::= [[real]self]{<level>|<priv>} determina o nível de acesso ou os privilégios de acesso específicos que o campo <who> terá. Os seu componente são definidos como
    - <level> ::= none|disclose|auth|compare|search|read|{write|add|delete}|manage
    - <priv> ::= {=|+|-}{0|d|x|c|s|r|{w|a|z}|m}+
  - O modificador self permite operações especiais como ter um certo nível de acesso ou privilégio apenas no caso em que a operação envolve o nome do usuário que está solicitando o acesso. Isso implica que o usuário que solicita o acesso é autorizado. O modificador realself refere-se ao DN autenticado em oposição ao DN autorizado do modificador self. Um exemplo é o acesso selfwrite para o atributo de membro de um grupo, que permite adicionar/excluir seu próprio DN da lista de membros de um grupo, apesar de não serem autorizados a afetar outros membros.
  - O modelo de acesso de nível baseia-se numa interpretação incremental dos privilégios de acesso. Os possíveis níveis são none, disclose, auth, compare, search, read, write, and manage. Cada nível de acesso implica todas as precedentes, assim manage concede todo o acesso, incluindo o acesso administrativo. O acesso de escrita é realmente a combinação de adicionar e excluir, que, respectivamente, restringir o privilégio de gravação para adicionar ou excluir o especificado <what>.
  - O nível de acesso none desautoriza todos os acessos incluindo a divulgação em erro.
  - O nível de acesso disclose permite a divulgação de informações em caso de erro.
  - O nível de acesso auth significa que é permitido o acesso a um atributo para executar operações de autenticação/autorização (por exemplo, bind) sem outro acesso. Isso é útil para conceder os clientes não autenticados o nível de acesso mínimo possível, a recursos críticos, como senhas.
  - O modelo de acesso priv baseia-se na definição explícita de privilégios de acesso para cada cláusula. O sinal = redefine acessos previamente definidos; como consequência, os privilégios de acesso finais serão apenas aqueles definidos pela cláusula. Os sinais + e - adicionam/removem os privilégios de acesso para os já existentes. Os privilégios são m para manage, w para write, a para add, z para delete, r para read, s para search, c para compare, x para auth, e d para disclose. Mais de um dos privilégios acimas podem ser adicionados em uma declaração. 0 indica que não há privilégios e é usado apenas por si só (por exemplo, 0). Note que +az é equivalente a + w.
  - Se nenhum acesso é dado, o padrão é +0.

- Cláusula <control>
  - O campo opcional <control> controla o fluxo de aplicação de regra de acesso. Ele pode ter as formas
    - stop
    - continue
    - break
  - onde stop, o padrão, significa que a verificação de acesso pára em caso de correspondência. As outras duas formas são usadas para manter em processamento cláusulas de acesso. Em detalhe, a forma continue permite outras cláusulas <who> na mesma cláusula <acccess> serem consideradas, de modo que elas podem resultar em incrementalmente alterar os privilégios, enquanto a forma break permite outras cláusulas <access> que correspondam ao mesmo alvo, serem processadas. Considere o exemplo (bobo)
    - access to dn.subtree="dc=example,dc=com" attrs=cn by * =cs break
    - access to dn.subtree="ou=People,dc=example,dc=com" by * +r
  - que permite pesquisar e comparar privilégios para todo mundo sob a árvore "dc=example,dc=com", com a segunda regra que permite também ler na subárvore "ou=People", ou o exemplo (ainda mais bobo)
    - access to dn.subtree="dc=example,dc=com" attrs=cn
    - by * =cs continue
    - by users +r
  - que concede todos pesquisar e comparar privilégios, e acrescenta privilégios de leitura para clientes autenticados.
  - Uma aplicação útil é conceder privilégios de gravação facilmente a um updatedn que é diferente do rootdn. Neste caso, uma vez que updatedn necessita de acesso de escreita para (quase) todos os dados, pode-se usar
    - access to *
    - by dn.exact="cn=The Update DN,dc=example,dc=com" write
    - by * break
  - como a primeira regra de acesso. Como consequência, a menos que a operação é realizada com a identidade updatedn, o controle é passado para a linha regras subsequentes.

- Exemplos [3]
  - Um simples exemplo:
    - access to * by * read
  - Essa diretiva de acesso concede acesso de leitura para todos.
    - access to *
      by self write
      by anonymous auth
      by * read
  - Essa diretiva permite que o usuário modifique sua entrada, permite autenticação anônima contra essas entradas, e permite que todos os outros leiam essas entradas. Note-se que apenas a primeira cláusula by <who> que corresponde aplica. Assim, os usuários anônimos são concedidos auth, não read. A última cláusula poderia muito bem ter sido "by users read".
  - O exemplo a seguir mostra o uso de um especificadores de estilo para selecionar as entradas por DN em duas diretivas de acesso onde ordenação é significativa.

        - access to dn.children="dc=example,dc=com" by * search
        - access to dn.children="dc=com" by * read
    - O acesso leitura é concedido para as entradas sob a subárvore dc=com, exceto para aquelas entradas sob a subárvore dc=example,dc=com, aos quais o acesso concedido é pesquisa. Nenhum acesso é concedido para dc=com como nenhuma diretiva de acesso corresponde a esse DN. Se a ordem dessas diretivas de acesso fossem invertida, a diretiva final nunca seria alcançada, uma vez que todas as entradas sob dc=example,dc=com, também estão sob entradas dc=com.

Nomes distintos
- O DN (nome distinto) deve ser exclusivo na DIT (Árvore de Informação de Diretório).
- Um DN é composto de uma série de RDNs (nomes distintos relativos) encontrados ao subir a árvore (DIT) até sua raiz (ou sufixo ou base) e é escrito da esquerda para a direita.

Operações de mudanças de tipo
- Adicionar - changetype: add
- Modificar - changetype: modify
- Deletar - changetype: delete
- Renomear (Modrdn) - changetype: modrdn

Esquemas e páginas brancas
- Esquemas [4]
    - Os conteúdos das entradas de uma subárvore são regidas por um esquema de diretório, um conjunto de definições e restrições relativas à estrutura da árvore de informação de diretório (DIT).
    - O esquema de um servidor de diretório define um conjunto de regras que governam os tipos de informação que o servidor pode conter. Ele tem um certo número de elementos, incluindo:
        - Sintaxes de Atributo - Fornece informações sobre o tipo de informação que pode ser armazenado em um atributo.
        - Regras de Correspondência - Fornece informações sobre como fazer comparações contra valores de atributos.
        - Uso de Regras de Correspondência - Indica que tipos de atributos podem ser usado em conjunto com uma regra de correspondência especial.
        - Tipos de Atributo - Define um identificador de objeto (OID) e um conjunto de nomes que podem ser usados para se referir a um determinado atributo, e associa aquele atributo com uma sintaxe e um conjunto de regras de correspondência.
        - Classes de Objeto - Define coleções nomeadas de atributos e as classifica em conjuntos de atributos obrigatórios e opcionais.
        - Formas de Nome - Define regras para o conjunto de atributos que devem ser incluídos no RDN para uma entrada.
        - Regras de Conteúdo - Define restrições adicionais sobre as classes de objetos e atributos que podem ser usados em conjunto com uma entrada.
        - Regra de Estrutura - Define regras que governam os tipos de entradas subordinadas que uma dada entrada pode ter.

- Os atributos são os elementos responsáveis pelo armazenamento de informações em um diretório, e o esquema define as regras para quais atributos podem ser usados em uma entrada, os tipos de valores que esses atributos podem ter, e como os clientes podem interagir com esses valores.
- Os clientes podem aprender sobre os elementos do esquema que o servidor suporta recuperando um apropriado subschemaSubentry.
- O esquema define as classes de objetos. Cada entrada deve ter um atributo objectClass, que contém as classes nomeadas definidas no esquema. A definição do esquema das classes de uma entrada define que tipo de objeto uma entrada pode representar - por exemplo, uma pessoa, organização ou domínio. As definições de classe de objeto também definem a lista de atributos que devem conter os valores e a lista dos atributos que podem conter valores.
- Por exemplo, uma entrada que representa uma pessoa pode pertencer à classe "top" e "person". A associação da classe "person" exigiria a entrada conter os atributos "sn" e "cn", e permitir a entrada também conter "userPassword", "telephoneNumber", e outros atributos. Desde que entradas podem ter vários valores objectClasses, cada entrada tem um complexo de conjuntos de atributos opcionais e obrigatórios formados a partir da união das classes de objeto que ele representa. ObjectClasses podem ser herdadas, e uma única entrada pode ter vários valores objectClasses que definem os atributos da próprio entrada, disponíveis e necessários. Um paralelo com o esquema de uma objectClass é uma definição de classe e uma instância na programação orientada a objeto, que representa objectClass LDAP e entrada LDAP, respectivamente.
- Servidores de diretório podem publicar o esquema do diretório controlando uma entrada em uma base DN dada pelo atributo operacional subschemaSubentry da entrada. (Um atributo operacional descreve a operação do diretório em vez de informações do usuário e só é devolvido a partir de uma pesquisa, quando explicitamente solicitado.)
- Os administradores do servidor podem adicionar entradas de esquema adicionais para além dos elementos de esquema fornecidos. Um esquema para representar as pessoas individuais dentro das organizações é denominado um esquema de páginas brancas.

- Páginas brancas [5]
  - Um esquema de páginas brancas é um modelo de dados, especificamente um esquema lógico, para organizar os dados contidos nas entradas em um serviço de diretório, banco de dados ou aplicativo, como um livro de endereços. Em um diretório de páginas brancas, cada entrada representa tipicamente um indivíduo que faz uso de recursos de rede, como por receber e-mails ou de ter uma conta para fazer logon em um sistema. Em alguns ambientes, o esquema pode também incluir a representação das divisões organizacionais, papéis, grupos e dispositivos. O termo é derivado das páginas brancas, a lista de indivíduos em uma lista telefônica, normalmente classificadas por localização do indivíduo em casa (por exemplo, cidade) e, em seguida, pelo seu nome.
  - Enquanto muitos provedores de serviços de telefonia têm por décadas publicado uma lista de seus assinantes em uma lista telefônica, e da mesma forma corporações publicaram uma lista de seus empregados em um diretório interno, não foi até o surgimento de sistemas de correio eletrônico que a exigência de normas para a troca eletrônica de informações entre os diferentes sistemas de assinantes apareceu.
  - Um esquema de páginas brancas tipicamente define, para cada objeto do mundo real que está sendo representado:
    - que atributos desse objeto estão a ser representados na entrada para esse objeto
    - que relações desse objeto a outros objetos são para ser representados

        - como a entrada será nomeada em um DIT
        - como uma entrada será localizada por um cliente a procurando
        - como as entradas similares devem ser distinguidas
        - como as entradas serão ordernadas quando exibidas em uma lista
    - Uma das primeiras tentativas de padronizar um esquema de páginas brancas para uso de correio eletrônico estavam no X.520 e X.521, parte das especificações X.500, que foi derivado dos requisitos de endereçamento do X.400 e definiu uma Árvore de Informação de Diretório que espelhava o sistema de telefonia internacional, com entradas representando os assinantes residenciais e organizacionais. Isso evoluiu para o esquema padrão Lightweight Directory Access Protocol na RFC 2256. Um dos mais implantados esquemas de Páginas Brancas utilizados no LDAP para representar os indivíduos em um contexto organizacional é inetOrgPerson, definido na RFC 2798, embora as versões do Active Directory exigem uma classe de objeto diferente, User. Muitas grandes organizações também definiram os seus próprios esquemas de páginas brancas para seus empregados ou clientes, como parte de sua arquitetura de gerenciamento de identidade. Conversão entre bases de dados e diretórios que usam esquemas diferentes é muitas vezes a função de um Meta-Diretório, e normas para o intercâmbio de dados, tais como Protocolo de Indexação Comum.
    - Algumas implementações de diretório no início sofreram devido a escolhas de design pobre em seu esquema de páginas brancas, tais como:
        - atributos utilizados para fins de nomeação eram não-exclusivo em grandes ambientes (tais como nome comum de uma pessoa)
        - atributos utilizados para fins de nomeação eram susceptíveis de alterar (como apelidos)
        - atributos foram incluídos o que poderia levar ao roubo de identidade, como um número de segurança social
        - usuários eram obrigados durante o provisionamento de escolher atributos que são únicos, mas ainda memoráveis para eles
    - Existem numerosos outros esquemas propostos, quer como definições independentes adequados para uso com pastas de uso geral, ou como incorporados em protocolos de rede.
    - Exemplos de outros esquemas genéricos de páginas brancas incluem vCard, definido na RFC 2426, e FOAF.

Diretórios

- Diretórios são definidos pelo instanciamento de um banco de dados no OpenLDAP
- Cada banco de dados deve ter pelo menos um sufixo
- Exemplo:
    - database bdb
    - suffix "dc=example,dc=com"
    - rootdn "cn=manager,dc=example,dc=com"
    - rootpw secretpw
    - directory /var/lib/ldap/example.com
    -
    - database bdb
    - suffix "dc=mycompany,dc=com"
    - rootdn "cn=manager,dc=mycompany,dc=com"
    - rootpw secretpw

    - directory /var/lib/ldap/mycompany.com

IDs de objetos, atributos e classes

- Identificadores de objetos [6]
    - Cada elemento do esquema é identificado por um identificador de objeto exclusivo (OID). Os OIDs também são usados para identificar outros objetos. Eles são comumente encontrados em protocolos descritos pelo ASN.1. Em particular, eles são muito utilizados pelo Simple Network Management Protocol (SNMP). Como OIDs são hierárquicos, uma organização pode obter um OID e ramificá-lo conforme necessário. Por exemplo, se a uma organização foi atribuído OID 1.1, pode-se ramificar a árvore como se segue:
        - OID Atribuição
        - 1.1 OID da Organização
        - 1.1.1 SNMP Elements
        - 1.1.2 LDAP Elements
        - 1.1.2.1 AttributeTypes
        - 1.1.2.1.1 x-my-Attribute
        - 1.1.2.2 ObjectClasses
        - 1.1.2.2.1 x-my-ObjectClass
    - É livre para se projetar uma hierarquia adequada às necessidades organizacionais sob o OID da organização. Não importa qual hierarquia se escolhe, deve-se manter um registro de trabalhos que se faz. Isso pode ser um arquivo plano simples ou algo mais sofisticado, como o Registro OID OpenLDAP.

Consciência do Daemon de Serviços de Segurança de Sistemas (SSSD)

- SSSD [7]
    - SSSD é um daemon do sistema. Sua função principal é fornecer o acesso à identidade e recursos remotos de autenticação através de um framework comum que pode fornecer armazenamento em cache e suporte offline para o sistema. Ele fornece módulos do PAM e NSS, e no futuro, interfaces baseadas no D-BUS para informações estendidas do usuários. Ele também fornece um banco de dados para melhor armazenar os usuários locais, bem como os dados do usuário estendidos.

Termos e utilitários

- slapd - (8) Stand-alone LDAP Daemon
- slapd.conf - (5) arquivo de configuração para o slapd, o stand-alone LDAP daemon
- LDIF - (5) Formato de Intercâmbio de Dados LDAP
- slapadd - (8) Adiciona entradas para um banco de dados SLAPD
- slapcat - (8) Utilitário banco de dados SLAPD para LDIF
- slapindex - (8) reindexa entradas em um banco de dados SLAPD
- /var/lib/ldap/ - (8) diretório dos bancos de dados SLAPD
- loglevel - nível de detalhamento de registros

Referências

1. http://en.wikipedia.org/wiki/OpenLDAP
2. slapd.access(5)

3. http://www.openldap.org/doc/admin24/access-control.html
4. http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
5. http://en.wikipedia.org/wiki/White_pages_schema
6. http://www.openldap.org/doc/admin24/schema.html
7. https://fedorahosted.org/sssd/
8. https://fedorahosted.org/sssd/wiki/HOWTO_Configure_1_0_2

Exercícios práticos

1. OpenLDAP
    a. Varrer o conteúdo do diretório de configuração do OpenLDAP (Ex.: find <diretório>)

    b. Varrer o conteúdo do diretório de dados do OpenLDAP (Ex.: find <diretório>)

2. Controle de acesso
    a. Listar as diretivas de controle de acesso (atributo olcAccess) para a base dc=my-domain,dc=com, através de pesquisa LDAP, usando como URI de acesso “ldapi:///” (Ex.: ldapsearch <opções>)
    b. Criar o arquivo my-domain-acl.ldif, declarando as seguintes regras de controle de acesso (Ex.: vi <arquivo>)
        i. Entrada
            1. Nome distinto: olcDatabase={2}bdb,cn=config
            2. Adicionar: olcAccess
            3. Acesso (olcAccess): to attrs=userPassword by self read by anonymous auth by * none
            4. Acesso: to * by * read

    c. Modificar a base dc=my-domain,dc=com, usando como URI de acesso “ldapi:///” e usando o arquivo LDIF criado (Ex.: ldapmodify <opções>)

    d. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

    e. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=user1,ou=people,dc=my-domain,dc=com” e a senha “secretUser1” a e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

    f. Alterar o arquivo my-domain-acl.ldif, declarando (Ex.: vi <arquivo>)
        i. Entrada
            1. Nome distinto: olcDatabase={2}bdb,cn=config
            2. Substituir: olcAccess
            3. Acesso: to attrs=userPassword by self read by anonymous auth by * none
            4. Acesso: to * by users read by * none

    g. Modificar a base dc=my-domain,dc=com, usando como URI de acesso “ldapi:///” e usando o arquivo LDIF criado (Ex.: ldapmodify <opções>)

h. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

i. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário "cn=user1,ou=people,dc=my-domain,dc=com" e a senha "secretUser1" a e a base de pesquisa dc=my-domain,dc=com (Ex.: ldapsearch <opções>)

3. Esquemas e páginas brancas
   a. Pesquisar as entradas do LDAP, usando como URI de acesso "ldapi:///", a base de pesquisa cn=schema,cn=config, o escopo "one", e retornando apenas os nomes distintos (Ex.: ldapsearch <opções>)

   b. Retornar a entrada do LDAP cn={1}core, usando como URI de acesso "ldapi:///" e a base de pesquisa cn=schema,cn=config (Ex.: ldapsearch <opções>)

4. Diretórios
   a. Criar o diretório /var/lib/openldap-data/example.com/ (Ex.: mkdir <diretório>)

   b. Alterar recursivamente o usuário e grupo dono diretório /var/lib/openldap-data/ para ldap (Ex.: chown <opções> <objeto>)

   c. Alterar recursivamente o contexto do SELinux do diretório /var/lib/openldap-data/ para system_u:object_r:slapd_db_t:s0 (chcon -R system_u:object_r:slapd_db_t:s0 /var/lib/openldap-data/)

   d. Criar o arquivo db.example.com.ldif, declarando (Ex.: vi <arquivo>)
      i. Entrada
         1. Nome distinto: olcDatabase=hdb,cn=config
         2. Banco de dados (olcDatabase): hdb
         3. Classe de objeto: olcDatabaseConfig
         4. Sufixo (olcSuffix): dc=example,dc=com
         5. DN do gerente (olcRootDN): cn=manager,dc=example,dc=com
         6. Senha do gerente (olcRootPW): secret
         7. Diretório do banco de dados (olcDbDirectory): /var/lib/openldap-data/example.com/

   e. Adicionar a base dc=example,dc=com, usando como URI de acesso "ldapi:///" e usando o arquivo LDIF criado (Ex.: ldapadd <opções>)

   f. Varrer o conteúdo do diretório /var/lib/openldap-data/ (Ex.: find <diretório>)

   g. Criar o arquivo example.com.ldif, declarando (Ex.: vi <arquivo>)
      i. Entrada
         1. Nome distinto: dc=example,dc=com
         2. Componente de domínio: example
         3. Classe de objeto: dcObject
         4. Classe de objeto: organization

5. Organização: My Company

ii. Entrada
1. Nome distinto: ou=people,dc=example,dc=com
2. Unidade organizacional: people
3. Classe de objeto: organizationalUnit
4. Classe de objeto: top

iii. Entrada
1. Nome distinto: ou=group,dc=example,dc=com
2. Unidade organizacional: group
3. Classe de objeto: organizationalUnit
4. Classe de objeto: top

h. Adicionar as entradas no LDAP, através de mecanismo de autenticação simples, usando como credenciais para autenticação o usuário “cn=manager,dc=example,dc=com” e a senha “secret”, e o arquivo LDIF criado (Ex.: ldapadd <opções>)

i. Pesquisar as entradas do LDAP, através de mecanismo de autenticação simples, usando conexão anônima e a base de pesquisa dc=example,dc=com (Ex.: ldapsearch <opções>)

Simulado

1. … é o arquivo de configuração do OpenLDAP.

2. Para se indexar uma base manualmente, o comando … deve ser usado.

3. Os comandos … e … são usados para carregar um LDIF em uma base de dados SLAPD e converter uma base de dados SLAPD para LDIF.

4. Os comandos de manipulação de bases de dados SLAPD só podem ser executados enquanto a base estiver offline.
   a. V
   b. F

5. ACLs definidas no escopo global do servidor, são anexadas às ACLs definidas nos diretórios.
   a. V
   b. F

6. São identificadores válidos para operações:
   a. read
   b. write
   c. auth
   d. none

7. São identificadores válidos para dados (what) em uma acl:
   a. *
   b. anonymous
   c. dn=
   d. attrs

8. São identificadores válidos para cliente (who) em uma acl:
    a. attrs
    b. dn=
    c. *
    d. none

9. Durante o processo de validação de acesso, por padrão, todas as vezes que uma correspondência é feita a uma regra, o processamento das regras é encerrado.
    a. V
    b. F

10. Nomes distintos são globalmente únicos enquanto nomes relativos podem ser repetidos em uma mesma hierarquia.
    a. V
    b. F

11. Nomes distintos são formados a partir da raiz do diretório até a entrada final.
    a. V
    b. F

12. Para se renomear uma entrada no diretório, a operação … deve ser usada.

13. Para se modificar uma entrada no diretório, a operação … deve ser usada.

14. Schemas são responsáveis por definir conjunto de regras que governam os tipos de informação que o servidor pode conter.
    a. V
    b. F

15. Esquemas de páginas brancas são conjuntos de atributos de uso comum a um determinado público alvo, onde são especificados atributos aos quais usuários podem consultar informações de outros usuários.
    a. V
    b. F

16. A diretiva directory define um novo diretório no OpenLDAP.
    a. V
    b. F

17. Quais são as diretivas necessárias para se criar o diretório “dc=company,dc=com”, usando como backend o bdb e como gerente, o usuário “cn=manager,dc=company,dc=com”?
    a. backend bdb
    b. suffix dc=company,dc=com
    c. rootdn cn=manager,dc=company,dc=com
    d. directory /var/lib/ldap

18. Um diretório pode ser acessado por diferentes sufixos.
    a. V

b. F

19. Ao especificar um esquema, a hierarquia dos OIDs dos objetos do esquema é livre.
    a. V
    b. F

20. O SSSD fornece o acesso à identidade e recursos remotos de autenticação através de um framework comum.
    a. V
    b. F

Tópico 211: Serviços de Correio Eletrônico

211.1 Utilização de servidores de e-mail

Visão geral

Peso: 4
Descrição: Os candidatos devem ser capazes de gerenciar um servidor de email, incluindo a configuração de apelidos de email, cotas de email, e domínios virtuais de email. Esse objetivo também inclui configurando retransmissão interna de emails e monitorando servidores de email.
Áreas de conhecimentos chave:

- Arquivos de configuração para postfix
- Conhecimento básico do protocolo SMTP
- Consciência do sendmail e exim

Termos e utiliários:

- arquivos de configuração e comandos para postfix
- /etc/postfix/
- /var/spool/postfix/
- camada de emulação de comandos sendmail
- /etc/aliases
- logs relacionados a email no /var/log/

Áreas de conhecimentos chave

Arquivos de configuração para postfix

- Postfix [1]
    - Em computação, Postfix é um livre e de código aberto, Mail Transfer Agent (MTA), que roteia e entrega o correio eletrônico, concebido como uma alternativa para o amplamente utilizado Sendmail MTA.
    - O Postfix é liberado sob a Licença Pública IBM 1.0 que é uma licença de software livre.
    - Originalmente escrito em 1997 por Wietse Venema no IBM Thomas J. Watson Research Center e lançado pela primeira vez em Dezembro de 1998, o Postfix continua a partir de 2014 a ser desenvolvido ativamente por seu criador e outros contribuidores. O software também é conhecido por seus antigos nomes VMailer e IBM Secure Mailer.
    - Em Abril de 2014, em um estudo realizado pela e-Soft, Inc., aproximadamente 28% dos servidores de correio alcançáveis publicamente na Internet executam o Postfix.
    - Implantação típica
        - Como um servidor SMTP, o Postfix implementa uma primeira camada de defesa contra spambots e malware. Os administradores podem combinar o Postfix com outro software que fornece filtragem de spam/vírus (por exemplo, Amavisd-new), acesso ao armazenamento de mensagem (por exemplo, Dovecot), ou políticas de acesso complexas em nível de SMTP (por exemplo, postfwd, policyd-weight ou greylisting).
        - Como cliente SMTP, o Postfix implementa um motor paralelizado de entrega de email de alto desempenho. O Postfix é freqüentemente combinado com software de mailing list (como Mailman).

- Características
  - O Postfix implementa um número limitado de características no MTA, e confia em extensões de terceiros para o resto.
  - Características principais embarcadas do Postfix
    - Conformidade com os padrões de suporte para SMTPUTF8, SMTP, LMTP, criptografia STARTTLS incluindo suporte ao protocolo DANE e "perfeito" forward secrecy, autenticação SASL, encapsulamento e transformação MIME, notificações de status de entrega DSN, IPv4, e IPv6.
    - Política de acesso configurável em nível de SMTP que se adapta automaticamente a sobrecarrega.
    - Domínios "virtuais" com espaços de nomes de endereços distintos.
    - Interfaces de sistema UNIX para a apresentação de linha de comando, para entrega de comando, e para entrega direta ao armazenamento de mensagens no formato mbox e maildir.
    - Leve inspeção de conteúdo com base em expressões regulares.
    - Um grande número de mecanismos de pesquisa de banco de dados, incluindo Berkeley DB, CDB, OpenLDAP LMDB, Memcached, LDAP e múltiplas implementações de banco de dados SQL.
    - Um agendador sofisticado que implementa entregas paralelas, com estratégias configuráveis de back-off e simultaneidade.
    - Um bloqueador de zumbi escalável que reduz a carga do servidor SMTP devido a spam de botnet.
  - Características típicas de extensões do Postfix
    - Extensões Postfix usam os protocolos SMTP ou Milter (filtro de correio Sendmail) que ambos dão controle total sobre o envelope da mensagem e conteúdo, ou um protocolo baseado em texto simples que habilita políticas de controle de acesso de nível de SMTP complexos.
      - Inspeção profunda de conteúdo antes ou depois que uma mensagem é aceita na fila de correio;
      - Autenticação de correio com DMARC, DKIM, SPF, ou outros protocolos;
      - Políticas de acesso de nível de SMTP, como greylisting ou controle da frequência.
- Sistemas operacionais
  - O Postfix é executado (ou foi executado) no AIX, BSD, HP-UX, o GNU/Linux, OS X, Solaris e, de modo geral, em todos os sistemas operacionais Unix-like que são fornecidos com um compilador C e oferece um ambiente de desenvolvimento padrão POSIX. É o MTA padrão para o OS X, sistemas operacionais NetBSD e Ubuntu.
- Arquitetura
  - O Postfix é composto por uma combinação de programas do servidor que são executados em segundo plano, e os programas de cliente que são invocados por programas do usuário ou por administradores de sistema.
  - O núcleo Postfix consiste em vários dúzias de programas de servidores que são executados em segundo plano, cada um lida com um aspecto específico da entrega de e-mail. Exemplos disso são o servidor SMTP, o agendador, o reescrevedor de endereços, e o servidor de entrega local. Para fins de controle de danos, a maioria dos programas de servidor executa com privilégios reduzidos fixos, e encerram voluntariamente depois de processar um número limitado de pedidos. Para

conservar os recursos do sistema, a maioria dos programas de servidor terminam quando eles se tornam ociosos.
        - Os programas cliente executam fora do núcleo Postfix. Eles interagem com programas de servidor Postfix através de instruções de entrega de correio no arquivo ~/.forward do utilizador, e através de pequenos programas de "gate" para enviar e-mail ou para solicitar informações sobre o status da fila.
- Arquivos de configuração
    - /etc/postfix
        - access - tabela de acesso ao servidor postfix
        - header_checks - inspeção de conteúdo do postfix
        - main.cf - parâmetros de configuração específicas de site
        - master.cf - parâmetros de configuração para definir processos daemon
    - Tabelas postfix
        - canonical - formato de tabela canônica do postfix
        - generic - formato de tabela genérica do postfix
        - relocated - formato de tabela de realocação do postfix
        - transport - formato de tabela de transporte do postfix
        - virtual - formato de tabela de apelido virtual do postfix
    - Apelidos de email
        - Arquivo /etc/aliases - arquivo de apelidos
        - Manual aliases.postfix(5)
    - Cotas de email
        - Diretiva mailbox_size_limit - cota padrão para domínios diretos do Postfix
        - Diretiva virtual_mailbox_limit - cota padrão para domínios virtuais do Postfix
    - Domínios virtuais de email
        - Arquivo /etc/postfix/virtual - tabela de apelidos de domínios virtuais
    - Utilitários
        - postalias(1) - manutenção de bancos de dados de apelidos Postfix
            - Exemplos:
                - postalias <arquivo> - cria ou atualiza um banco de dados de apelidos postfix
                - postalias -i - modo incremental
                - postalias -d <chave> - pesquisa os mapas especificados pela chave e remove uma entrada por mapa
                - postalias -q <chave> - pesquisa os mapas especificados pela chave e escreve a primeira correspondência na saída padrão
                - postalias -r - quando atualizando a tabela, não reclama de tentativas de atualizar entradas existentes e faz as atualizações assim mesmo
                - postalias -s - recupera todos elementos do banco de dados e escreve uma linha para cada chave:valor na saída, para cada elemento
                - postalias -v - habilita o modo verbose
                - postalias -w - quando atualizando uma tabela, não reclama de tentativas de atualizar entradas existentes e ignora essa tentativas
        - postconf(1) - utilitário de configuração do Postfix
            - Exemplos:
                - postconf - exibe os parâmetros configurados no Postfix
                - postconf -d - exibe os valores padrões dos parâmetros do Postfix
                - postconf -e - edita o arquivo de configuração main.cf

        - postconf -n - exibe as definições de parâmetros que não foram deixados nos seus valores padrão, por estarem explicitamente especificados no main.cf
        - postconf -v - habilita o modo verbose
    - postfix(1) - program de controle do Postfix
        - Exemplos:
            - postfix <comando> - envia um comando para o daemon Postfix
                - check - avisa sobre proprietários ou permissões erradas e cria diretórios em falta
                - start - inicia o sistema de correio Postfix
                - stop - para o sistema de correio Postfix em uma ordem adequada
                - abort - para o sistema de correio Postfix de forma abrupta
                - flush - força a re-entrega de mensagens
                - reload - recarreca os arquivos de configuração
                - status - indica se o sistema de correio Postfix está em execução
                - set-permissions [nome=valor] - define proprietário e permissões aos arquivos relacionados ao Postfix
                - upgrade-configuration [nome=valor] - atualiza os arquivos main.cf e master.cf com informação que o Postfix precisa para executar

Conhecimento básico do protocolo SMTP

- SMTP [2]
    - Simple Mail Transfer Protocol (SMTP) é um padrão da Internet para transmissão de correio eletrônico (e-mail). Primeiro definido pela RFC 821, em 1982, foi atualizado em 2008 com as adições SMTP Estendido pela RFC 5321 - que é o protocolo em uso difundido hoje.
    - O SMTP por padrão usa a porta TCP 25. O protocolo para o envio de correio é o mesmo, mas usa a porta 587. Conexões SMTP protegidas por SSL, conhecido como SMTPS, usa a porta padrão 465 (fora do padrão, mas às vezes utilizados por razões de legado).
    - Embora os servidores de correio eletrônico e outros agentes de transferência de correio usam o SMTP para enviar e receber mensagens de email, aplicações de correio de cliente de nível de usuário normalmente usam SMTP somente para envio de mensagens para um servidor de correio para encaminhamento. Para receber mensagens, os aplicativos clientes geralmente usam POP3 ou IMAP.
    - Embora sistemas proprietários (como o Microsoft Exchange e Lotus Notes/Domino) e sistemas de webmail (como Hotmail, Gmail e Yahoo! Mail) usam seus próprios protocolos não-padrão para acessar as contas de caixa de correio em seus próprios servidores de correio, todos usam SMTP quando enviam ou recebem e-mails de fora seus próprios sistemas.
    - Modelo de processamento de correio
        - O E-mail é enviado por um cliente de email (MUA, agente de usuário de correio) para um servidor de correio (MSA, agente de submissão de correio) usando SMTP na porta 587 TCP. A maioria dos provedores de caixas de correio ainda permite a submissão tradicional na porta 25. De lá, o MSA entrega o correio para seu agente de transferência de correio (MTA). Muitas vezes, esses dois agentes são apenas diferentes instâncias do mesmo software executados com opções diferentes na

mesma máquina. O processamento local pode ser feito em uma única máquina, ou dividido entre vários aparelhos (appliances); no primeiro caso, os processos envolvidos podem compartilhar arquivos; no último caso, o SMTP é utilizado para transferir a mensagem internamente, com cada host configurado para usar o aparelho seguinte como um smart host. Cada processo é um MTA em seu próprio direito; ou seja, um servidor SMTP.

- O MTA de fronteira tem de localizar o host de destino. Ele usa o sistema de nome de domínio (DNS) para procurar o registro de intercâmbio de correio (MX record) para o domínio do destinatário (a parte do endereço de e-mail à direita do @). O registro MX retornado contém o nome do host de destino. O MTA próximo se conecta ao servidor de troca como um cliente SMTP.
- Uma vez que o alvo MX aceita a mensagem recebida, ele repassa para um agente de entrega de mail (MDA) para entrega de correio local. Um MDA é capaz de salvar as mensagens no formato de caixa de correio relevante. Novamente, a recepção de correio pode ser feita usando vários computadores ou apenas um. Um MDA pode entregar mensagens diretamente para o armazenamento, ou encaminhá-las através de uma rede usando SMTP, ou quaisquer outros meios, incluindo o Local Mail Transfer Protocol (LMTP), um derivado de SMTP projetado para essa finalidade.
- Uma vez entregue ao servidor de correio local, o correio é armazenado para recuperação em lote por clientes de correio autenticados (MUAs). O correio é recuperado por aplicativos de usuário final, chamados clientes de email, usando o Internet Message Access Protocol (IMAP), um protocolo que tanto facilita o acesso a e-mail e gerencia e-mails armazenados, ou o Post Office Protocol (POP), que normalmente usa o formato de arquivo de correio tradicional mbox ou um sistema proprietário, como o Microsoft Exchange/Outlook ou Lotus Notes/Domino. Clientes Webmail podem usar qualquer um dos métodos, mas o protocolo de recuperação não é muitas vezes um padrão formal.
- O SMTP define transporte de mensagens, e não o conteúdo da mensagem. Assim, define o envelope do correio e os seus parâmetros, tais como o remetente do envelope, mas não o cabeçalho (exceto informações de rastreio), nem o corpo da mensagem propriamente dita. STD 10 e RFC 5321 definem o SMTP (o envelope), enquanto STD 11 e RFC 5322 definem a mensagem (cabeçalho e corpo), formalmente conhecido como o Internet Message Format.

Consciência do sendmail e exim

- Sendmail [3]
  - Sendmail é uma facilidade de roteamento de correio eletrônico, entre redes de finalidade geral, que suporta muitos tipos de métodos de transferência e de entrega, incluindo o Simple Mail Transfer Protocol (SMTP), utilizado para o transporte de e-mail através da Internet.
  - Um descendente do programa delivermail, escrito por Eric Allman, o Sendmail é um projeto bem conhecido do software livre e open-source, e comunidades Unix. Ele se espalhou tanto como software livre e software proprietário.
  - Visão geral
    - Allman tinha escrito o original ARPANET delivermail, que foi embarcado em 1979 com o BSD 4.0 e 4.1. Ele escreveu o Sendmail como um derivado do delivermail no início de 1980 na Universidade de Berkeley. Foi embarcado com o BSD 4.1c em 1983, a primeira versão BSD, que incluiu protocolos TCP/IP.

        - Em 1996, aproximadamente 80% dos servidores de correio alcançáveis publicamente na Internet executavam o Sendmail. Pesquisas mais recentes sugerem um declínio, com 8,86% dos servidores de correio em Abril 2014 detectados como executando Sendmail, em um estudo realizado pela E-Soft, Inc.
        - Allman projetou o Sendmail para incorporar uma grande flexibilidade, mas pode ser assustador para configurar, para os novatos. Pacotes de configuração padrão fornecidos com a distribuição do código fonte requerem o uso da linguagem de macros M4 que esconde muito da complexidade de configuração. A configuração define as opções de entrega de correio de sites locais e seus parâmetros de acesso, o mecanismo de redirecionamento para sites remotos, assim como muitos parâmetros de ajuste de aplicação.
        - O Sendmail suporta uma variedade de protocolos de transferência de correio, incluindo SMTP, ESMTP, Mail-11 do DECnet, HylaFax, Quickpage e UUCP. Além disso, o Sendmail V8.12 a partir de Setembro de 2001, introduziu suporte para milters - programas de filtragem de email externos que podem participar em cada etapa da conversa SMTP.
- Exim [4]
    - Exim é um agente de transferência de correio (MTA) usado em sistemas operacionais Unix-like. O Exim é um software livre distribuído sob os termos da GNU General Public License, e tem como objetivo ser um mailer geral e flexível com amplas facilidades para controle de entrada de e-mail.
    - O Exim foi portado para a maioria dos sistemas Unix-like, bem como para o Microsoft Windows utilizando a camada de emulação Cygwin. O Exim 4 é atualmente o MTA padrão em sistemas Debian GNU/Linux.
    - Um grande número de instalações Exim existem, especialmente junto dos prestadores de serviços de Internet e universidades no Reino Unido. O Exim também é amplamente utilizado com o gerente de lista de discussão GNU Mailman, e cPanel.
    - Em Abril de 2014, em um estudo realizado pela e-Soft, Inc., aproximadamente 51% dos servidores de correio alcançáveis publicamente na Internet executam Exim.
    - Origem
        - A primeira versão do Exim foi escrita em 1995, por Philip Hazel, para uso em sistemas de e-mail da Universidade de Cambridge Serviços de Computação. O nome inicialmente representava EXperimental Internet Mailer. Ele foi originalmente baseado em um MTA mais velho, Smail-3, mas desde então divergiu do Smail-3 na sua concepção e filosofia.
    - Modelo de projeto
        - O Exim, como Smail, ainda segue o modelo de design Sendmail, onde um único binário controla todas as facilidades do MTA. Ele tem fases bem definidas durante as quais ele ganha ou perde privilégios.
        - O registro de segurança do Exim tem sido bastante limpo, com apenas um punhado de problemas de segurança graves diagnosticados ao longo dos anos. Uma vez que a redesenhada versão 4 foi lançada, houve quatro falhas de execução remota de código e uma falha conceitual a respeito de quanta confiança é adequada para colocar no usuário de tempo de execução; a última foi corrigida em um bloqueio de segurança na revisão 4,73, uma das raras ocasiões quando Exim quebrou a compatibilidade com configurações de trabalho. Essa questão não teria sido impedida por meio de um projeto não monolítico.

Termos e utilitários

- arquivos de configuração e comandos para postfix
  - postalias - (1) manutenção de banco de dados de apelidos do Postfix
  - postconf - (1) utilitário de configuração do Postfix
  - postfix - (1) program de controle do Postfix
  - sendmail.postfix - camada de emulação de comandos sendmail
- /etc/postfix/ - diretório de configuração do postfix
- /var/spool/postfix/ - diretório de dados do postfix
- camada de emulação de comandos sendmail
  - Para manter a compatibilidade, sistemas de mensageria que usam o sendmail e SMTP para entrega de email, implementam comandos originalmente criados pelo sendmail. Comandos como mailq, newaliases e o próprio sendmail são comumente implementados por outros MTAs. O Exim por exemplo, substitui o binário do sendmail para manter essa compatibilidade.
- /etc/aliases - (5) arquivo de aliases para sendmail
- logs relacionados a email no /var/log/
  - /var/log/maillog - Postfix / Sendmail
  - /var/log/exim/
    - main.log - arquivo de log principal
    - reject.log - mensagens rejeitadas
    - panic.log - erros graves no serviço

Referências

1. http://en.wikipedia.org/wiki/Postfix_(software)
2. http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol
3. http://en.wikipedia.org/wiki/Sendmail
4. http://en.wikipedia.org/wiki/Exim

Simulado

1. O Postfix implementa uma primeira camada de contra spambots e malware e através de softwares externos, ele pode aumentar o nível de proteção do serviço.
   a. V
   b. F

2. O Postfix implementa um motor paralelizado de entrega de mensagens.
   a. V
   b. F

3. São protocolos suportados pelo Postfix:
   a. SMTP
   b. LMTP
   c. IMAP

d. POP3

4. O Postfix suporta domínios virtuais.
   a. V
   b. F

5. O Postfix suporta pesquisas nos seguintes mecanismos externos:
   a. LDAP
   b. MySQL
   c. Berkley DB
   d. PostgreSQL

6. … é o arquivo principal de configuração do Postfix.

7. O Postfix pode usar o arquivo padrão de apelidos do Sendmail.
   a. V
   b. F

8. As diretivas … e … são responsáveis pode definir um limite padrão de cota para usuários de domínio direto e domínios virtuais, respectivamente.

9. … é o arquivo padrão de definição de domínios e endereços virtuais.

10. As informações de cota e endereços virtuais podem ser obtidas através de pesquisa a bancos de dados.
    a. V
    b. F

11. São elementos do serviço de correio eletrônico:
    a. MTA
    b. MDA
    c. MSA
    d. MUA

12. A comunicação entre servidores MTA se da através do protocolo ….

13. A sequência adequada de uso do serviço de email é MUA -> MSA -> MTAs -> MDA -> MUA.
    a. V
    b. F

14. O protocolo SMTP utiliza de registros … no sistema de nomes de domínios, para alcançar os endereços de destino das correspondências.

15. O protocolo SMTP define o transporte e conteúdo de mensagens.
    a. V
    b. F

16. O Sendmail foi o software mais popular de uso em servidores de correio eletrônico, mas devido principalmente a sua complexidade de configuração, foi sendo substituído por outros softwares.
    a. V
    b. F

17. O Sendmail foi o primeiro software de correio eletrônico baseado em TCP/IP.
    a. V
    b. F

18. O Exim usa uma arquitetura monolítica.
    a. V
    b. F

19. O Exim é um software de correio eletrônico que implementa as funcionalidades MSA, MTA e MDA.
    a. V
    b. F

20. Devido a arquitetura monolítica do Exim, muitas vulnerabilidades foram detectadas no software.
    a. V
    b. F

211.2 Gerenciando entrega local de e-mail

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de implementar software cliente de gerenciamento de email, para filtrar, ordenar e monitorar entrada de email de usuários.
Áreas de conhecimentos chave:

- Arquivos de configuração, ferramentas e utilitários procmail
- Uso do procmail em ambos lados do servidor e do cliente

Termos e utilitários:

- ~/.procmailrc
- /etc/procmailrc
- procmail
- Formatos mbox e Maildir

Áreas de conhecimentos chave

Arquivos de configuração, ferramentas e utilitários procmail

- procmail [1]
  - Em sistemas de e-mail, procmail, um agente de entrega de mail (MDA), pode classificar e-mails recebidos em vários diretórios e filtrar mensagens de spam. O Procmail é estável, mas não mais mantido. Os usuários que desejam usar um programa de manutenção, são aconselhados a usar um MDA alternativo, como o maildrop.
  - Invocação
    - O agente de entrega de correio procmail, geralmente não é iniciado a partir da linha de comando, mas é invocado por subsistemas de entrega de correio, como um agente de transporte de correio (como Sendmail ou Postfix), ou a partir de um agente de recuperação de email (como o fetchmail). Isso faz com que o processamento de correio seja orientado a eventos. A ferramenta acompanhante formail permite o procmail ser usado em processamento em lote no correio que já está na caixa de entrada do usuário.
  - Receitas
    - O agente procmail usa receitas, para determinar onde entregar as várias mensagens.
    - Cada receita que o procmail usa consiste de:
      - modo
      - condições
      - ação
    - As receitas usadas pelo procmail podem ser condicionais ou não condicionais. Se as condições são deixadas de fora, a receita é não condicional.
    - O procmail tem dois tipos de receitas:
      - Receitas de entrega
      - Receitas de não entrega
    - As receitas são lidas de cima para baixo. A primeira receita de entrega termina o processo de entrega (a não ser que a flag de modo especifique o contrário)
  - Condições

        - As condições são geralmente expressões regulares estendidas, apesar de existir outras formas de condição também.
    - Operação básica
        - A ferramenta procmail lê as mensagens de correio dadas a partir da entrada padrão.
        - A ferramenta procmail irá processar as receitas antes de distribuir as mensagens de correio nas apropriadas caixas de correio.
    - Outras operações
        - Outras operações comuns realizadas com o procmail incluem filtragem e ordenação de emails em diferentes pastas de acordo com palavras no campos de, para, assunto, texto do correio, ou enviando auto-respostas, mas operações mais sofisticadas também são possíveis.
        - Filtragem de Spam
            - Uma prática comum é deixar o procmail chamar um program externo de filtragem de Spam, tal como SpamAssassin. Esse método pode aceitar que Spam seja filtrado ou mesmo deletado.
        - Gerenciando listas de email
            - Os desenvolvedores do procmail construiram um gerenciador de lista de correio eletrônico chamado SmartList, sob o procmail.
- procmailrc [2]
    - O procmail deve ser invocado automaticamente sobre o mecanismo arquivo .forward, assim que chegar correio. Em alternativa, quando instalado por um administrador de sistema (e na configuração padrão do Red Hat Linux), ele pode ser invocado de dentro do mailer imediatamente. Quando invocado, ele primeiro define algumas variáveis de ambiente para os valores padrão, lê a mensagem de correio a partir de stdin até um EOF, separa o corpo do cabeçalho e, em seguida, se argumentos de linha de comando não estão presentes, ele começa a procurar um arquivo chamado $HOME/.procmailrc. De acordo com as receitas de processamento desse arquivo, a mensagem de correio eletrônico que acabou de chegar é distribuída para a pasta devida (e mais). Se nenhum rcfile for encontrado, ou o processamento do rcfile chega ao fim, o procmail irá armazenar o e-mail na caixa de correio padrão do sistema.
    - Se nenhum rcfile e a opção -p não foi especificada na linha de comando, o procmail irá, antes da leitura do $HOME/.procmailrc, interpretar comandos a partir de /etc/procmailrc (se houver). Cuidados devem ser tomados ao criar o /etc/procmailrc, porque, se as circunstâncias permitirem, ele será executado com privilégios de root (ao contrário do arquivo $HOME/.procmailrc é claro).
    - Se estiver executando suid root ou com privilégios de root, o procmail será capaz de executar, como uma funcionalidade melhorada, agente de entrega de correio compatíveis com versões anteriores.
    - O procmail também pode ser usado como um filtro de correio de propósito geral, ou seja, foram tomadas disposições para permitir o procmail ser chamado em uma regra sendmail especial.
    - O formato rcfile é descrito em detalhe na página de manual procmailrc(5).
    - A técnica de pontuação ponderada é descrita, em detalhe, na página de manual procmailsc(5).
    - Exemplos de receitas rcfile podem ser consultadas na página de manual procmailex(5).
- Arquivos de configuração
    - ~/.procmailrc(procmailrc(5)) - arquivo rc do procmail, por usuário
    - /etc/procmailrc(procmailrc(5)) - arquivo rc global do procmail

- Ferramentas e utilitários
    - procmail(1) - processador de email autônomo
        - Exemplos:
            - procmail - lê da entrada padrão e processa com as receitas encontradas
            - procmail -p - preserva o ambiente antigo
            - procmail -t - faz o procmail falhar suavemente
            - procmail -f <origem> - faz o procmail regerar o cabeçalho “From”
            - procmail -o - ao invés de aceitar qualquer um gerar o “From”, simplesmente sobrescreve o falso
            - procmail -a <argumento> - define o argumento $N com valor especificado
            - procmail -d <destinatário> - habilita o modo de entrega explícita
            - procmail -m - torna o procmail um filtro de email de propósito geral
    - formail
        - Exemplos:
            - formail - formata o email vindo da entrada no formato de caixa de correio
            - formail -b - não escapa quaisquer cabeçalhos falsos
            - formail -p <prefixo> - define um prefixo de citação diferente
            - formail -c - concatena continuamente campos no cabeçalho
            - formail -z - garante que exista espaço em branco entre o nome do campo e o conteúdo
            - formail -r - gera um cabeçalho de auto-resposta
            - formail -s - a entrada será divida em mensagens de correio separadas
            - formail -n [máximo de processos] - faz o formail trabalhar de forma paralela
            - formail -e - não requer linhas em branco para processar o cabeçalho do uma nova mensagem
            - formail -q - diz ao formail para ser quieto em relação a escrever erros
            - formail +<salto> - salta as primeiras N mensagens enquanto está separando

Uso do procmail em ambos lados do servidor e do cliente

- Servidor [3]
    - /etc/aliases
        - <endereço>: “| /usr/bin/procmail <procmailrc>”
- Cliente [3]
    - ~/.forward
        - "|IFS=' ' && exec /usr/local/bin/procmail || exit 75 <usuário>"

Termos e utilitários

- ~/.procmailrc - procmailrc(5) script procmail pessoal de usuário
- /etc/procmailrc - procmailrc(5) script procmail geral para usuários
- procmail - (1) processador de email autônomo
- formatos mbox e Maildir
    - mbox - [4] formato de arquivo usado para armazenar coleções de mensagens de correio eletrônico. Todas as mensagens em um mbox são concatenadas e armazenadas como um único arquivo de texto plano.
    - Maildir - [5] formato de armazenamento de mensagens de correio eletrônico, onde cada mensagem é mantida em um arquivo separado de nome único e cada pasta é um diretório.

Referências

1. http://en.wikipedia.org/wiki/Procmail
2. procmailrc(5)
3. http://tldp.org/LDP/LG/issue14/procmail.html
4. http://en.wikipedia.org/wiki/Mbox
5. http://en.wikipedia.org/wiki/Maildir

Simulado

1. O procmail é orientado a receitas que consistem em três itens: …, … e ….

2. As receitas do procmail podem trabalhar no modo de entrega ou não entrega de mensagens.
   a. V
   b. F

3. Através de programas externos, é possível o procmail realizar filtragem de Spam.
   a. V
   b. F

4. Para armazenar na pasta EXTERNAL, todos os emails que não têm origem em domínios example.com, usando arquivo de lock, uma regra válida seria:
   a. :0
   b. :0:
   c. :0:
   d. :0

5. Para armazenar na pasta Example.com, todos os emails que tem origem no domínio example.com, usando arquivo de lock e, enviar uma cópia para user1example@gmail.com, uma regra válida seria:
   a. :0
   b. :0:
   c. :0:
   d. :0

6. Ao usar o arquivo /etc/procmailrc, deve-se ter o cuidado de verificar se a opção de execução como root está habilitada, o que pode gerar problemas de segurança.
   a. V
   b. F

7. O procmail pode ser invocado através do formail para verificar as mensagens que já estão armazenadas na caixa do usuário.
   a. V
   b. F

8. O procmail pode ser configurado do lado do servidor (mta), através do arquivo ….

9. O arquivo … é usado para configurar o procmail do lado do cliente.

10. Tanto do lado do servidor quanto do cliente, o procmail é invocado através de canalização do email para o processo procmail.
    a. V
    b. F

211.3 Gerenciando entrega remota de e-mail

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de instalar e configurar daemons POP e IMAP.
Áreas de conhecimentos chave:

- Configuração do Courier IMAP e do Courier POP
- Configuração do Dovecot

Termos e utilitários:

- /etc/courier/
- Configuração do Dovecot

Áreas de conhecimentos chave

Configuração do Courier IMAP e do Courier POP

- Courier [1]
  - O servidor de correio Courier é um servidor de agente de transferência de correio (MTA) que fornece ESMTP, IMAP, POP3, SMAP, webmail, e serviços de listas de discussão com os componentes individuais. Ele é mais conhecido por seu componente de servidor IMAP.
  - O Courier pode funcionar como um encaminhador de email intermediário, entre uma rede interna e a Internet, ou realizar a entrega final para caixas de correio. Ele utiliza maildirs como seu formato de armazenamento nativo e também pode entregar correio para arquivos de caixa postal (mailbox) legados. Os arquivos de configuração estão em formato de texto simples e pode incluir scripts Perl.
  - O Courier pode prestar serviços de correio para as contas regulares do sistema operacional. Ele também pode fornecer serviços de correio para contas de correio virtuais, gerenciados por qualquer um serviço de diretório LDAP ou banco de dados de autenticação Berkeley DB, MySQL ou PostgreSQL.
  - Partes de Courier, tais como o sistema de filtragem maildrop, o servidor de webmail e IMAP, também podem ser instalados como pacotes independentes, que podem ser usados com outros servidores de correio. O Courier-IMAP é uma combinação particularmente popular entre os servidores Qmail, Exim e Postfix que são configurados para usar maildirs.
  - O fonte do Courier compila na maioria dos sistemas operacionais baseados em POSIX baseados em kernels Linux e derivados do BSD. Ele usa extensões SMTP para a gestão de lista e filtragem de spam.
  - Uma revisão do SourceForge em 24 de Fevereiro de 2009 indica que o Courier tinha sido baixado do site mais de 1.000.000 de vezes no ranking 282 em todos os pacotes, com inúmeras aplicações de terceiros.
- Configuração
  - /etc/courier/imapd
  - /etc/courier/imapd-ssl
  - /etc/courier/pop3d

Configuração do Dovecot

- Dovecot [3]

    - Dovecot é um servidor IMAP e POP3 open-source para sistemas UNIX-like/Linux, escrito principalmente com a segurança em mente. Timo Sirainen originou o Dovecot e lançou primeiro em Julho de 2002. Desenvolvedores do Dovecot primariamente objetivam produzir um servidor de correio eletrônico open-source, leve, rápido e de fácil configuração.
    - De acordo com o Openemailsurvey, o Dovecot tem uma base instalada de mais de 2,9 milhões de servidores de e-mail em todo o mundo e uma quota de 57% de todos os servidores IMAP no mercado global. Enquanto o software Dovecot pode ser usado em uso comercial sem quaisquer taxas de licenciamento, uma versão comercial também está disponível como Dovecot Pro. A versão comercial é fornecida pela Dovecot Oy juntamente com suporte e add-ons empresariais como o plugin de armazenamento de objetos. Desde Março de 2015, Dovecot Oy tem sido parte da família Open-Xchange.
    - Características
        - O Dovecot pode trabalhar com formatos padrão mbox, Maildir, e seu próprio nativo dbox de alto desempenho. É totalmente compatível com implementações de servidores UW IMAP e Courier IMAP, bem como clientes de email que acessam as caixas de correio diretamente.
        - Ele também inclui um agente de entrega correio (chamado de agente de entrega local na documentação do Dovecot), com o suporte de filtragem Sieve opcional.
        - O Dovecot suporta uma variedade de esquemas de autenticação para acesso IMAP e POP incluindo CRAM-MD5 e o mais seguro DIGEST-MD5.
        - Com a versão 2.2 alguns novos recursos foram adicionados ao Dovecot, por exemplo, extensões de comando IMAP adicionais, o dsync foi reescrito ou otimizado, e caixas de correio compartilhadas agora suportam flags por usuários.
        - A Apple Inc. inclui o Dovecot para os serviços de e-mail desde o Mac OS X Server 10.6 Snow Leopard.
- Configuração
    - /etc/dovecot/
        - conf.d
        - dovecot.conf(5) - o arquivo de configuração para o servidor imap e pop3 dovecot

Termos e utilitários

- /etc/courier/ - diretório de configuração dos componentes courier
- dovecot.conf - (5) o arquivo de configuração para o servidor imap e pop3 dovecot

Referências

1. http://en.wikipedia.org/wiki/Courier_Mail_Server
2. http://www.courier-mta.org/install.html
3. http://en.wikipedia.org/wiki/Dovecot_(software)
4. http://wiki2.dovecot.org/FrontPage

Simulado

1. O Courier é um software MSA/MTA/MDA.
    a. V
    b. F

2. O Courier pode ter seus componentes instalados individualmente.
    a. V
    b. F

3. O software Courier permite autenticação em serviços de diretórios e alguns bancos de dados como MySQL e PostgreSQL.
    a. V
    b. F

4. … e … são os arquivos de configuração do serviço imap e imaps do software Courier.

5. … é o arquivo de configuração para os serviços pop3 e pop3s do software Courier.

6. O Dovecot é um software MSA/MTA/MDA.
    a. V
    b. F

7. O Dovecot permite os formatos de armazenamento de mensagem mbox e dbox.
    a. V
    b. F

8. … é o arquivo de configuração do software Dovecot.

9. O Dovecot suporta o esquema de autenticação DIGEST-MD5.
    a. V
    b. F

10. O Dovecot possui suporte ao protocolo SMTP.
    a. V
    b. F

Tópico 212: Segurança do Sistema

212.1 Configurando um roteador

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de configurar o sistema para executar a tradução de endereço de rede (NAT, IP masquerading) e dizer sua significância em protegendo uma rede. Esse objetivo inclui configurando redirecionamento de porta, gerenciando regras de filtro e evitando ataques.
Áreas de conhecimentos chave:

- Arquivos de configuração, ferramentas e utilitários iptables
- Ferramentas, comandos e utilitários para gerenciar tabelas de roteamento
- Faixas de endereços privados
- Redirecionamento de portas e encaminhamento de IP
- Listar e escrever filtragem e regras que aceitam ou bloqueiam datagramas baseados no protocolo de origem ou destino, porta e endereço
- Protocolo, porta e endereço de destino
- Salvar e recarregar configurações de filtragem
- Consciência do ip6tables e filtragem

Termos e utilitários:

- /proc/sys/net/ipv4/
- /etc/services
- iptables

Áreas de conhecimentos chave

Arquivos de configuração, ferramentas e utilitários iptables

- IPTables [1]
    - iptables é um programa aplicativo de espaço do usuário que permite que um administrador de sistema configure as tabelas fornecidas pelo firewall do kernel Linux (implementado como diferentes módulos Netfilter) e as cadeias e regras que armazena. Diferentes módulos do kernel e programas são atualmente utilizados para diferentes protocolos; iptables se aplica ao IPv4, IP6Tables para o IPv6, arptables ao ARP e ebtables para frames Ethernet.
    - O iptables requer privilégios elevados para operar e deve ser executado pelo usuário root. Caso contrário, ele não funciona. Na maioria dos sistemas Linux, o iptables é instalado como /usr/sbin/iptables e documentado em suas páginas man, que podem ser abertas com o man iptables, quando instalado. Ele também pode ser encontrado em /sbin/iptables, mas desde que o iptables é mais como um serviço em vez de um "binário essencial", o local preferido continua a ser /usr/sbin.
    - O termo iptables também é comumente usado para se referir inclusive para os componentes no nível do kernel. x_tables é o nome do módulo do kernel levando a porção de código compartilhado usado por todos os quatro módulos, que também fornece a API utilizada para extensões; posteriormente, Xtables é mais ou menos usado para se referir a toda a arquitetura do firewall (v4, v6, arp, e eb).

- O sucessor do iptables é nftables, que foi incorporado na linha principal do kernel Linux no kernel versão 3.13, que foi lançado em 19 de Janeiro de 2014.
- Visão geral
  - O Xtables permite que o administrador do sistema defina as tabelas que contêm cadeias de regras para o tratamento de pacotes. Cada tabela está associada com um tipo diferente de processamento de pacotes. Os pacotes são processados sequencialmente atravessando as regras em cadeias. Uma regra em uma cadeia pode causar um “go to” ou saltar para outra cadeia, e isso pode ser repetido para qualquer nível de aninhamento desejado. Um salto é como uma "chamada", ou seja, o ponto de onde se saltou é lembrado. Cada pacote de rede que entra ou sai do computador, atravessa pelo menos uma cadeia.
  - A origem do pacote determina a cadeia que esse atravessa inicialmente. Há cinco cadeias predefinidas (mapeamento aos cinco ganchos Netfilter disponíveis), apesar que uma tabela pode não ter todas as cadeias. Cadeias predefinidas têm uma política, por exemplo DROP, que é aplicada ao pacote se atingir o final da cadeia. O administrador do sistema pode criar tantas outras cadeias como desejado. Essas cadeias têm nenhuma política; se um pacote atinge o fim da cadeia, ele é devolvido para a cadeia que a chamou. Uma cadeia pode estar vazia.
    - PREROUTING: Pacotes vão entrar nessa cadeia antes de ser tomada uma decisão de roteamento.
    - INPUT: Pacote que vai ser entregue localmente. Ele não tem nada a ver com os processos que têm um soquete aberto; entrega local é controlada pela tabela de roteamento "de entrega local" (“local-delivery”): ip route show table local.
    - FORWARD: Todos os pacotes que foram roteados e não eram para entrega local, irão percorrer essa cadeia.
    - OUTPUT: Os pacotes enviados a partir da própria máquina vão visitar esta cadeia.
    - POSTROUTING: Foi feita a decisão de roteamento. Pacotes entram nessa cadeia antes de entregá-los ao hardware.
  - Cada regra em uma cadeia contém a especificação de quais pacotes ela corresponde. Pode também conter um alvo (usado para extensões) ou veredito (uma das decisões embarcadas). Como um pacote percorre uma cadeia, cada regra, por sua vez é examinada. Se uma regra não coincide com o pacote, o pacote é passado para a próxima regra. Se uma regra coincide com o pacote, a regra leva a ação indicada pelo alvo/veredito, o que pode resultar em que o pacote possa ser autorizado a prosseguir ao longo da cadeia ou possa não ser. Correspondências compõem a grande parte dos conjuntos de regras, como elas contêm as condições que os pacotes são testados para. Isso pode acontecer em qualquer camada do modelo OSI, tais como, por exemplo, com os parâmetros “--mac-source” e “-p tcp --dport”, e há também correspondências independentes de protocolos, tais como “-m time”.
  - O pacote continua a percorrer a cadeia até que
    - uma regra corresponde ao pacote e decide o destino final do pacote, por exemplo, chamando um dos ACCEPT ou DROP, ou um módulo retornando um destino final; ou
    - uma regra chama o veredito RETURN, que no processamento do caso retorna para a cadeia de chamada; ou

                - o final da cadeia é atingido; travessia tanto pode continuar na cadeia principal (como se o RETURN foi usado), ou a política de cadeia de base, que é um destino final, é utilizada.
            - Alvos também retornam um veredito como ACCEPT (módulos NAT vão fazer isso) ou DROP (por exemplo, o módulo REJECT), mas também podem implicar CONTINUE (por exemplo, o módulo LOG; CONTINUE é um nome interno) para continuar com a próxima regra como se nenhum alvo/veredito fosse especificado.
- Arquivos de configuração
    - RH e derivados
        - /etc/sysconfig/iptables
        - /etc/sysconfig/iptables-config
    - Debian e derivados
        - /etc/iptables/rules.v4
- Ferramentas e utilitários
    - iptables(8) - ferramenta de administração para filtragem de pacote IPv4 e NAT
    - iptables-save(8) - despeja regras do iptables para a saída padrão
    - iptables-restore(8) - restaura IP Tables
    - iptables-xml(8) - converte o formato iptables-save para XML

Ferramentas, comandos e utilitários para gerenciar tabelas de roteamento

- route(8) - exibe / manipula a tabela de roteamento IP
- ip(8) - exibe / manipula roteamento, dispositivos, política de roteamento e tuneis
- iptables
    - Filtrar pacotes do processo de roteamento

Faixas de endereços privados

- Rede privada [2]
    - Na arquitetura de endereçamento da Internet, uma rede privada é uma rede que utiliza o espaço de endereço IP privado, seguindo as normas estabelecidas pela RFC 1918 para o Internet Protocol versão 4 (IPv4), e RFC 4193 para o Internet Protocol versão 6 (IPv6). Esses endereços são comumente usados para casa, escritório, e de redes locais de empresa (LANs), quando os endereços globalmente roteáveis não são obrigatórios, ou não estão disponíveis, para as aplicações de rede pretendidas. Sob IPv4, os espaços de endereços IP privados foram originalmente definidos em um esforço para atrasar esgotamento dos endereços IPv4, mas eles também são um recurso do IPv6, a próxima geração Internet Protocol.
    - Esses endereços são caracterizados como privado, porque não são globalmente delegados, o que significa que eles não são atribuídos a nenhuma organização específica, e os pacotes IP abordado com eles, não podem ser transmitidos através da Internet pública. Qualquer pessoa pode usar esses endereços sem a aprovação de um Registro Regional da Internet (RIR). Se uma rede privada tal precisa se conectar à Internet, é preciso usar um gateway conversor de endereços de rede (NAT), ou um servidor proxy.
    - Faixa de endereços IPv4
        - 10.0.0.0 - 10.255.255.255
        - 172.16.0.0 - 172.31.255.255
        - 192.168.0.0 - 192.168.255.255
    - Faixa de endereços IPv6
        - fd00::/8

- Endereços de link local
  - Outro tipo de rede privada usa o intervalo de endereços da ligação local (link-local). A validade de endereços da ligação local é limitada a uma única ligação; por exemplo, para todos os computadores conectados a um interruptor, ou para uma rede sem fio. Hosts em lados diferentes de uma ponte (bridge) também estão no mesmo link, enquanto os hosts em diferentes lados de um roteador estão em links diferentes.
  - IPv4
    - Em IPv4, endereços link-local são codificados nas RFCs 6890 e 3927. Sua utilidade está na configuração automática própria por meio de dispositivos de rede, quando os serviços de DHCP não estão disponíveis e a configuração manual por um administrador de rede não é desejável.
    - O bloco 169.254.0.0/16 está reservada para esse propósito, com a exceção da primeira e última faixa /24 de subredes. Se um host em uma rede IEEE 802 (ethernet) não pode obter um endereço de rede via DHCP, um endereço de 169.254.1.0 a 169.254.254.255 pode ser atribuído pseudo-aleatório. A norma prescreve que as colisões de endereço devem ser tratadas graciosamente.
  - IPv6
    - Em IPv6, os endereços link-local são codificadas na RFC 4862. A sua utilização é obrigatória, e é uma parte integral da norma IPv6.
    - A arquitetura do endereçamento IPv6 (RFC 4291) deixa de lado o bloco fe80::/10 para o endereço IP de configuração automática.

Redirecionamento de portas e encaminhamento de IP

- Redirecionamento de portas [3]
  - Em redes de computadores, o encaminhamento de porta ou mapeamento de porta é um aplicativo de tradução de endereços de rede (NAT), que redireciona o pedido de comunicação de combinação de um endereço e número de porta para outro, enquanto os pacotes estão atravessando um gateway de rede, como um roteador ou firewall. Essa técnica é mais comumente usada para fazer serviços em um host que residem em uma rede protegida ou mascarada (interna), disponíveis para hosts no lado oposto da porta de entrada (rede externa), por remapeamento do endereço IP de destino e o número da porta de comunicação para um host interno.
  - Exemplos:
    - iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 192.168.0.1:80 - adiciona uma regra de redirecionamento de porta onde todo o tráfego que entrar na interface eth0 e tiver como destino a porta tcp 80, será redirecionado para o endereço 192.168.0.1 na porta 80
    - iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to-destination 192.168.0.1:3128 - adiciona uma regra de redirecionamento de porta onde todo o tráfego que entrar na interface eth1 e tiver como destino a porta tcp 80, será redirecionado para o endereço 192.168.0.1 na porta 3128
- Encaminhamento de IP [4]
  - Encaminhamento IP, também conhecido como encaminhamento Internet, é um processo utilizado para determinar o caminho que um pacote ou datagrama pode ser enviado. O processo utiliza informações de roteamento para tomar decisões e é projetado para enviar um pacote ao longo de várias redes.

    - Exemplos:
        - iptables -t filter -A FORWARD -j ACCEPT - adiciona uma regra que aceita o encaminhamento de pacotes
        - iptables -t filter -A FORWARD -j REJECT --reject-with icmp-host-prohibited - adiciona uma regra que rejeita o encaminhamento de pacotes com o erro “icmp-host-prohibited”
- IP Masquerade - NAT N para 1 [5]
    - A maioria dos NATs mapeia vários hosts privados para um endereço IP exposto publicamente. Em uma configuração típica, uma rede local utiliza uma das designadas subredes de endereço IP "privados" (RFC 1918). Um roteador na rede tem um endereço privado no espaço de endereço. O roteador também está conectado à Internet com um endereço "público" atribuído por um provedor de serviços de Internet. Como o tráfego passa a partir da rede local para a Internet, o endereço de origem em cada pacote é traduzido em tempo real a partir de um endereço privado para o endereço público. O roteador monitora dados básicos sobre cada conexão ativa (em particular o endereço de destino e porta). Quando uma resposta retorna para o roteador, ele usa os dados de rastreamento da conexão, armazenados durante a fase de saída, para determinar o endereço privado ao qual encaminhar a resposta.
    - Todos os pacotes de datagrama sobre redes IP tem um endereço IP de origem e um endereço IP de destino. Normalmente pacotes passando a partir da rede privada para a rede pública, terão seu endereços de origem modificados, enquanto os pacotes de passagem da rede pública de volta para a rede privada, terão seus endereços de destino modificados. Mais configurações complexas também são possíveis.
    - Para evitar ambigüidade em como traduzir pacotes devolvidos, são necessárias novas modificações para os pacotes. A grande maioria do tráfego da Internet é de pacotes TCP e UDP, e para esses protocolos, os números das portas são alteradas, para que a combinação de endereço IP e informações de porta no pacote retornado, possam ser inequivocamente mapeados, para o endereço privado correspondente e informações da porta. A RFC 2663 usa o termo tradução de endereço de rede e porta (NAPT) para este tipo de NAT. Outros nomes incluem a tradução de endereços de porta (PAT), o mascaramento de IP, a sobrecarga de NAT e NAT muitos-para-um. Esse é o tipo mais comum de NAT, e tornou-se sinônimo com o termo NAT de uso comum. Esse método permite a comunicação através do roteador somente quando a conversa se origina na rede mascarada, uma vez que essa estabelece as tabelas de tradução. Por exemplo, um navegador da Web na rede mascarada pode navegar em um site fora, mas um navegador do lado de fora não poderia navegar um site hospedado na rede mascarada. No entanto, a maioria dos dispositivos NAT hoje, permitem ao administrador da rede configurar entradas de tabela de tradução estáticas, para conexões da rede externa para a rede interna mascarada. Esse recurso é muitas vezes referido como "NAT estático" e existem dois sabores: o encaminhamento de porta, que encaminha o tráfego de uma porta externa específica para um host interno em uma porta especificada, e host DMZ, que encaminha o tráfego recebido na interface externa em qualquer número de porta para um endereço IP interno, preservando a porta de destino. Esses tipos podem ser combinados.
    - Protocolos não baseados em TCP ou UDP exigem outras técnicas de tradução. Pacotes ICMP normalmente referem-se a uma conexão existente e precisam ser mapeados usando o mesmo mapeamento de endereço IP e porta como essa conexão.
    - Exemplos de mascaramento:

        - iptables -t nat -A POSTROUTING -i eth1 -o etho -j SNAT --to-source 200.200.200.200 - adiciona uma regra de mascaramento indicando que todo tráfego que entrar pela interface eth1 e sair pela interface eth0, será mascarado com o endereço 200.200.200.200
        - iptables -t nat -A POSTROUTING -i eth1 -o eth0 -j MASQUERADE - adiciona uma regra de mascaramento indicando que todo tráfego que entrar pela interface eth1 e sair pela interface eth0, será mascarado com o endereço principal disponível

Listar e escrever filtragem e regras que aceitam ou bloqueiam datagramas baseados no protocolo de origem ou destino, porta e endereço

- Listar
    - Sintaxe:
        - iptables -t <tabela> -L
        - iptables -t <tabela> -L --line-numbers
- Escrever regras
    - Sintaxe:
        - iptables -t <tabela> -A <cadeia> <regra>
        - iptables -t <tabela> -I <cadeia> <número> <regra>
    - Exemplos:
        - iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT - adiciona a regra para aceitar conexões tcp na porta 80 ao fim da cadeia INPUT
        - iptables -t filter -I INPUT 5 -p tcp --dport 80 -j ACCEPT - adiciona a regra para aceitar conexões tcp na porta 80 na linha 5 da cadeia INPUT
- Remover regras
    - Sintaxe:
        - iptables -t <tabela> -D <cadeia> <regra>
        - iptables -t <tabela> -D <cadeia> <número>
    - Exemplos:
        - iptables -t filter -D INPUT -p tcp --dport 80 -j ACCEPT - apaga a regra para aceitar conexões tcp na porta 80 da cadeia INPUT
        - iptables -t filter -D INPUT 6 - apaga a linha 6 da cadeia INPUT
- Alterar regras
    - Sintaxe: iptables -t <tabela> -R <cadeia> <número> <regra>
    - Exemplo: iptables -t filter -R INPUT 5 -p tcp --dport 443 -j ACCEPT - altera a regra na linha 5 da cadeia INPUT para aceitar conexões tcp na porta 443

Protocolo, porta e endereço de destino

- Opções do iptables [7]
    - Parâmetros das regras
        - [!] -p, --protocol <protocolo>
            - O protocolo da regra ou do pacote, para verificar. O protocolo especificado pode um de tcp, udp, udplite, icmp, esp, ah, sctp ou all, ou pode ser um valor numérico, representando um desses protocolos ou um protocolo diferente. Um nome de protocolo vindo de /etc/protocols, também é aceito. Um argumento "!" antes do protocolo inverte o teste. O numero zero é equivalente a todos (all). Protocolo all corresponderá a todos os protocolos e é usado como padrão quando essa opção é omitida.
        - [!] -s, --source <endereço>[/máscara][,...]

- Especificação de origem. Endereço pode ser tanto um nome de rede, um nome de host, um endereço IP de rede (com /máscara), ou um endereço IP . Nomes de host serão resolvidos apenas uma vez, antes que a regra seja submetida ao kernel. Por favor note que especificando qualquer nome para ser resolvido com uma consulta remota, como DNS, é realmente uma ideia ruim. A máscara pode ser tanto uma máscara de rede ou um número. Um argumento "!" antes da especificação de endereço inverte o sentido de endereço. A flag --src é um apelido para essa opção. Múltiplos endereços podem ser especificados, mas isso irá expandir para múltiplas regras (quando adicionando com -A), o irá causar múltiplas regras serem deletadas (com -D).

■ [!] -d, --destination <endreço>[/máscara][,...]

- Especificação de destino. Veja a descrição da flag -s (origem) para uma descrição detalhada da sintaxe. A flag --dst é um apelido para essa opção.

- Extensões de correspondência [7]
  - multiport
    - Esse módulo corresponde a um conjunto de portas de origem ou destino. Até 15 portas podem ser especificadas. Uma faixa de porta (porta:porta) conta como duas portas. Esse pode ser usado apenas em conjunto com -p tcp ou -p udp.
    - [!] --source-ports,--sports porta[,porta|,porta:porta]...
      - Corresponde se a porta de origem é uma das portas dadas. A flag --sports é um apelido conveniente para essa opção. Múltiplas portas ou faixas de portas podem ser separadas usando uma vírgula, e uma faixa de porta pode ser especificada usando dois pontos.
      - 53,1024:65535 iria portanto corresponder às porta 53 e todas de 1024 até 65535.
    - [!] --destination-ports,--dports porta[,porta|,porta:porta]...
      - Corresponde se a porta de destino é uma das portas dadas. A flag --dports é um apelido conveniente para essa opção.
    - [!] --ports porta[,porta|,porta:porta]...
      - Corresponde se tanto as portas de origem ou destino são iguais a uma das portas dadas.

Salvar e recarregar configurações de filtragem

- Salvar
  - iptables-save
    - Exemplos:
      - iptables-save - exibe as regras usadas em tempo de execução, em um formato de inserção
      - iptables-save > <arquivo> - salva as regras usadas em tempo de execução, no arquivo especificado
- Recarregar
  - iptables-restore
    - Exemplos:
      - iptables -F - descarrega as regras usadas em tempo de execução (necessário para o processo de recarregamento de regras)
      - iptables-restore - insere as regras no firewall, através da entrada padrão

- iptables-restore < <arquivo> - insere as regras no firewall, através do arquivo especificado

Consciência do ip6tables e filtragem

- Arquivos de configuração
  - RH e derivados
    - /etc/sysconfig/ip6tables
    - /etc/sysconfig/ip6tables-config
  - Debian e derivados
    - /etc/iptables/rules.v6
- Ferramentas e utilitários
  - ip6tables - administração do filtro de pacotes IPv6
  - ip6tables-save - (8) despeja regras do iptables para a saída padrão
  - ip6tables-restore - (8) restaura IPv6 Tables

Termos e utilitários

- /proc/sys/net/ipv4/ - [] variáveis do IPv4
- /etc/services - banco de dados de nome de serviços
- iptables - (8) ferramenta de administração para filtragem de pacotes IPv4 e NAT

Referências

1. http://en.wikipedia.org/wiki/Iptables
2. http://en.wikipedia.org/wiki/Private_network
3. http://en.wikipedia.org/wiki/Port_forwarding
4. http://en.wikipedia.org/wiki/IP_forwarding
5. http://en.wikipedia.org/wiki/Network_address_translation#MASQUERADING
6. https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt
7. iptables(8)

Simulado

1. São comandos válidos do iptables:
   a. iptables
   b. iptables-import
   c. iptables-export
   d. iptables-xml

2. São relações corretas de tabelas com cadeias no iptables:
   a. NAT: PREROUTING / FORWARD / POSTROUTING
   b. FILTER: INPUT / FORWARD / OUTPUT
   c. MANGLE: PREROUTING / INPUT / FORWARD / OUTPUT / POSTROUTING
   d. RAW: PREROUTING / POSTROUTING / OUTPUT

3. É possível criar novas cadeias nas tabelas do iptables, porém não é possível designar uma política padrão para as cadeias criadas.

a. V
b. F

4. Através do utilitário iptables, é possível controlar o fluxo de roteamento de pacotes.
   a. V
   b. F

5. Através do utilitário iptables, não é possível definir a tabela de roteamento de pacotes.
   a. V
   b. F

6. As faixas de endereço privado podem ser publicamente roteadas.
   a. V
   b. F

7. São faixas de endereço privado para o protocolo IPv4:
   a. 10.0.0.0-10.255.255.255
   b. 172.16.0.0-172.33.255.255
   c. 169.254.0.0-169.254.255.255
   d. 192.168.0.0-192.168.255.255

8. Para redirecionar todo o tráfego entrando pela interface eth0, para o host 192.168.0.1, através do comando iptables, os argumentos … podem ser usados.

9. Para habilitar o encaminhamento de pacotes para todos os hosts da rede 192.168.0.0/24, através do iptables, o comando com argumentos … podem ser usados.

10. Para listar o conteúdo da tabela NAT, exibindo o número de linhas, através do comando iptables, os argumentos … podem ser usados.

11. Para bloquear todo o tráfego de entrada de pacotes, usando política, através do comando iptables, os argumentos … podem ser usados.

12. Para salvar a tabela de filtragem de pacotes atual para o IPv4, no arquivo ipv4.table, o comando com argumentos … pode ser usado.

13. Para limpar a tabela atual e em seguida, carregar o arquivo de filtragem de pacotes IPv4 ipv4.table, os comandos com argumentos … e … podem ser usados.

14. O ip6tables não possui a tabela NAT.
   a. V
   b. F

15. O ip6tables utiliza a mesma sequência de comandos do iptables para salvar e restaurar tabelas de filtragem.
   a. V
   b. F

212.2 Protegendo servidores FTP

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de configurar um servidor para downloads e uploads anônimos. Esse objetivo inclui precauções para serem tomadas se uploads anônimos são permitidos e configurando acesso de usuário.
Áreas de conhecimentos chave:

- Arquivos de configuração, ferramentas e utilitários para Pure-FTPd e vsftpd
- Consciência do ProFTPd
- Entendimento de conexões FTP passivas e ativas

Termos e utilitários:

- vsftpd.conf
- opções de linha de comando do Pure-FTPd importantes

Áreas de conhecimentos chave

Arquivos de configuração, ferramentas e utilitários para Pure-FTPd e vsftpd

- FTP [1]
    - O File Transfer Protocol (FTP) é um protocolo de rede padrão, usado para transferir arquivos de computador, a partir de um host para outro host, através de uma rede baseada em TCP tal como a Internet.
    - O FTP é construído em uma arquitetura cliente-servidor e usa conexões de controle e de dados separadas entre o cliente e o servidor. Usuários de FTP podem se autenticar usando um protocolo de login de texto claro, normalmente na forma de um nome de usuário e senha, mas podem ligar se anonimamente, se o servidor está configurado para permitir isso. Para a transmissão segura, que protege o nome de usuário e senha e criptografa o conteúdo, o FTP é muitas vezes protegido com SSL/TLS (FTPS). O SSH File Transfer Protocol (SFTP) é, por vezes, também usado no lugar, mas é tecnologicamente diferente.
    - As primeiras aplicações do cliente de FTP eram aplicativos de linha de comando desenvolvidos antes que os sistemas operacionais tiveram interfaces gráficas, e ainda são enviados com a maioria dos sistemas operacionais Windows, UNIX e Linux. Muitos clientes FTP e utilitários de automação têm sido desenvolvidos para desktops, servidores, dispositivos móveis e hardware, e o FTP foi incorporado a aplicativos de produtividade, como editores de páginas Web.
- Pure-FTPd [2] [3]
    - Pure-ftpd é um servidor FTP livre (licença BSD) com um forte foco na segurança de software. Ele pode ser compilado e executado em uma variedade de sistemas operacionais de computador Unix-like, incluindo Linux, OpenBSD, NetBSD, FreeBSD, DragonFly BSD, Solaris, Tru64, Darwin, Irix e HP-UX. Ele também foi portado para Android.
    - Características
        - Contas do sistema podem ter imediatamente acesso FTP. Autenticação via módulos PAM também é suportado. Contas abaixo de um uid (por exemplo <500 para contas daemon) podem ser não permitidas.

- Todas as contas podem ser facilmente “chrooted” por padrão. Para a administração fácil, um grupo “confiável" sem chroot pode ser definido.
- Contas de FTP podem ser distintas das contas do sistema, armazenados em um banco de dados independente. Várias contas podem compartilhar o mesmo id do sistema. Um banco de dados de indexação incorporado permite consultas muito rápidas. É executando com sucesso com mais de 1,5 milhões de contas no mesmo servidor. Contas do sistema podem ser copiadas para contas FTP virtuais, de modo que os usuários podem ter senhas diferentes para acesso shell e acesso FTP.
- Autenticação LDAP também é totalmente suportada. Funções hash de criptografia Plaintext, Crypt, MD5, SMD5, SHA e SSHA são implementadas. O Pure-FTPd foi testado com sucesso com OpenLDAP e iPlanet Directory Server. Ele usa as classes posixAccounts padrão.
- Hashes criptográficos seguros embutidos (SMD5, SSHA) podem ser usados com qualquer servidor LDAP, mesmo aqueles que estão faltando suporte para esses hashes.
- Informações dos usuários também podem ser centralizadas em bancos de dados MySQL, com ou sem transações. Todas as consultas são totalmente personalizáveis, e os pedidos podem ser construídos com nomes de usuário, endereços de clientes remotos, endereços IP locais e portas. Dessa forma, as regras complexas de hospedagem podem ser facilmente implementadas, mesmo com vários servidores virtuais no mesmo host, e vários domínios virtuais com muitos usuários.
- Vários métodos de autenticação podem ser encadeados em qualquer ordem. Por exemplo, contas SQL, diretórios LDAP e contas de sistema podem ser utilizadas ao mesmo tempo.
- Podem ser facilmente adicionados métodos de autenticação personalizados. O Pure-FTPd suporta módulos de autenticação externos, e escrevendo um novo backend pode ser tão simples como algumas linhas de script shell.
- O Pure-FTPd suporta um sistema de cotas virtuais: contas podem ter cota individual (número máximo de arquivos, tamanho total máximo), mesmo quando elas compartilham o mesmo uid de sistema.
- Largura de banda é suportada, com configurações distintas para upload e download.
- Cada usuário pode ser atribuído com cota individual, taxa e largura de banda.
- Cada usuário pode ter permissão para conectar-se apenas a partir de uma faixa específica de endereço IP, ou apenas para seu próprio host virtual.
- Cada usuário pode ser restrito individualmente para seu diretório home ou não.
- Cada usuário pode ter permissão para conectar-se apenas durante tempo-intervalos configurados (por exemplo, apenas durante o horário comercial).
- Um sistema anti-warez impede os usuários de negociação se eles encontram um diretório público-gravável. Arquivos de propriedade dos usuários de ftp anônimos não podem ser baixados (o sysadmin tem de moderá-los, alterando sua propriedade). Além disso, os usuários de ftp não podem criar diretórios por padrão para ocultar arquivos.
- Qualquer script shell externo pode ser chamado após um upload bem-sucedido. Scanners de vírus e arquivadores de banco de dados podem ser facilmente configurados.
- Um número máximo de conexões simultâneas do mesmo endereço IP podem ser aplicadas para evitar a falta de largura de banda e ataques de negação de serviço.

- Os downloads podem ser recusados se a carga do sistema está muito alta.
- Listagens de diretório listam um número máximo parametrizável de arquivos. Listagens recursivas são totalmente suportadas, com uma profundidade máxima parametrizável. Então pode-se fornecer pesquisa recursiva para usuários sem fornecer qualquer negação de serviço simples.
- O comando pure-ftpwho fornece relatórios em tempo real de quem está fazendo o que no servidor de FTP, incluindo o uso da banda. O resultado pode ser uma página web completa, e o programa também pode funcionar como um programa CGI padrão, compatível com qualquer servidor web. Relatórios XML e de texto também estão disponíveis, bem como um formato compacto e facilmente analisável para scripts shell.
- Os arquivos de log são precisos, e eles usam facilidades syslog padrão. Arquivos de log adicionais Apache-like (CLF) podem ser produzidos. Eles são compatíveis com todos os softwares de estatística web. Um formato estendido chamado "Estatísticas" também é implementado, e trabalha com avançados softwares de terceiros de estatística FTP como FTPStats e ModLogAn. O FTPStats fornece estatísticas detalhadas por utilizador.
- Diretórios pessoais podem ser criados sob demanda. Isso é especialmente útil com backends LDAP e SQL: basta inserir uma linha no banco de dados, e a conta está pronta para ir. Não há necessidade de criar qualquer diretório para esse usuário: ele será criado automaticamente na primeira vez que o usuário faz login.
- Vários servidores FTP virtuais podem ser hospedados no mesmo computador, com um IP confiável independente para administração.
- O acesso a dot-files pode ser restringido, de modo que os usuários não podem lêr/gravar diretórios, arquivos .ssh, .bash_history, .rhosts arquivos etc.
- Permissões de segurança são aplicadas em diretórios de usuários. Os clientes não podem desativar suas contas, por erro, com um comando inseguro "chmod 0 /". O comando "chmod" também pode ser totalmente desativado.
- Vários servidores Pure-FTPd com diferentes configurações podem ser executados no mesmo host sem qualquer conflito.
- O Pure-FTPd pode atuar como servidor FTP privado e não permitir todas as conexões anônimas independentemente da conta "ftp" do sistema. Com outro parâmetro, o servidor pode ser apenas anônimo, e recusar conexões para todas as contas de shell.
- Os links simbólicos podem ser seguidos quando os usuários estão chrooted, mesmo quando eles estão apontando para fora da jaula. Essa característica única torna o conteúdo compartilhado fácil de configurar.
- Apelidos de diretórios podem ser ativados, para fornecer atalhos para diretórios comuns.
- Uploads são verdadeiramente atômicos. Os servidores Web não vão servir imagens parciais, nem scripts PHP quebrados, quando os arquivos estão sendo enviados, mesmo quando o conteúdo está sendo atualizado.

- Arquivo de configuração
  - /etc/pure-ftpd
    - /etc/pure-ftpd/pure-ftpd.conf - arquivo de configuração para os wrappers do pure-ftpd
  - Diretivas comuns
    - chrooteveryone - chroot para todos os usuários

                - maxclientsnumber - número máximo de clientes
                - maxclientsperip - número máximo de clientes por IP
                - displaydotfiles - exibe arquivos começados por .
                - anonymousonly - apenas conexões anônimas
                - noanonymous - sem conexões anônimas
                - createhomedir - criar diretório home na primeira conexão
                - anonymouscancreatedirs - permite criação anônima de diretório
                - quota - ativa quota padrão
                - nochmod - desativa a operação chmod
                - umask - define a máscara de criação de objetos
        - Utilitários
            - pure-config.pl - wrapper de configuração para o Pure-FTPd
            - pure-ftpd(8) - servidor simples de Protocolo de Transferência de Arquivo
            - pure-ftpwho(8) - relata as sessões FTP atuais
            - pure-pw(8) - gerencia arquivos de usuários virtual para o Pure-FTPd
            - pure-quotacheck(8) - atualiza arquivos de cota virtual para o Pure-FTPd
- vsftpd [4] [5]
    - vsftpd, (ou very secure FTP daemon), é um servidor de FTP para sistemas Unix-like, incluindo Linux. Ele está licenciado sob a GNU General Public License. Ele suporta IPv6 e SSL.
    - O vsftpd suporta FTPS explícito (desde 2.0.0) e implícito (desde 2.1.0).
    - Ele é o servidor FTP padrão no Ubuntu, CentOS, Fedora, NimbleX, Slackware e distribuições RHEL Linux.
    - Em comparação com outros softwares de servidor ftp, o vsftpd é construído para ser especialmente eficaz e muito seguro.
    - Características
        - Apesar de ser pequeno para fins de velocidade e segurança, configurações de FTP muitos mais complicadas são realizáveis com o vsftpd! De nenhuma maneira uma lista exclusiva, o vsftpd irá lidar com:
            - Configurações de IP virtuais
            - Usuários virtuais
            - Operação independente ou inetd
            - Configurabilidade poderosa por usuário
            - Otimização de largura de banda
            - Configurabilidade por IP de origem
            - Limites por IP de origem
            - IPv6
            - Suporte à criptografia através da integração SSL
    - Arquivo de configuração
        - /etc/vsftpd
            - /etc/vsftpd/vsftpd.conf(5) - arquivo de configuração para vsftpd
        - Diretivas comuns
            - write_enable - permite escrita no servidor ftp
            - nopriv_user - usuário não privilegiado para execução
            - anonymous_enable - permite acesso anônimo
            - anon_upload_enable - permite postagem anônima de arquivos
            - anon_mkdir_write_enable - permite criação anônima de diretório
            - local_enable - permite acesso de usuários locais

        - chroot_local_user - habilita chroot para usuários locais
    - Utilitário
        - vsftpd(8) - Very Secure FTP Daemon

Consciência do ProFTPd

- ProfFTPd [6]
    - ProFTPd (abreviação de Pro FTP daemon) é um servidor de FTP. O ProFTPd é um software gratuito e de código aberto, compatível com sistemas Unix-like e Microsoft Windows (via Cygwin). Junto com vsftpd e Pure-FTPd, o ProFTPD está entre os servidores de FTP mais populares em ambientes Unix-like hoje. Em comparação com aqueles que se focam por exemplo na simplicidade, velocidade ou segurança, o objetivo principal do projeto ProFTPd é ser um servidor FTP altamente rico em recursos, expondo uma grande quantidade de opções de configuração para o usuário.
    - Configuração e características
        - O ProFTPd inclui uma série de opções que não estão disponíveis com muitos outros daemons FTP. A configuração do ProFTPd é realizada em um único arquivo de configuração principal, chamado /etc/proftpd/proftpd.conf. Devido às suas semelhanças com o arquivo de configuração do Servidor HTTP Apache é intuitivamente compreensível para alguém que usa esse servidor web popular.
        - Algumas das características mais notáveis são:
            - Configuração ".ftpaccess" por diretório de semelhante ao ".htaccess" do Apache
            - Vários servidores FTP virtuais e serviços de FTP anônimos
            - Executa como um servidor autônomo ou a partir do inetd/xinetd, dependendo da carga do sistema
            - Diretórios raiz FTP anônimos não exigem qualquer estrutura de diretório específica, binários do sistema ou outros arquivos de sistema
            - Nenhum comando SITE EXEC, que em ambientes de Internet modernos representam um problema de segurança
            - Diretórios e arquivos ocultos, com base em permissões do estilo Unix ou propriedade de usuário/grupo
            - Funciona como um usuário não-privilegiado configurável em modo stand-alone, a fim de diminuir as chances de ataques que podem explorar suas habilidades "root"
            - Suporte a logging e utmp/wtmp
            - Suporte a suite de senha shadow, incluindo suporte para contas expiradas
            - Design modular, permitindo o servidor ser estendido facilmente com módulos. Módulos têm sido escritos para bancos de dados SQL, servidores LDAP, criptografia SSL/TLS, suporte RADIUS, etc.
            - Suporte IPv6

Entendimento de conexões FTP passivas e ativas

- Comunicações e transferência de dados [1]
    - O FTP pode ser executado no modo ativo ou passivo, que determina como a conexão de dados é estabelecida. Em ambos os casos, o cliente cria uma conexão de controle TCP de uma aleatória, geralmente sem privilégios, porta N para a porta 21 de comando do servidor FTP. No modo ativo, o cliente começa a escutar conexões de dados recebidos do servidor na porta M. Ele envia o comando FTP PORT M para informar o servidor em qual a porta ele

está escutando. Por padrão, M = N + 1. O servidor, em seguida, inicia um canal de dados para o cliente a partir de sua porta 20, a porta de dados do servidor de FTP. Em situações em que o cliente está atrás de um firewall e incapaz de aceitar conexões TCP de entrada, o modo passivo pode ser usado. Neste modo, o cliente utiliza a ligação de controle para enviar um comando para o servidor PASV e, em seguida, recebe um número de endereço IP do servidor e porta do servidor a partir do servidor, que em seguida, o cliente usa para abrir uma conexão de dados de uma porta cliente arbitrária para o número do endereço IP do servidor e porta do servidor recebido. Ambos os modos foram atualizados em setembro de 1998 para suportar IPv6. Outras mudanças foram introduzidas para o modo passivo, nesse momento, atualizando-o para o modo passivo prolongado.

Termos e utilitários

- vsftpd.conf - (5) arquivo de configuração para vsftpd
- opções de linha de comando do Pure-FTPd importantes

Referências

1. http://en.wikipedia.org/wiki/File_Transfer_Protocol
2. http://en.wikipedia.org/wiki/Pure-FTPd
3. http://www.pureftpd.org/project/pure-ftpd
4. http://en.wikipedia.org/wiki/Vsftpd
5. https://security.appspot.com/vsftpd.html#features
6. http://en.wikipedia.org/wiki/ProFTPD

Exercícios práticos

1. Preparação para exercícios
    a. Instalar o pacote ftp (Ex.: yum install <pacote>)

    b. Instalar o pacote vsftpd (Ex.: yum install <pacote>)

    c. Instalar o pacote pure-ftpd, habilitando o repositório EPEL (Ex.: yum install --enablerepo=<repositório> <pacote>)

    d. Criar o usuário ftpuser (Ex: useradd <usuário>)

    e. Criar o usuário teste-ftp (Ex.: useradd <usuário>)

    f. Alterar o usuário e grupo do diretório /var/ftp/pub para ftp (Ex.: chown <usuário>.<grupo> <diretório>)

    g. Alterar o boleano do SELinux allow_ftp_full_access para on (setsebool allow_ftp_full_access on)

    h. Alterar o boleano do SELinux allow_ftpd_anon_write para on (setsebool allow_ftpd_anon_write on)

i. Alterar o boleano SELinux ftp_home_dir para on (setsebool ftp_home_dir on)

j. Definir uma senha para o usuário teste-ftp (Ex.: passwd <usuário>)

2. Arquivos de configuração, ferramentas e utilitários para Pure-FTPd e vsftpd
   a. Pure-FTPd
      i. Iniciar o serviço pure-ftpd (Ex. service <serviço> start)

      ii. Acessar o servidor ftp local através do cliente ftp, de forma anônima (login anonymous) (Ex.: ftp <host>)
         1. Entrar no diretório pub (Ex.: cd <diretório>)

         2. Tentar enviar o arquivo local /etc/fstab como fstab (Ex.: put <local> <remoto>)

         3. Sair do cliente ftp (Ex.: quit)

      iii. Habilitar a postagem anônima de arquivos (Ex.: vi <arquivo>)

      iv. Reiniciar o serviço pure-ftpd (Ex.: service <serviço> restart)

      v. Repetir a postagem do arquivo fstab

      vi. Remover o acesso anônimo (Ex.: vi <arquivo>)

      vii. Reiniciar o serviço pure-ftpd (Ex.: service <serviço> restart)

      viii. Tentar acessar o servidor ftp local através do cliente ftp, de forma anônima (login anonymous) (Ex.: ftp <host>)

      ix. Tentar acessar o servidor ftp local através do cliente ftp, com o login teste-ftp (Ex.: ftp <host>)

      x. Criar o arquivo /etc/ftpusers com o conteúdo teste-ftp (Ex.: vi <arquivo>)

      xi. Acessar o servidor ftp local através do cliente ftp, com o login teste-ftp (Ex.: ftp <host>)
         1. Repetir a postagem do arquivo com o usuário teste-ftp

         2. Listar o conteúdo do diretório / (Ex.: ls)

      xii. Ativar o chroot de usuários (Ex.: vi <arquivo>)

      xiii. Reiniciar o serviço pure-ftpd (Ex.: service <serviço> restart)

      xiv. Acessar o servidor ftp local através do cliente ftp, com o login teste-ftp (Ex.: ftp <host>)

1. Listar o conteúdo do diretório / (Ex.: ls)

xv. Verificar o usuário do processo pure-ftpd (Ex.: ps <opções>)

xvi. Alterar o pure-ftpd para executar com o usuário ftpsecure (Ex.: vi <arquivo>)

xvii. Reiniciar o serviço pure-ftpd (Ex.: service <serviço> restart)

xviii. Verificar o usuário do processo pure-ftpd (Ex.: ps <opções>)

xix. Parar o serviço pure-ftpd (Ex.: service <serviço> stop)

b. vsftpd

i. Iniciar o serviço vsftpd (Ex.: service <serviço> start)

ii. Acessar o servidor ftp local através do cliente ftp, de forma anônima (login anonymous) (Ex.: ftp <host>)

1. Entrar no diretório pub (Ex.: cd <diretório>)

2. Tentar enviar o arquivo local /etc/fstab como fstab (Ex.: put <local> <remoto>)

3. Sair do cliente ftp (Ex.: quit)

iii. Habilitar a postagem anônima de arquivos (Ex.: vi <arquivo>)

iv. Reiniciar o serviço vsftpd (Ex.: service <serviço> restart)

v. Repetir a postagem do arquivo fstab

vi. Remover o acesso anônimo (Ex.: vi <arquivo>)

vii. Reiniciar o serviço vsftpd (Ex.: service <serviço> restart)

viii. Tentar acessar o servidor ftp local através do cliente ftp, de forma anônima (login anonymous) (Ex.: ftp <host>)

ix. Acessar o servidor ftp local através do cliente ftp, com o login teste-ftp (Ex.: ftp <host>)

1. Repetir a postagem do arquivo com o usuário teste-ftp

2. Listar o conteúdo do diretório / (Ex.: ls)

x. Ativar o chroot de usuários (Ex.: vi <arquivo>)

xi. Reiniciar o serviço vsftpd (Ex.: service <serviço> restart)

xii. Acessar o servidor ftp local através do cliente ftp, com o login teste-ftp (Ex.: ftp <host>)
    1. Listar o conteúdo do diretório / (Ex.: ls)

xiii. Verificar o usuário do processo vsftpd (Ex.: ps <opções>)

xiv. Alterar o vsftpd para executar com o usuário ftpsecure (Ex.: vi <arquivo>)

xv. Reiniciar o serviço vsftpd (Ex.: service <serviço> restart)

xvi. Verificar o usuário do processo vsftpd (Ex.: ps <opções>)

xvii. Parar o serviço vsftpd (Ex.: service <serviço> stop)

Simulado

1. … é o arquivo de configuração do Pure-FTPd.

2. O daemon … do servidor Pure-FTPd usa parâmetros na linha comando, sendo necessário o uso do wrapper … para que a configuração seja feita a partir do arquivo de configuração do Pure-FTPd.

3. Os utilitários do Pure-FTPd, respectivos a relatar sessões ativas, gerenciar arquivos de usuários virtual e gerenciar arquivos de cota virtual são …, … e ….

4. … e ... é o arquivo de configuração e o binário do daemon do vsftpd.

5. As diretivas de configuração do vsftpd, …, …, … e … são usadas respectivamente para permitir acesso anônimo, permitir postagem anônima de arquivos, permitir acesso das contas locais e definir usuário não privilegiado para execução.

6. O ProFTPd suporta arquivos de configuração por diretório, nomeados ….

7. O ProFTPd possui o arquivo de configuração com uma sintaxe similiar à configuração do Apache HTTP Server.
    a. V
    b. F

8. Durante uma a conexão FTP, são usados dois canais: o canal de controle (geralmente usado na porta 20 do servidor FTP) e um canal de dados (geralmente N+1) que vai depender do tipo de conexão (ativo/passivo).
    a. V
    b. F

9. Quando o servidor FTP opera em modo ativo, o cliente deve habilitar seu firewall para aceitar conexões do servidor.
    a. V
    b. F

10. Em modo passivo, o servidor se conecta ao servidor para transferir dados.
    a. V
    b. F

212.3 Shell seguro (SSH)

Visão geral

Peso: 4
Descrição: Os candidatos devem ser capazes de configurar e proteger um daemon SSH. Esse objetivo inclui gerenciando chaves e configurando SSH para usuários. Candidatos também deve ser capazes de encaminhar um protocolo de aplicação sobre SSH e gerenciar o login SSH.
Áreas de conhecimentos chave:

- Arquivos de configuração, ferramentas e utilitários OpenSSH
- Restrições de login para o superusuário e usuários normais
- Gerenciando e usando chaves de servidor e cliente para login com e sem senha
- Uso de múltiplas conexões de múltiplos hosts para se proteger de perda de conexão ao host remoto após alterações de configurações

Termos e utilitários:

- ssh
- sshd
- /etc/ssh/sshd_config
- /etc/ssh/
- Arquivos de chave pública e privada
- PermitRootLogin, PubkeyAuthentication, AllowUsers, PasswordAuthentication, Protocol

Áreas de conhecimentos chave

Arquivos de configuração, ferramentas e utilitários OpenSSH

- SSH [1]
    - Secure Shell ou SSH é um protocolo de rede criptográfico (criptografado), para iniciar sessões de shell baseados em texto, em máquinas remotas de forma segura.
    - Isso permite um usuário executar comandos no prompt de comando de uma máquina sem ele estar fisicamente presente perto da máquina. Ele também permite um usuário estabelecer um canal seguro em uma rede insegura, em uma arquitetura cliente-servidor, ligando um aplicativo cliente SSH com um servidor SSH. As aplicações comuns incluem login remoto de linha de comando e execução de comando remoto, mas qualquer serviço de rede pode ser protegido com SSH. A especificação do protocolo distingue duas versões principais, referidas como SSH-1 e SSH-2.
    - A aplicação mais visível do protocolo é para acesso a contas de shell em sistemas operativos do tipo Unix, mas também pode ser utilizado de forma semelhante no Windows.
    - O SSH foi concebido como um substituto para o Telnet e outros protocolos de shell remoto inseguros, como os protocolos Berkeley rsh e rexec, que enviam informações, notavelmente senhas, em texto simples, tornando-os suscetíveis a interceptação e divulgação, por meio de análise de pacotes. A criptografia usada pelo SSH se destina a fornecer confidencialidade e integridade dos dados através de uma rede não segura, como a Internet, embora os arquivos vazados por Edward Snowden indicam que a Agência de Segurança Nacional pode, por vezes, descriptografar SSH.
- OpenSSH [2]

    - OpenSSH, também conhecido como o OpenBSD Secure Shell, é um conjunto de utilitários de nível de rede relacionados à segurança, baseados no protocolo SSH, que ajudam a proteger as comunicações de rede via criptografia do tráfego de rede sobre vários métodos de autenticação, e fornecendo capacidades de encapsulamento seguro. O OpenSSH foi concebido como uma alternativa livre e de código aberto para a implementação SSH proprietária desenvolvida por Tatu Ylönen e oferecida pela SSH Communications Security.
    - O OpenSSH é um projeto da equipe do OpenBSD e é financiado através de doações. A implementação SSH proprietária foi originalmente desenvolvida sob uma licença que permitia outros desenvolvedores ou usuários birfurcar ou criar um ramo do software com suas próprias personalizações. Como resultado, OpenSSH é uma bifurcação de uma dessas personalizações e foi lançado pela primeira vez como parte de um sistema operacional Unix-like chamado OpenBSD, em 1999. Como parte do processo de bifurcação, o OpenSSH foi lançado sob uma licença BSD, um licença de código aberto que permite manipulações abertas e contribuições. A fim de manter eficazmente o programa, o projeto OpenSSH é desenvolvido sob uma política de produção de código limpo e auditado.
    - O OpenSSH não é um único programa de computador, mas sim um conjunto de programas no sistema operacional OpenBSD que oferecem uma alternativa para os protocolos de comunicação de rede sem criptografia, como FTP e Rlogin. Enquanto o OpenSSH não é mantido ativamente para sistemas operacionais diferentes do OpenBSD, uma equipe dedicada, ocasionalmente, libera uma versão que pode ser portada ou usada em outros sistemas operacionais. Isso permitiu o OpenSSH e seus derivados faturar uma cota de mercado de quase 88% em de Julho de 2008.
- Encaminhamento de porta [3]
    - Se o servidor remoto está executando o sshd(8), pode ser possível para encapsular determinados serviços via ssh. Isso pode ser desejável, por exemplo, para criptografar conexões POP ou SMTP, mesmo que o software não suporta diretamente comunicações criptografadas. Tunnelling usa o encaminhamento de porta para criar uma conexão entre o cliente e o servidor. O software cliente deve ser capaz de especificar uma porta não-padrão para se conectar, para que isso funcione.
    - A idéia é que o usuário se conecta ao host remoto usando ssh, e especifica qual porta na máquina do cliente deve ser usada para encaminhar conexões com o servidor remoto. Depois disso, é possível iniciar o serviço que está a ser criptografado (por exemplo, o fetchmail, irc) na máquina do cliente, especificando a mesma porta local passada ao ssh, e a conexão será encapsulada através do ssh. Por padrão, o sistema executando o encaminhamento só irá aceitar conexões de si.
    - As opções mais relevantes para encapsulamento são as opções -L e -R, que permitem o usuário encaminhar conexões, a opção -D, que permite o encaminhamento de porta dinâmico, a opção -g, que permite outros hosts usar encaminhamento de porta, e a opção -f, que instrui o ssh se colocar em segundo plano após a autenticação. Veja a página de manual ssh(1) para mais detalhes.
- Arquivos de configuração
    - Configuração do cliente
        - ssh_config(5) - arquivos de configuração do cliente OpenSSH
        - A configuração do cliente pode ser obtida pelas seguintes origens, usando a seguinte ordem:
            - opções do comando
            - arquivo de configuração individual do usuário (~/.ssh/config)
            - arquivo de configuração ampla do sistema (/etc/ssh/ssh_config)

    - Configuração do servidor
        - /etc/ssh - diretório de arquivos de configuração ampla do OpenSSH
            - sshd_config(5) - arquivo de configuração do daemon OpenSSH
- Utilitários
    - ssh(1) - cliente SSH do OpenSSH (programa de login remoto)
    - sshd(8) - daemon SSH do OpenSSH

Restrições de login para o superusuário e usuários normais

- Através das opções no sshd_config
    - PermiteRootLogin - especifica se o root pode ou não logar usando ssh
    - AllowUsers - lista de usuários separada por espaço. As diretivas allow/deny são processadas na seguinte ordem: DenyUsers, AllowUsers, Deny Groups e finalmente, AllowGroup.

Gerenciando e usando chaves de servidor e cliente para login com e sem senha

- Configuração do servidor
    - PubKeyAuthentication - especifica se autenticação por chave pública é permitido - padrão sim
- Arquivos de chave do cliente
    - Chave RSA individual
        - ~/.ssh/id_rsa(ssh-keygen(1)) - chave privada rsa pessoal (protocolo 2)
        - id_rsa.pub(ssh-keygen(1)) - chave pública rsa pessoal (protocolo 2)
    - Chave DSA individual
        - ~/.ssh/id_dsa(ssh-keygen(1)) - chave privada dsa pessoal (protocolo 2)
        - id_dsa.pub(ssh-keygen(1)) - chave pública dsa pessoal (protocolo 2)
    - Utilitários
        - ssh-keygen(1) - geração de chave de autenticação, gerenciamento e conversão
        - ssh-agent(1) - agente de autenticação
        - ssh-add(1) - adiciona identidades RSA ou DSA para o agente de autenticação

Uso de múltiplas conexões de múltiplos hosts para se proteger de perda de conexão ao host remoto após alterações de configurações

Termos e utilitários

- ssh - (1) cliente SSH do OpenSSH (programa de login remoto)
- sshd - (8) daemon SSH do OpenSSH
- sshd_config(5) - arquivo de configuração do daemon OpenSSH
- /etc/ssh - diretório de arquivos de configuração ampla do
- Arquivos de chave pública e privada
- PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication, Protocol
    - PermitRootLogin - especifica se o root pode ou não logar usando ssh
    - PubKeyAuthentication - especifica se autenticação por chave pública é permitido
    - AllowUsers - lista de usuários aceitos
    - PasswordAuthentication - especifica se autenticação por senha é permitido
    - Protocol - especifica as versões de protocolo suportadas pelo sshd

Referências

1. http://en.wikipedia.org/wiki/Secure_Shell
2. http://en.wikipedia.org/wiki/OpenSSH
3. http://www.openssh.com/faq.html#2.11

Exercícios práticos

1. Preparação para exercícios
    a. Criar o usuário teste-ssh (Ex.: useradd <usuário>)

    b. Definir uma senha para o usuário teste-ssh (Ex.: passwd <usuário>)

2. Restrições de login para o superusuário e usuários normais
    a. Negar a permissão de autenticação do usuário root no servidor ssh (Ex.: vi <arquivo>)

    b. Reiniciar o serviço sshd (Ex.: service <serviço> restart)

    c. Tentar autenticar como usuário root no servidor ssh local (Ex.: ssh <usuário>@<host>)

    d. Verificar o arquivo de log de segurança (Ex.: tail <arquivo>)

    e. Retirar a restrição de autenticação do usuário root no servidor ssh (Ex.: vi <arquivo>)

    f. Reiniciar o serviço sshd (Ex.: service <serviço> restart)

    g. Autenticar como usuário root no servidor ssh local (Ex.: ssh <host>)

    h. Permitir a autenticação no servidor ssh apenas para o usuário teste-ssh (Ex.: vi <arquivo>)

    i. Reiniciar o serviço sshd (Ex.: service <serviço> restart)

    j. Tentar autenticar como usuário root no servidor ssh local (Ex.: ssh <usuário>@<host>)

    k. Autenticar como usuário teste-ssh no servidor ssh local (Ex.: ssh <usuário>@<host>)

    l. Retirar a restrição de autenticação de usuários no servidor ssh (Ex.: vi <arquivo>)

    m. Reiniciar o serviço sshd (Ex.: service <serviço> restart)

3. Gerenciando e usando chaves de servidor e cliente para login com e sem senha
    a. Autenticação por chaves RSA, com senha interativa
        i. Gerar um par chaves RSA, definindo uma senha qualquer (Ex.: ssh-keygen)

        ii. Criar o diretório .ssh no diretório home do usuário teste-ssh (Ex.: mkdir <diretório>)

        iii. Alterar a permissão do diretório criado para 700 (Ex.: chmod <permissão> <objeto>)

iv. Alterar a propriedade do diretório criado para teste-ssh (Ex.: chown <usuário>.<grupo> <objeto>)

v. Copiar o conteúdo da chave pública para o arquivo ~/.ssh/authorized_keys, no diretório home do usuário teste-ssh (Ex.: cat <arquivo> >> <arquivo>)

vi. Alterar a propriedade do arquivo gerado para teste-ssh (Ex.: chown <usuário>.<grupo> <arquivo>)

vii. Conectar por ssh no endereço localhost, com o usuário teste-ssh, informando a senha usada na chave RSA criada (Ex.: ssh <usuário>@<endereço>)

viii. Finalizar a conexão ssh (Ex.: exit)

b. Autenticação por chaves RSA, com senha por agente
   i. Executar o agente ssh (Ex.: ssh-agent)

   ii. Executar a saída gerada pelo agente ssh

   iii. Adicionar a identidade da chave privada no agente ssh (Ex.: ssh-add)

   iv. Conectar por ssh no endereço localhost, com o usuário teste-ssh (Ex.: ssh <usuário>@<endereço>)

Simulado

1. Para se conectar ao host “remoto”, com o usuário root, as seguintes opções são válidas:
   a. ssh -l root remoto
   b. ssh -u root remoto
   c. ssh root@remoto
   d. ssh --login=root remoto

2. … é o arquivo de configuração do cliente ssh, pessoal do usuário.

3. A diretiva … especifica quais versões do protocolo o daemon sshd suporta.

4. … é o diretório de configurações globais do servidor e cliente OpenSSH.

5. … é o daemon do servidor OpenSSH.

6. O software OpenSSH permite o encaminhamento de portas de forma a encapsular e criptografar dados de uma aplicação qualquer, sobre a rede.
   a. V
   b. F

7. É possível fazer um túnel reverso de encaminhamento (servidor para cliente), através da opção … do cliente ssh.

8. O cliente OpenSSH pode ser usado como um servidor SOCKS, através da opção ….

9. A diretiva AllowUsers tem precedência sobre a diretiva DenyUsers.
    a. V
    b. F

10. São formas de negar a autenticação do usuário root no daemon sshd:
    a. usar a diretiva DenyUser root
    b. usar a diretiva PermitRootLogin no
    c. usar a diretiva AllowUsers sem especificar o usuário root
    d. usar a diretiva DenyGroup root

11. São opções válidas da diretiva PermitRootLogin
    a. never
    b. true
    c. without-password
    d. forced-commands-only

12. A diretiva … permite que seja suportado autenticação por par de chaves no servidor OpenSSH.

13. Ao se usar o comando …, os arquivos ~/.ssh/.id_rsa e … são criados.

14. … é o comando referente ao agente ssh.

15. O papel do agente ssh é gerenciar as identidades usadas por um usuário, para realizar autenticação por par de chaves, com interação de senha.
    a. V
    b. F

16. Para se adicionar uma identidade ao agente ssh, o comando … deve ser usado.

17. Após a execução do agente ssh, é necessário exportar as variáveis de ambiente que identificam seu arquivo de socket e PID.
    a. V
    b. F

18. Não é possível usar múltiplos pares de chaves para autenticação em diferentes servidores ssh.
    a. V
    b. F

19. A autenticação por chaves sem uso interativo de senha é mais seguro se realizado pelo agente ssh em vez de se usar chave privada sem senha.
    a. V
    b. F

20. O uso de múltiplas conexões ssh com um host remoto é uma boa prática para se tentar evitar problemas de acesso após alterações de configuração no mesmo.
    a. V

b. F

212.4 Tarefas de segurança

Visão geral

Peso: 3
Descrição: Os candidatos devem ser capazes de receber alertas de segurança de várias fontes, instalar, configurar e executar sistemas de detecção de intrusão e aplicar patches de seguranca e bugfixes.
Áreas de conhecimentos chave:

- Ferramentas e utilitários para varrer e testar portas em um servidor
- Localização e organizações que relatam alertas de segurança como Bugtraq, CERT e outras fontes
- Ferramentas e utilitários para implementar um sistema de detecção de intrusão (IDS)
- Consciência do OpenVAS e Snort

Termos e utilitários:

- telnet
- nmap
- fail2ban
- nc
- iptables

Áreas de conhecimentos chave

Ferramentas e utilitários para varrer e testar portas em um servidor

- telnet(1) - interface de usuário para o protocolo TELNET
- nmap(1) - ferramenta de exploração de rede e segurança / varredor de portas
- nc(1) - conexões arbitrárias TCP e UDP e escuta

Localização e organizações que relatam alertas de segurança como Bugtraq, CERT e outras fontes

- Bugtraq [1]
    - Bugtraq é uma lista de endereços eletrônicos dedicados a questões sobre a segurança de computador. Questões em tópicos são novas discussões sobre vulnerabilidades, anúncios relacionados com a segurança do fornecedor, métodos de exploração, e como corrigí-los. É uma lista de discussão de alto volume, e quase todas as novas vulnerabilidades são discutidas lá. Esse fórum oferece um veículo para fabricantes de software e de sistema para se comunicar de forma orientada com a sua base instalada, para informá-los de novas vulnerabilidades, para que possam ser rapidamente resolvidas. Do ponto de vista da empresa, ele também fornece uma visão consolidada de vulnerabilidades, eliminando a necessidade de tentar rastrear anúncios de vendedores individuais; bem como proporcionar um fórum para buscar informações de seus pares.
    - A Bugtraq foi criada em 5 de Novembro de 1993 por Scott Chasin em resposta às falhas percebidas da infra-estrutura de segurança da Internet existente do momento, particularmente o CERT. A política da Bugtraq era publicar vulnerabilidades, independentemente da resposta do fornecedor, como parte do movimento divulgação completa de divulgação de vulnerabilidades.
    - Elias Levy, também conhecido como Aleph One (aludindo ao número cardinal aleph um), observou em uma entrevista que "o ambiente naquele momento era tal que os fornecedores não estavam fazendo nenhum patch. Assim, o foco era sobre como corrigir software que as empresas não estavam corrigindo. "

    - A lista de discussão não foi moderada originalmente, mas a relação sinal-ruído, eventualmente, tornou-se inaceitavelmente ruim. A moderação começou em 5 de Junho de 1995. Elias Levy moderou a lista de 14 de Junho de 1996 até que ele deixou o cargo em 15 de Outubro de 2001. David Mirza Ahmad, um dos muitos co-autores de “Hack Proofing Your Network, Second Edition”, assumiu do Levy e continuou até que ele deixou o cargo em 23 de Fevereiro de 2006. David McKinney, analista de ameaças DeepSight da Symantec, assumiu a partir de Ahmad, embora moderação agora foi passado para outro analista DeepSight, Prasanna.
    - A Bugtraq foi originalmente hospedada no Crimelab.com. Ela foi transferida para a Brown University NetSpace Project - que já foi reorganizada como a Fundação NetSpace - em 5 de Junho de 1995, o mesmo dia em que a sua moderação começou. Em Julho de 1999, tornou-se propriedade da SecurityFocus e foi movida para lá. A SecurityFocus foi adquirida integralmente pela Symantec em 6 de Agosto de 2002.
- CERT [2]
    - Equipes de Resposta a Emergências de Computador (CERT) são grupos de peritos que lidam com incidentes de segurança de computador. Os nomes alternativos para tais grupos incluem Equipe de Prontidão a Emergências de Computador e Equipe de Resposta a Incidentes de Segurança de Computador (CSIRT).
    - O nome equipe de resposta a emergências de computador é a designação histórica para a primeira equipe (CERT-CC) da Carnegie Mellon University (CMU). A abreviação CERT do nome histórico foi pego por outras equipes ao redor do mundo. Algumas equipes assumiram o nome mais específico de CSIRT a apontar a tarefa de lidar com incidentes de segurança de computador em vez de outro trabalho de suporte técnico, e porque a CMU estava ameaçando tomar medidas legais contra indivíduos ou organizações que se referiam a qualquer outra equipe diferente do CERT-CC como um CERT. Após a virada do século, a CMU relaxou sua posição, e os termos CERT e CSIRT são agora usados como sinônimos.
    - A história do CERT está ligada à existência de malware, especialmente vermes e vírus de computador. Sempre que uma nova tecnologia chega, seu mau uso não é muito tempo em seguida. O primeiro worm no IBM VNET foi encoberto. Pouco depois, um worm atingiu a Internet em 3 de Novembro de 1988, quando o chamado Morris Worm paralisou uma boa porcentagem dela. Isso levou à formação da primeira equipe de resposta a emergências de computador da Universidade Carnegie Mellon sob contrato do Governo dos EUA. Com o enorme crescimento no uso de tecnologias de informação e comunicação ao longo dos anos seguintes, o termo agora genérico "CERT”/”CSIRT” refere-se a uma parte essencial de grandes estruturas da maioria das empresas. Em muitas organizações o CERT evolui para um centro de operações de segurança da informação.
- Outras Fontes
    - Banco de Dados de Vulnerabilidades Nacional [3]
        - O National Vulnerability Database é o repositório do governo dos EUA de dados de gerenciamento de vulnerabilidades baseado em padrões representados usando a Protocolo de Automação de Conteúdo de Segurança (SCAP). Esses dados permitem a automação de gerenciamento de vulnerabilidades, medida de segurança e conformidade. O NVD inclui bases de dados de listas de verificação de segurança, falhas de software relacionados com a segurança, configurações incorretas, nomes de produtos e métricas de impacto. Ele suporta o Programa de Automação de Segurança da Informação (ISAP).
        - Além de fornecer uma lista de Vulnerabilidades e Exposições Comuns (CVEs), o NVD pontua os CVEs para quantificar o risco de vulnerabilidades, calculado a partir

de um conjunto de equações com base em métricas tais como a complexidade de acesso e disponibilidade de um remédio.

- Packet Storm [4]
  - Tempestade de Pacotes de Segurança é um popular site de segurança da informação oferecendo ferramentas de segurança de computador atuais e históricos, exploits, e avisos de segurança. É operado por um grupo de entusiastas de segurança que publicam novas informações de segurança e oferecem ferramentas para fins educacionais e de teste.

Ferramentas e utilitários para implementar um sistema de detecção de intrusão (IDS)

- IDS [5]
  - Um sistema de detecção de intrusão (IDS) é um dispositivo ou aplicativo de software que monitora rede ou atividades de sistema, por atividades maliciosas ou violações de políticas, e produz relatórios para uma estação de gerenciamento. O IDS vêm em uma variedade de "sabores" e se aproxima do objetivo de detectar tráfego suspeito de maneiras diferentes. Existem sistemas de detecção de intrusão baseados em rede (NIDS) e host (HIDS) . Alguns sistemas podem tentar parar uma tentativa de intrusão, mas isso não é necessário nem esperados de um sistema de monitoramento. Sistemas de detecção e prevenção de intrusão (IDPS) concentram-se principalmente sobre a identificação de possíveis incidentes, registrando informações sobre eles, e relatando tentativas. Além disso, as organizações utilizam IDPSes para outros fins, tais como a identificação de problemas com as políticas de segurança, documentando as ameaças existentes e dissuadindo as pessoas de violar as políticas de segurança. Os IDPSes tornaram-se um complemento necessário para a infra-estrutura de quase todas as organizações de segurança.
  - O IDPSes tipicamente registram informações relacionadas a eventos observados, notificam os administradores de segurança sobre eventos observados importantes e produzem relatórios. Muitos IDPSes também podem responder a uma ameaça detectada pela tentativa de impedir que suceda. Eles usam várias técnicas de resposta, que envolvem os IDPS parando o ataque em si, mudando o ambiente de segurança (por exemplo, reconfiguração de um firewall) ou alterando o conteúdo do ataque.
- fail2ban [6]
  - Fail2ban é um framework de software de prevenção de intrusão que protege servidores de computador de ataques de força bruta. Escrito na linguagem de programação Python, é capaz de rodar em sistemas POSIX que têm uma interface para um sistema de controle de pacotes ou firewall instalado localmente, por exemplo, iptables ou TCP Wrapper.
  - Funcionalidade
    - O Fail2ban opera monitorando arquivos de log (por exemplo /var/log/auth.log, /var/log/apache/access.log, etc.) para as entradas selecionadas e executando scripts com base nelas. Mais comumente esse é usado para bloquear endereços IP selecionados que podem pertencer aos hosts que estão tentando violar a segurança do sistema. Ele pode proibir qualquer IP do host que faz muitas tentativas de login ou realiza qualquer outra ação indesejada, dentro de um prazo definido pelo administrador. O Fail2ban é tipicamente configurado para remover a restrição de um host bloqueado dentro de um determinado período, de modo a não "bloquear" qualquer conexão genuína que podem ter sido temporariamente configurada incorretamente. No entanto, um tempo de desbloqueio de vários minutos é geralmente suficiente para parar uma conexão de rede de ser inundada por

conexões maliciosas, bem como reduzir a probabilidade de um ataque de dicionário bem-sucedido.
        - O Fail2ban pode executar várias ações sempre que um IP abusivo é detectado: atualizar Netfilter/iptables ou regras do firewall PF, tabela hosts.deny do TCP Wrapper para rejeitar o endereço IP de um abusador; notificações de e-mail; ou qualquer ação definida pelo utilizador que pode ser levada a cabo por um script Python.
        - A configuração padrão embarca com filtros para Apache, Lighttpd, sshd, vsftpd, qmail, Postfix e Courier Mail Server. Os filtros são definidos por expressões regulares Python, que podem ser convenientemente personalizados por um administrador familiarizado com expressões regulares. A combinação de um filtro e uma ação é conhecida como uma "prisão" e é o que faz com que um host mal-intencionado seja impedidos de acessar os serviços de rede especificados. Bem como os exemplos que são distribuídos com o software, uma "prisão" pode ser criada por qualquer processo voltado para a rede, que cria um arquivo de registro de acesso.
        - "Fail2ban é semelhante ao DenyHosts [...] mas, ao contrário DenyHosts que incide sobre SSH, o fail2ban pode ser configurado para monitorar qualquer serviço que escreve tentativas de login em um arquivo de log, e em vez de usar /etc/hosts.deny apenas para bloquear IP endereços/hosts, o fail2ban pode usar Netfilter/iptables e TCP Wrappers /etc/hosts.deny."
    - Arquivos de configuração
        - /etc/fail2ban - diretório de configuração do fail2ban
            - /etc/fail2ban/fail2ban.conf - arquivo de configuração do fail2ban
            - /etc/fail2ban/action.d - diretório de configuração de ações
            - /etc/fail2ban/filter.d - diretório de configuração de filtros
    - Utilitários
        - fail2ban-client(1) - configura e controla o servidor
        - fail2ban-regex(1) - testa a opção failregex do fail2ban
        - fail2ban-server(1) - inicia o servidor
- Firewall
    - iptables(8) - ferramenta de administração para filtragem de pacote e NAT IPv4

Consciência do OpenVAS e Snort

- OpenVAS [7] [8]
    - OpenVAS (Sistema de Avaliação de Vulnerabilidade Aberto, o nome do fork originalmente conhecido como GNessUs) é uma estrutura de vários serviços e ferramentas que oferecem uma solução de varredura de vulnerabilidade e gerenciamento de vulnerabilidades.
    - Todos os produtos OpenVAS são Software Livre. A maioria dos componentes são licenciados sob a GPL.
    - História
        - O OpenVas começado, sob o nome de GNessUs, como um fork da ferramenta de varredura previamente de código aberto Nessus, após a Tenable Network Security mudar para uma licença proprietária (de fonte fechada) em Outubro de 2005. O OpenVAS foi originalmente proposto por testadores de penetração (pentesters) no Portcullis Computer Security e, em seguida, anunciado por Tim Brown no Slashdot.
        - OpenVAS é um projeto membro do Software no Interesse Público.
    - Visão geral das características

        - OpenVAS Scanner
            - Muitos hosts de destino são varridos simultaneamente
            - OpenVAS Transfer Protocol (OTP)
            - Suporte SSL para OTP (sempre)
            - Suporte WMI (opcional)
        - OpenVAS Manager
            - OpenVAS Management Protocol (OMP)
            - Banco de dados SQL (sqlite) para configurações e os resultados da verificação
            - Suporte SSL para OMP (sempre)
            - Muitas tarefas de varreduras simultâneas (muitos OpenVAS Scanners)
            - Notas de gestão para resultados de varredura
            - Gerenciamento de falsos positivos para os resultados da verificação
            - Verificações agendadas
            - Escaladores flexíveis sob o status de uma tarefa de verificação
            - Parar, pausar e retomar as tarefas de verificação
            - Modo Master-Slave para controlar muitas instâncias de uma central
            - Framework de Plugin de Formato de Relatórios com vários plugins para: XML, HTML, látex, etc.
            - Gerenciamento de usuários
            - Exibir o status de alimentação
            - Sincronização de alimentação
        - Greenbone Security Assistant (GSA)
            - Cliente para OMP e OAP
            - HTTP e HTTPS
            - Servidor web por conta própria (microhttpd), portanto, nenhum servidor web extra necessário
            - Sistema de ajuda online integrada
            - Suporte multi-idioma
        - OpenVAS CLI
            - Cliente para OMP
            - Funciona em Windows, Linux, etc.
            - Plugin para o Nagios
- Snort [9]
    - Snort é um sistema de prevenção de intrusão à rede (NIPS) livre e de origem aberto e sistema de detecção de intrusão de rede (NIDS) criado por Martin Roesch em 1998. O Snort é agora desenvolvido pela Sourcefire, dos quais Roesch é o fundador e CTO . Em 2009, o Snort entrou no Open Source Hall of Fame da InfoWorld como um dos "maiores software de fonte aberta de todos os tempos".
    - Uso
        - O sistema de detecção de intrusão baseada em rede (NIDS), open source, do Snort tem a capacidade de executar análise de tráfego em tempo real e registro de pacotes em redes de Internet Protocol (IP). O Snort realiza análise de protocolo, a pesquisa de conteúdo, e o conteúdo correspondente. Esses serviços básicos tem muitas finalidades, incluindo qualidade de serviço desencadeada de reconhecimento de aplicativos, para despriorizar o tráfego em massa quando os aplicativos sensíveis à latência estão em uso.

- O programa também pode ser usado para detectar sondas ou ataques, incluindo, mas não limitado a, tentativas de impressão digital de sistemas operacionais, interface de gateway comum, buffer overflows, sondas de bloco de mensagens do servidor, e varredores de portas ocultos.
- O Snort pode ser configurado em três modos principais:. sniffer, logger de pacotes e detecção de intrusão de rede. No modo sniffer, o programa irá ler os pacotes de rede e exibi-los no console. No modo logger de pacotes, o programa vai registrar pacotes para o disco. No modo de detecção de intrusão, o programa irá monitorar o tráfego de rede e analisá-lo contra um conjunto de regras definidas pelo usuário. O programa irá, em seguida, executar uma ação específica com base no que foi identificado.

Termos e utilitários

- telnet - (1) interface de usuário para o protocolo TELNET
- nmap - (1) ferramenta de exploração de rede e segurança / varredor de portas
- fail2ban - (1) um conjunto programas cliente e servidor para limitar tentativas de autenticação por força bruta
- nc - (1) conexões arbitrárias TCP e UDP e escuta
- iptables - (8) ferramenta de administração para filtragem de pacote e NAT IPv4

Referências

1. http://en.wikipedia.org/wiki/Bugtraq
2. http://en.wikipedia.org/wiki/Computer_emergency_response_team
3. http://en.wikipedia.org/wiki/National_Vulnerability_Database
4. http://en.wikipedia.org/wiki/Packet_Storm
5. http://en.wikipedia.org/wiki/Intrusion_detection_system
6. http://en.wikipedia.org/wiki/Fail2ban
7. http://en.wikipedia.org/wiki/OpenVAS
8. http://www.openvas.org/software.html
9. http://en.wikipedia.org/wiki/Snort_(software)

Simulado

1. São ferramentas que podem ser usadas para verificar teste de conectividade de portas:
    a. nc
    b. telnet
    c. ping
    d. iptables

2. Através do comando nc, é possível realizar a varredura de portas de um determinado host.
    a. V
    b. F

3. O comando telnet permite teste em portas udp.
    a. V

b. F

4. Através do comando nmap, para se realizar varredura de todas as portas em todos os hosts da rede 192.168.0.0/16, os argumentos … podem ser usados.

5. CVE é um formato de dados usado para informar vulnerabilidades.
   a. V
   b. F

6. O Bugtraq é um site onde se tem informações sobre vulnerabilidades e suas correções.
   a. V
   b. F

7. O CERT é um termo usado para designar equipes de resposta a incidentes de segurança.
   a. V
   b. F

8. O software fail2ban pertence a categoria NIPS.
   a. V
   b. F

9. O comando … é usado para controlar o servidor fail2ban.

10. … é o arquivo de configuração do fail2ban.

11. Uma diferença do software fail2ban para o denyhosts é que o denyhosts trabalha com bloqueios utilizando o iptables.
    a. V
    b. F

12. Através do iptables, usando um software IPS, é possível bloquear um tráfego malicioso.
    a. V
    b. F

13. O OpenVAS é um software HIDS.
    a. V
    b. F

14. O OpenVAS possibilita testes de penetração em hosts remotos.
    a. V
    b. F

15. O Snort é um software HIDS e por tanto, não permite tomada de ações preventivas.
    a. V
    b. F

212.5 Open VPN

Visão geral

Peso: 2
Descrição: Os candidatos devem ser capazes de configurar uma VPN (Rede Virtual Privada) e criar conexões seguras ponto a ponto e site a site.
Áreas de conhecimentos chave:

- OpenVPN

Termos e utilitários:

- /etc/openvpn/
- openvpn

Áreas de conhecimentos chave

OpenVPN

- VPN [1]
  - Uma rede privada virtual (VPN) estende-se de uma rede privada através de uma rede pública, como a Internet. Ele permite que um computador ou dispositivo habilitado para rede envie e receba dados através de redes públicas ou compartilhadas como se estivesse conectado diretamente à rede privada, e beneficiando das políticas da rede pública de funcionalidade, segurança e gerenciamento. Uma VPN é criada através do estabelecimento de uma conexão virtual ponto-a-ponto através do uso de conexões dedicadas, protocolos de encapsulamento virtuais, ou criptografia de tráfego. As principais implementações de VPNs incluem OpenVPN e IPsec.
  - A conexão VPN na Internet é semelhante a um link de rede de área ampla (WAN) entre sites. Da perspectiva do usuário, os recursos de rede expandidos são acessados da mesma maneira como os recursos disponíveis dentro da rede privada. Uma grande limitação de VPNs tradicionais é que eles são ponto-a-ponto, e não tendem a suportar ou conectar domínios de broadcast. Portanto, comunicação, software e rede, que são baseados em camada 2 e broadcast de pacotes, tais como NetBIOS usados na rede do Windows, podem não ser totalmente suportados ou funcionar exatamente como fariam em uma verdadeira LAN. Variantes sobre VPN, como Virtual Private LAN Service (VPLS), e protocolos de túnel de camada 2, são projetados para superar essa limitação.
  - VPNs permitem os funcionários acessar com segurança a intranet da sua empresa, enquanto viajam fora do escritório. Da mesma forma, VPNs conectam com segurança escritórios geograficamente separados de uma organização, criando uma rede coesa. A tecnologia VPN também é usada por usuários individuais da Internet para proteger suas transações sem fio, para contornar as restrições geográficas e censura, e se conectar a servidores de proxy para o propósito de proteger a identidade pessoal e localização.
- OpenVPN [2]
  - OpenVPN é uma aplicação de software de código aberto que implementa técnicas de rede privada virtual (VPN) para criar conexões seguras ponto-a-ponto ou site-a-site em configurações roteadas ou em ponte e facilidades de acesso remoto. Ele usa um protocolo de segurança personalizado que utiliza SSL/TLS para troca de chaves. Ele é capaz de

percorrer conversores de endereço de rede (NAT) e firewalls. Foi escrito por James Yonan e é publicado sob a GNU General Public License (GPL).

- O OpenVPN permite pares autenticar um ao outro usando uma chave secreta pré-compartilhada, certificados, ou nome de usuário/senha. Quando usado em uma configuração multicliente-servidor, ele permite o servidor liberar um certificado de autenticação para cada cliente, utilizando assinatura e certificado de autoridade. Ele usa a biblioteca de criptografia OpenSSL extensivamente, bem como os protocolos SSLv3/TLSv1, e contém muitos recursos de segurança e controle.
- O OpenVPN foi portado e incorporado a vários sistemas. Por exemplo, o DD-WRT tem a função de servidor OpenVPN. O SoftEther VPN, um servidor VPN multi-protocolo, tem uma implementação do protocolo OpenVPN.
- Arquitetura
  - Encriptação
    - O OpenVPN usa a biblioteca OpenSSL para fornecer criptografia de ambos os canais de dados e de controle. Ele permite o OpenSSL fazer todo o trabalho de criptografia e autenticação, permitindo o OpenVPN utilizar todas as cifras disponíveis no pacote OpenSSL. Ele também pode usar o recurso de autenticação de pacotes HMAC para adicionar uma camada adicional de segurança para a conexão (referida como uma "HMAC Firewall" pelo criador). Ele também pode usar a aceleração de hardware para obter um melhor desempenho de criptografia. Suporte para PolarSSL está disponível a partir da versão 2.3.
  - Autenticação
    - O OpenVPN tem várias maneiras de autenticar pares uns com os outros. Ele oferece autenticação com base em chaves pré-compartilhadas, em certificado e em nome de usuário/senha. A chave secreta pré-compartilhada é a mais fácil, com base em certificado, sendo a mais robusta e rica em recursos. Na versão 2.0 autenticações por nome de usuário/senha podem ser ativadas, tanto com ou sem certificados. No entanto, para fazer uso de autenticações nome de usuário/senha, o OpenVPN depende de módulos de terceiros. Veja o parágrafo extensibilidade para mais informações.
  - Rede
    - O OpenVPN pode rodar sobre transportes User Datagram Protocol (UDP) ou Transmission Control Protocol (TCP), multiplexando túneis de SSL criados em uma única porta TCP/UDP (RFC 3948 para UDP). A partir da série 2.3.x, o OpenVPN suporta plenamente IPv6 como protocolo de rede virtual dentro de um túnel e as aplicações OpenVPN também podem estabelecer conexões via IPv6. Ele tem a capacidade de trabalhar com a maioria dos servidores proxy (incluindo HTTP) e é bom em trabalhar através da tradução de endereços de rede (NAT) e sair através de firewalls. A configuração do servidor tem a capacidade de "empurrar" algumas opções de configuração de rede para os clientes. Essas incluem endereços IP, os comandos de roteamento, e algumas opções de conexão. O OpenVPN oferece dois tipos de interfaces para a rede através do driver Universal TUN/TAP. Ele pode criar um túnel de camada 3 baseado em IP (TUN), ou um de camada 2 baseado em TAP Ethernet que pode transportar qualquer tipo de tráfego Ethernet. O OpenVPN pode usar, opcionalmente, a biblioteca de compressão LZO para comprimir o fluxo de dados. A porta 1194 é o número da porta

oficial atribuída pela IANA para o OpenVPN. Novas versões do programa agora padronizam para essa porta. Uma característica na versão 2.0 permite a um processo administrar vários túneis simultâneos, em oposição à restrição inicial de "um túnel por processo" na série 1.x.
        - O uso no OpenVPN de protocolos de rede comuns (TCP e UDP) faz com que seja uma alternativa desejável para o IPsec em situações em que um ISP pode bloquear protocolos VPN específicos, a fim de forçar os usuários a se inscrever em um com preços mais elevados, "série business", tier de serviço.
    - Segurança
        - O OpenVPN oferece vários recursos de segurança interna. Tem-se a criptografia de até 256 bits através da biblioteca OpenSSL, embora alguns provedores de serviços podem oferecer taxas mais baixas efetivamente fazendo a conexão mais rápida. Ele é executado no espaço do usuário, em vez de exigir operação de pilha IP (e, portanto kernel). O OpenVPN tem a capacidade de deixar de usar privilégios de root, usar o mlockall para evitar a troca de dados sensíveis para o disco, inserir uma jaula após a inicialização e aplicar um contexto SELinux após a inicialização.
        - O OpenVPN executa um protocolo de segurança personalizado com base em SSL e TLS. Ele oferece suporte de cartões inteligentes através de tokens criptográficos baseados em PKCS#11.
    - Extensibilidade
        - O OpenVPN pode ser estendido com plug-ins de terceiros ou scripts que podem ser chamados em pontos de entrada definidos. A finalidade disso é muitas vezes estender o OpenVPN com log mais avançado, autenticação reforçada com nome de usuário e senhas, atualizações dinâmicas de firewall, integração RADIUS e assim por diante. Os plug-ins são módulos carregáveis dinamicamente, geralmente escritos em C, enquanto a interface de scripts pode executar qualquer script ou binários disponíveis para o OpenVPN. No código-fonte do OpenVPN existem alguns exemplos de tais plug-ins, incluindo um plug-in de autenticação PAM. Vários plug-ins de terceiros também existem para autenticar contra o LDAP ou bases de dados SQL como MySQL e SQLite. Há uma visão geral sobre muitas desses extensões no wiki relacionado do projeto para a comunidade OpenVPN.
- Arquivos de configuração
    - /etc/openvpn - diretório dos arquivos de configuração do OpenVPN
        - /etc/openvpn/*.conf - arquivos de configuração do OpenVPN (criados manualmente sob demanda)
        - /etc/openvpn/*.sh - scripts de configuração do OpenVPN (criados manualmente sob demanda)
        - /etc/openvpn/openvpn-startup - script executado durante a inicialização do serviço (se existir)
        - /etc/openvpn/openvpn-shutdown - script executado durante o encerramento do serviço (se existir)
    - Diretivas comuns
        - local - endereço IP de escuta da instância
        - port - porta de escuta da instância
        - nobind - clientes não precisam escutar uma porta local
        - proto - protocolo usado pela instância (tcp/udp)

        - dev - irá criar um túnel IP roteado (tun) ou ethernet (tap)
        - ca - certificado da CA
        - cert - certificado da instância
        - key - chave privada da instância
        - server - executa a instância em modo servidor (túnel IP) e especifica a subrede
        - server-bridge - executa a instância em modo bridge (túnel ethernet) e especifica a subrede
        - client - executa a instância em modo cliente
        - remote - especifica o endereço do servidor remoto
        - push - empurra parâmetros para os clientes
        - keepalive - configura o tempo de mensagens keepalive e timeout
        - comp-lzo - usa compressão lzo
        - max-clients - define o número máximo de clientes
        - user - define o usuário que executa a instância
        - group - define o grupo que executa a instância
        - status - arquivo de estado exibindo conexões atuais
        - log - arquivo de log
        - verb - nível de log de saída de informações
        - mute - silencia mensagens repetidas
- Utilitários
    - openvpn(8) - daemon de túnel IP protegido
- Exemplos de configuração
    - Modo servidor ponto a ponto
        - port 1194
        - proto udp
        - dev tun
        - ca ca.crt
        - cert server.crt
        - key server.key
        - server 10.8.0.0 255.255.255.0
        - keepalive 10 120
        - comp-lzo
        - status openvpn-status.log
        - verb 3
    - Modo cliente ponto a ponto
        - client
        - dev tun
        - proto udp
        - remote servidor1 1194
        - nobind
        - ca ca.crt
        - cert server.crt
        - key server.key
        - comp-lzo
        - verb 3
    - Modo servidor site a site
        - port 1194
        - proto udp

        - dev tap0
        - ca ca.crt
        - cert server.crt
        - key server.key
        - server-bridge 192.168.8.4 255.255.255.0 192.168.8.128 192.168.8.254
        - keepalive 10 120
        - comp-lzo
        - status openvpn-status.log
        - verb 3
    - Modo cliente site a site
        - client
        - dev tap
        - proto udp
        - remote servidor1 1194
        - nobind
        - ca ca.crt
        - cert server.crt
        - key server.key
        - comp-lzo
        - verb 3

Termos e utilitários

- /etc/openvpn - diretório dos arquivos de configuração do OpenVPN
- openvpn - (8) daemon de túnel IP protegido

Referências

1. http://en.wikipedia.org/wiki/Virtual_private_network
2. http://en.wikipedia.org/wiki/OpenVPN
3. http://openvpn.net/howto.html

Simulado

1. São implementações de VPN:
    a. IPSEC
    b. OpenVPN
    c. VPLS
    d. VLAN

2. O OpenVPN possibilita autenticação de clientes através de uma ICP.
    a. V
    b. F

3. O OpenVPN pode ser usado no modo cliente ou servidor, com túnel IP ou bridge.
    a. V
    b. F

4. Através de um túnel IP, é possível se usar aplicações que dependam de broadcast.
    a. V
    b. F

5. … é o diretório de configuração do OpenVPN.

6. Para iniciar uma instância do OpenVPN, como cliente de túnel IP, as 3 diretivas devem ser usadas: …, … e ….

7. No modo servidor de bridge, a diretiva de dispositivo deve informar o dispositivo final e não a categoria do mesmo.
    a. V
    b. F

8. As diretivas …, … e … são usadas respectivamente para compressão lzo, número máximo de clientes e chave privada.

9. Para cada arquivo .conf encontrado no diretório /etc/openvpn, uma instância do software será criada.
    a. V
    b. F

10. Para definir uma instância no modo bridge, usando como IP do servidor 192.168.0.1 e a faixa de clientes 192.168.0.100 a 192.168.0.200, no mínimo as diretivas … e … devem ser criadas.

Respostas dos Simulados

Tópico 207

Objetivo 207.1

1 - todas	2 - b	3 - a	4 - a	5 - options, zone, view, controls, logging, acl
6 - -t ns .	7 - @8.8.8.8 example.com mx	8 - directory, options, /etc/named.conf	9 - view internal { zone "." in { type hint; file "named.ca"; }; };	10 - todas
11 - reload example.com	12 - flush	13 - a	14 - a	15 - b

Objetivo 207.2

1 - zone	2 - b	3 - recursion false	4 - notify	5 - dig google.com soa
6 - a	7 - a	8 - c	9 - a	10 - CNAME
11 - todas	12 - a	13 - a	14 - b	15 - 10.in-addr.arpa

Objetivo 207.3

1 - key, managed-keys, trusted-keys	2 - b	3 - a	4 - a	5 - a
6 - a	7 - a	8 - a	9 - a	10 - dnssec-keygen, dnssec-signzone

Tópico 208

Objetivo 208.1

1 - /etc/httpd/conf/httpd.conf	2 - apache2ctl ou apachectl	3 - a	4 - ErrorLog, LogLevel	5 - LogFormat, CustomLog
6 -	7 - a	8 - a	9 - /etc/php.ini	10 - a

mod_authz_hosts

11 - .htaccess	12 - htpasswd	13 - prefork, worker	14 - MaxClients	15 - StartServers, MinSpareServers, MaxSpareServers
16 - NameVirtualHost	17 - a	18 - a	19 - Alias, Redirect, RedirectMatch	20 - mod_rewrite, RewriteBase, RewriteRule

Objetivo 208.2

1 - a, c	2 - openssl.cnf	3 - d	4 - openssl genrsa -des3 -out server1.key 2048	5 - openssl req -new
6 - CA.pl	7 - a	8 - SSLCertificateChainFile	9 - SSLCertificateFile	10 - SSLProtocol, SSLCipherSuite
11 - a	12 - a	13 - a	14 - a	15 - a

Objetivo 208.3

1 - /etc/squid/squid.conf	2 - a	3 - b	4 - a, c	5 - a
6 - b	7 - a	8 - acl <nome> dstdomain "/etc/squid/sites-bloqueados.txt", http_access deny <nome>	9 - acl <nome> time 01:00-04:00, http_access deny <nome>	10 - a

Objetivo 208.4

1 - a	2 - todas	3 - /etc/nginx/nginx.conf	4 - b	5 - a
6 - proxy_cache_path	7 - a	8 - b	9 - autoindex	10 - a

Tópico 209

Objetivo 209.1

1 - a	2 - Samba-3-HOWTO, Samba-3-ByExample	3 - a	4 - a	5 - /etc/samba/smb.conf, /etc/samba/smbusers, /etc/samba/lmhosts
6 - a, c	7 - a	8 - workgroup	9 - net	10 - testparm
11 - nmblookup	12 - smbpasswd, net	13 - smbstatus	14 - mount	15 - a
16 - smbd, nmbd	17 - nmbd	18 - b	19 - guest = John Doe	20 - /etc/samba/smbusers
21 - a	22 - b	23 - b	24 - a	25 - security, global

Objetivo 209.2

1 - /etc/exports	2 -	3 - a	4 - rpcinfo	5 - a
6 - b	7 - a	8 - b	9 - no_root_squash	10 - b
11 - vers=3	12 - no_wdelay	13 - rpcbind	14 - a	15 - a

Tópico 210

Objetivo 210.1

1 - dhcpd.conf	2 - a	3 - a	4 - BOOTP	5 - dhcpd.leases
6 - a	7 - arp	8 - a	9 - host	10 - dhcrelay

Objetivo 210.2

1 - a	2 - a	3 - a	4 - b	5 - a
6 - a	7 - session, password, account, auth	8 - a	9 - a	10 - pam_limits, pam_listfile, pam_unix, pam_cracklib
11 - 000	12 - a	13 - b	14 - b	15 - a

Objetivo 210.3

1 - a	2 - a	3 - LDIF	4 - 389	5 - BIND
6 - c, d	7 - ldapdelete, -r	8 - ldapsearch	9 - -x, -W, -D, -c	10 - a

Objetivo 210.4

1 - slapd.conf	2 - slapindex	3 - slapadd, slapcat	4 - a	5 - a
6 - todas	7 - a, c, d	8 - b, c, d	9 - a	10 - a
11 - a	12 - modrdn	13 - modify	14 - a	15 - a
16 - b	17 - todas	18 - a	19 - a	20 - a

Tópico 211

Objetivo 211.1

1 - a	2 - a	3 - a, b	4 - a	5 - todas
6 - /etc/postfix/main.cf	7 - a	8 - mailbox_size_limit, virtual_mailbox_limit	9 - /etc/postfix/virtual	10 - a
11 - todas	12 - SMTP	13 - a	14 - MX	15 - b
16 - a	17 - a	18 - a	19 - b	20 - b

Objetivo 211.2

1 - modo, condições, ação	2 - a	3 - a	4 - b	5 - c
6 - a	7 - a	8 - /etc/aliases	9 - ~/.procmailrc	10 - a

Objetivo 211.3

1 - a	2 - a	3 - a	4 - /etc/courier/imapd, /etc/courier/imapd-ssl	5 - /etc/courier/pop3d/ etc/courier/pop3d
6 - b	7 - a	8 -	9 - a	10 - b

		/etc/dovecot/dovecot.conf		

Tópico 212

Objetivo 212.1

1 - a, d	2 - b, c	3 - a	4 - a	5 - a
6 - b	7 - a, c, d	8 - -t nat -A POSTROUTING -i eth0 -j DNAT --to-destination 192.168.0.1	9 - -t filter -A FORWARD --src 192.168.0.0/24 -j ACCEPT	10 - -t nat -L --line-numbers
11 - -t nat -P INPUT DROP	12 - iptables-save > ipv4.table	13 - iptables -F, iptables-restore < ipv4.table	14 - a	15 - a

Objetivo 212.2

1 - /etc/pure-ftpd/pure-ftpd.conf	2 - pure-ftpd, pure-config.pl	3 - pure-ftpwho, pure-pw, pure-quotacheck	4 - /etc/vsftpd/vsftpd.conf, vsftpd	5 - anonymous_enable, anon_upload_enable, local_enable, nopriv_user
6 - .ftpaccess	7 - a	8 - b	9 - a	10 - b

Objetivo 212.3

1 - a, c	2 - ~/.ssh/config	3 - Protocol	4 - /etc/ssh	5 - sshd
6 - a	7 - -R	8 - -D	9 - b	10 - todas
11 - c, d	12 - PubKeyAuthentication	13 - ssh-keygen, ~/.ssh/id_rsa.pub	14 - ssh-agent	15 - b
16 - ssh-add	17 - a	18 - b	19 - a	20 - a

Objetivo 212.4

1 - a, b	2 - a	3 - b	4 - -p1-192.168.0.0/16	5 - a

6 - b	7 - a	8 - b	9 - fail2ban-client	10 -fail2ban.conf
11 - b	12 - a	13 - b	14 - a	15 - b

Objetivo 212.5

1 - a, b	2 - a	3 - a	4 - b	5 - /etc/openvpn/
6 - client, dev, remote	7 - a	8 - comp-lzo, max-clients, key	9 - a	10 - dev tap0, server 192.168.0.1 255.255.255.0 192.168.0.100 192.168.0.200