Wagner de Almeida Junior

Kerberos com Backend LDAP: Análise e Implantação

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. Herlon Ayres Camargo

Lavras
Minas Gerais - Brasil
2011

Wagner de Almeida Junior

Kerberos com Backend LDAP: Análise e Implantação

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *30 de Abril de 2011*

Profa. Ana Paula Piovesan Melchiori

Prof. Arlindo Follador Neto

Prof. Herlon Ayres Camargo
(Orientador)

Lavras
Minas Gerais - Brasil
2011

À minha mãe Maria Cristina, pelo amor, confiança e apoio incondicionais.

Agradecimentos

Agradeço a Deus pela iluminação nos momentos de dúvida.

Agradeço também à minha namorada Paula pelo apoio, compreensão e companheirismo.

Por fim, agradeço a meu mestre e orientador Herlon que tanto me ensinou nesses últimos anos.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

A heterogeneidade de usuários e sistemas que compõem uma rede de computadores torna necessária a implementação de diversos serviços que visam a atender as necessidades, facilitar o acesso e manter a segurança e a estabilidade da rede. Tais serviços muitas vezes requerem que os usuários identifiquem-se a fim de determinar se possuem ou não permissão de acesso. Essa identificação normalmente é feita através de uma senha, conhecida pelo usuário e pelo sistema. No entanto, quanto maior a quantidade de serviços disponíveis, mais senhas o usuário deve memorizar e, como regra geral, à medida que a quantidade de senhas aumenta, a qualidade de cada uma delas diminui. Para centralizar a autenticação de usuários, garantir a segurança das senhas e ainda integrar diversos serviços a uma única base de dados, surge como alternativa a implementação de um sistema Kerberos com *backend* LDAP.

Palavras-Chave: Kerberos; LDAP; Autenticação.

Capítulo 1

Introdução

Atualmente as redes de computadores estão presentes na maioria das empresas e organizações e, graças à popularização do acesso à Internet, é crescente também o número de pessoas que possuem os mais diversos tipos de dispositivos conectados a uma rede local em sua própria casa. Essas redes, sejam elas domésticas ou de grandes corporações, vem se tornando cada vez mais complexas e agregando diversos serviços que podem facilitar o uso, a administração ou acrescentar novas funcionalidades.

Em uma rede doméstica o número de usuários é restrito e o gerenciamento dos serviços geralmente é simples. No entanto, o número de dispositivos e pessoas que acessam a rede de computadores em uma organização tende a ser proporcional ao tamanho da mesma. Dessa maneira a configuração e manutenção de serviços que visam facilitar a administração pode se tornar complexa, tendo em vista que cada um deles necessita de informações que geralmente são armazenadas em bancos de dados diferentes. Exemplificando: se para cada serviço utilizado, entre os quais pode-se citar *proxy*, *webmail* e compartilhamento de arquivos, exista a necessidade de efetuar o cadastro de um usuário e uma senha, cria-se um transtorno tanto para o administrador que é o responsável por efetuar cadastros, quanto para o usuário que precisa memorizar diversos *logins* e senhas.

Outro problema relacionado a redes que possuem múltiplos usuários é a segurança. Empresas normalmente possuem uma estrutura organizacional bem definida e deve haver restrições de acesso a informações e serviços. Para isso é necessário que haja, na rede de computadores, uma forma de autenticação e uma política de autorização. Isso é válido tanto para os usuários da rede local quanto para quem acessa algum tipo de serviço através da Internet. A captura de senhas

de usuários em uma rede corporativa pode se tornar uma fonte potencial de espionagem industrial ou de obtenção de informações pessoais sobre os funcionários.

A partir da análise dessas duas questões é possível perceber a necessidade da implementação de soluções que ajudem a aumentar a segurança e a organizar as informações, principalmente de usuários, de forma a diminuir ou mesmo eliminar a redundância.

O protocolo Kerberos[1], analisado em (JORDÃO, 2005), foi desenvolvido como alternativa para a autenticação segura e centralizada em redes de computadores. Através de sua implementação são eliminadas as transferências de senhas pela rede e garantidas a segurança e a confiabilidade entre servidores e clientes.

Para facilitar a implementação do Kerberos, uma base de dados única e centralizada que elimine a redundância de informações na rede faz-se necessária. O protocolo LDAP (*Lightweight Directory Access Protocol*[2]), em sua implementação OpenLDAP[3] é uma excelente ferramenta para trabalhar em conjunto com o Kerberos e facilitar ainda mais a administração da rede. Tanto o protocolo quanto sua implementação podem ser vistos também em (MACHADO; JUNIOR, 2006).

Este trabalho visa a apresentar o protocolo Kerberos e o sistema de diretórios LDAP, discutir suas características, funcionamento, principais implementações e apresentar as vantagens e desvantagens do uso de cada um deles. A integração entre os dois serviços também será explicada e analisada e as configurações necessárias para tal serão demonstradas.

Embora esses dois protocolos e sua integração sejam o foco desse trabalho, serão ainda mostradas possibilidades de uso prático dessa solução na forma de autenticação de usuários e compartilhamento de arquivos.

O trabalho será realizado em caráter experimental, portanto será desenvolvido em uma rede de testes completamente virtualizada. O sistema operacional usado no desenvolvimento da implementação será o Ubuntu Linux Versão 10.04[4]. Para configuração em outros sistemas alguns ajustes serão necessários.

O texto encontra-se organizado como se segue.

O Capítulo 2 apresenta o protocolo Kerberos, suas origens, conceitos, componentes, vantagens e desvantagens e explica seu funcionamento básico.

[1] http://web.mit.edu/Kerberos/

[2] Protocolo Leve de Acesso a Diretório, em tradução livre do autor

[3] http://www.openldap.org

[4] http://www.ubuntu.com/

O Capítulo 3 define o que é um serviço de diretórios, apresenta as origens do LDAP e do padrão X.500 e faz uma breve apresentação da implementação *open source* OpenLDAP.

O Capítulo 4 demonstra as vantagens da integração entre Kerberos e LDAP, apresenta seus requisitos demonstrando como atendê-los e trata da instalação e configuração dos serviços.

O Capítulo 5 mostra os resultados da integração entre Kerberos e LDAP e adiciona possibilidades de integração de outros serviços ao sistema.

O Capítulo 6 apresenta comentários e observações finais bem como possibilidades de expansão e trabalhos futuros.

Capítulo 2

Apresentando o protocolo Kerberos

Este capítulo apresenta as origens do protocolo Kerberos, os elementos que motivaram o seu desenvolvimento e o seu funcionamento.

2.1 As Origens

O nome Kerberos advém da mitologia grega de Cerberus, o guardião dos portões do submundo. Os gregos acreditavam que para lá eram enviadas as almas dos mortos e era trabalho de Cerberus garantir que apenas eles entrassem no reino e que nenhuma alma pudesse sair dele. Na maioria das versões da história, Cerberus é um cão feroz de três cabeças a que poucos se arriscaram enfrentar. Embora o nome Cerberus tenha sido popularizado, a grafia correta para o seu nome em grego é Kerberos. Não por acaso, o protocolo Kerberos nasceu de um projeto desenvolvido a partir de maio de 1983 pelo *Massachusetts Institute of Technology* (MIT[1]) chamado Athena[2].

O advento das redes de comutação de pacotes mudou muito a maneira como os usuários interagiam com computadores. Se antes eles constituíam um recurso

[1] `http://web.mit.edu/`

[2] Assim como o cão Kerberos, Athena é um personagem da mitologia grega. Ela é a deusa da guerra e da sabedoria

caro e centralizado, acessado através de terminais leves[3] em um sistema de compartilhamento de tempo, com a rede os usuários passaram a ter seus próprios computadores pessoais conectados a todos os outros computadores de determinada organização, por exemplo. No entanto, os sistemas pessoais chamados de *desktops*, não possuíam grande poder de processamento.

Isso tornou necessário que alguns serviços fossem disponibilizados por computadores com maior poder de processamento, chamados de servidores. Esse modelo, conhecido como cliente-servidor, possibilitou diversas novas situações e o MIT percebeu que era necessária uma mudança dramática na arquitetura de *software* e na maneira de se usar o computador. A principal mudança em relação ao modelo antigo era que, se antes o computador central que todos os usuários utilizavam era controlado pelo administrador, agora cada usuário podia controlar seu próprio sistema da maneira que desejasse, tornando-se assim não mais confiável.

Em resposta a essa situação, foi desenvolvido o projeto Athena. O foco do projeto era desenvolver estratégias e *software* para a integração de computadores no MIT. Embora seu objetivo tenha sido inicialmente educacional, muito do que foi desenvolvido na época incluindo o protocolo Kerberos, ainda está em uso atualmente. Mais sobre o projeto Athena pode ser visto em (GARMAN, 2003).

2.2 Introdução ao Protocolo

Uma das principais formas que um usuário de computador possui para se identificar é através de uma senha. Essa senha deve ser conhecida pelo sistema no qual se deseja autenticar. No modelo de terminais leves, cada usuário possuía uma conexão serial direta com o servidor e sua senha era enviada através dessa conexão única e exclusiva. Além disso, uma única senha era utilizada para acessar todos os serviços disponíveis.

As senhas sempre foram um problema para o usuário, pois é difícil memorizar as longas cadeias de caracteres contendo números, letras e símbolos que compõem uma boa senha. Também de nada adianta uma senha complexa se ela está anotada embaixo do teclado ou mesmo em um papel colado no monitor. A descentralização dos recursos computacionais ocasionada pelo advento das redes de modelo cliente-servidor agravou esse problema, uma vez que em uma organização o nú-

[3]Terminais leves são computadores que possuem pouca ou nenhuma capacidade de processamento e armazenamento, dependendo de um servidor para executar tarefas. Mais sobre o assunto pode ser visto em (SINCLAIR; MERKOW, 1999).

mero de sistemas aos quais o usuário tem acesso pode ser grande e cada um deles vai precisar de uma senha diferente.

Sob o aspecto técnico, embora na maioria das vezes as senhas sejam ocultadas no momento da inserção seja por asteriscos ou qualquer outro símbolo, grande parte dos serviços como `telnet` e `ftp` utilizam a rede para trafegar senhas, estejam elas em texto puro ou criptografadas. Essas senhas podem ser facilmente interceptadas e descobertas por outros usuários da rede.

Pensando nesses problemas, o projeto Athena desenvolveu o protocolo Kerberos para autenticação em rede. A ideia básica era estender o serviço de autenticação dos terminais leves para uma rede de comutação de pacotes. O novo sistema deveria centralizar a confiança em máquinas controladas estritamente pelos administradores e, ao mesmo tempo, encriptar toda a transmissão entre essas máquinas e as outras da rede.

Atualmente diversos serviços suportam integração ao Kerberos, entre eles `telnet`, `rlogin`, `ftp`, `samba`, `nfs`[4]. O Kerberos encontra-se na versão 5[5]. A versão 4 foi descontinuada devido à fragilidade do protocolo criptográfico DES (*Data Encryption Standard*), único suportado nessa versão. As versões anteriores eram limitadas e foram usadas somente para testes.

Existem outras implementações do Kerberos disponíveis, distribuídas sob as mais diferentes licenças. Entre elas, pode-se destacar o Heimdal Kerberos[6], que foi desenvolvido na Suécia e é distribuído sob uma licença semelhante à dos sistemas BSD[7].

A implementação abordada nesse estudo será a do MIT. Além de ser a versão original, ela vem por padrão nos repositórios do Ubuntu.

2.3 Funcionamento

O Kerberos é um serviço seguro, de autenticação única e mútua através de uma terceira parte confiável. O serviço é seguro uma vez que as senhas nunca são enviadas pela rede. A autenticação é única porque os usuários precisam logar-se apenas uma vez para acessar todos os serviços da rede e é mútua pois garante a identidade não

[4] Informações sobre esses serviços podem ser vistas em (NEMETH; SNYDER; HEIN, 2009), (KUROSE; ROSS, 2010)

[5] Definida em (KHOL; NEUMANN, 2005)

[6] http://www.h5l.org/

[7] http://www.bsd.org/

só do usuário, mas também do servidor. O termo terceira parte confiável refere-se ao fato de que o Kerberos trabalha através de servidores centralizados nos quais todos os sistemas na rede confiam.

O objetivo do Kerberos é aumentar a segurança e a conveniência para os administradores de redes e usuários. Ele opera através de um ou mais servidores centralizados chamados *Key Distribution Centers*, ou KDCs. Cada KDC possui um banco de dados contendo todos os *logins* e senhas dos usuários da rede. A centralização é conveniente para o administrador à medida em que ele passa a ter que se preocupar em manter um único banco de dados. Além disso, se todas as informações encontram-se centralizadas em uma máquina ou em um pequeno grupo delas, torna-se mais fácil manter a segurança e a confidencialidade.

O Kerberos adiciona segurança a redes inseguras. Ao invés de enviar senhas através da rede ele faz uso de *tickets* encriptados para comprovar a identidade dos usuários.

2.3.1 Encriptação e Integridade

Para entender de maneira mais completa o funcionamento do Kerberos é interessante definir alguns conceitos, entre eles os de encriptação e integridade.

A palavra criptografia é originada a partir de duas palavras gregas, *cryptos*, que significa segredo ou escondido e *graphein*, que significa escrita. A encriptação é o processo de tornar secreta ou escondida uma determinada informação, geralmente na forma de texto. Isso é feito usando-se um algorítimo chamado cifra para converter o conteúdo do texto de modo que ele se torne ilegível a todos, exceto àqueles que possuam determinada informação ou chave. Essa chave é utilizada no processo de desencriptação, que é tornar novamente legível o texto encriptado. A encriptação impede que determinada informação seja acessada por pessoas não autorizadas mesmo no caso de interceptação.

O Kerberos utiliza vários algorítimos de encriptação de dados, sendo os principais na versão 5 o *triple* DES, sucessor do DES e o RC4[8], utilizado principalmente em implementações da *Microsoft*. É importante utilizar um algorítimo de criptografia forte para evitar que usuários mal intencionados consigam facilmente desencriptar informações interceptadas.

A encriptação oferece privacidade, mas é importante saber se a mensagem enviada chegou a seu destino da maneira como foi concebida, sem alterações. Para

[8]http://www.wisdom.weizmann.ac.il/ itsik/RC4/rc4.html

isso existem algorítimos conhecidos como *hashes* especializados em manter a integridade da mensagem. Os *hashes* são funções matemáticas que recebem uma informação e, através de uma fórmula, transformam-na em uma combinação de caracteres de tamanho fixo que a representam. Como as funções *hash* são de via única torna-se difícil fazer o caminho inverso e descobrir a mensagem original através de sua saída. Entre os algorítimos de integridade suportados pelo Kerberos encontram-se CRC-32[9], MD5[10] e SHA1[11]. Mais sobre criptografia pode ser encontrado em (FERGUSON; SCHNEIER, 2003).

2.3.2 Componentes do Kerberos

2.3.2.1 Reinos

Cada implementação do Kerberos define um reino sobre o qual ela terá controle administrativo. O reino é um conjunto de sistemas conectados em rede que utiliza e confia no servidor Kerberos para se autenticar. Normalmente o reino Kerberos definido dentro de um determinado domínio possui o mesmo nome do domínio convertido para letras maiúsculas.

Dessa forma, o domínio meudominio.org seria o reino MEUDOMINIO.ORG. Esse tipo de definição, embora facilite a configuração, não é obrigatório, podendo haver um reino MEUREINO.COM ou MeuReino.COM dentro de um domínio meudominio.org, ressaltando que a nomenclatura dos reinos, ao contrário da dos nomes de domínio, é sensível ao caso. Dessa forma, MEUREINO.COM e MeuReino.COM são exemplos de reinos diferentes.

2.3.2.2 *Principals*

Principals ou diretores, em tradução livre do autor, são associações feitas a cada entidade que deverá ser autenticada pelo Kerberos, seja ela um usuário, máquina ou serviço. Cada *principal* possui um nome único e uma chave, na forma de senha, que o identifica. Para garantir que um *principal* seja único, a sua nomenclatura é dividida de forma hierárquica.

[9] http://www.ciphersbyritter.com/ARTS/CRCMYST.HTM
[10] http://tools.ietf.org/html/rfc1321
[11] http://tools.ietf.org/html/rfc4634

Para usuários, a primeira parte do *principal* é o nome, seguido ou não de uma ou mais instâncias opcionais, seguido de @ e o nome do reino. O exemplo a seguir é a forma mais simples de definir um usuário:

usuario@MEUDOMINIO.ORG

As instâncias opcionais são usadas em duas situações, para definir usuários com privilégios administrativos e especificar serviços ou máquinas. Um usuário com permissões de administrador pode ser representado da seguinte forma:

usuario/admin@MEUDOMINIO.ORG

Principals associados a serviços e máquinas também são necessários em um reino Kerberos, uma vez que a autenticação deve ser mútua. No caso dos serviços, ao invés do nome de usuário, é utilizado o nome do serviço seguido do nome completo de domínio (FQDN[12]) da máquina no qual ele está instalado. Por exemplo:

ftp/ftp.meudominio.org@MEUDOMINIO.ORG

Máquinas são representadas pelo nome *host* seguido de seu FQDN:

host/pc1.meudominio.org@MEUDOMINIO.ORG

2.3.2.3 *Tickets*

O Kerberos trabalha com o conceito de *tickets*. Um *ticket* é um conjunto de informações encriptadas que confirma a identidade de um *principal*. Ele possui duas funções: a primeira é identificar os participantes de uma transação, e a segunda é estabelecer uma chave de sessão de curta duração que será usada pelas partes para estabelecer uma comunicação segura. Isso dispensa o tráfego de senhas todas as vezes que uma autenticação é solicitada. Dentre as informações que formam um *ticket*, encontramos:

- o nome do *principal* solicitante;
- o nome do *principal* do serviço solicitado;
- a partir de quando o *ticket* é válido e quando ele expira;
- uma lista de endereços IP a partir dos quais o *ticket* pode ser usado;
- uma chave de sessão encriptada que é usada na comunicação entre o usuário e o serviço.

[12] *Fully Qualified Domain Name*

Tickets possuem tempo de validade limitada, geralmente variando entre 8 e 24 horas. Isso acontece para minimizar a ameaça que um *ticket* roubado possa causar, e simultaneamente, manter a autenticação única por um prazo de tempo razoável.

Um tipo especial de *ticket*, que é sempre o primeiro a ser emitido em um processo de autenticação, é o *Ticket Granting Ticket*, ou TGT. Quando um cliente deseja autenticar-se, ele recebe um TGT encriptado com a sua senha. Caso seja informada a senha correta, ele passa a ter permissão para solicitar novos *tickets* para serviços específicos. O TGT é o principal responsável pela autenticação única.

Por padrão, *tickets* são armazenados em um arquivo temporário. Neste arquivo são armazenados o *principal* do usuário e todos os *tickets* obtidos por ele. A Figura 2.1 mostra um exemplo de visualização de um arquivo temporário.

```
$ klist
Ticket cache: FILE:/tmp/krb5cc_502_auJKaJ
Default principal: usuario@MEUDOMINIO.ORG
Valid starting     Expires            Service principal
06/07/10 10:32:17  06/09/10 20:32:17  krbtgt/MEUDOMINIO.ORG@MEUDOMINIO.ORG
06/07/10 10:32:20  06/09/10 20:32:17  host/pc1.meudominio.org@MEUDOMINIO.ORG
06/07/10 11:10:32  06/09/10 20:32:17  host/pc2.meudominio.org@MEUDOMINIO.ORG
```

Figura 2.1: Armazenamento de *tickets*

2.3.2.4 *Key Distribution Center*

O *Key Distribution Center*, ou KDC, é a parte central de um sistema Kerberos. É ele o responsável por armazenar os dados dos *principals* e autenticá-los. O KDC é composto por três partes lógicas:

- banco de dados - cada KDC deve armazenar todos os *principals* contidos em um reino, bem como suas chaves e diversas outras informações opcionais. Para isso, deve haver um sistema de banco de dados. As principais implementações do Kerberos possuem um banco de dados leve e especializado, que é executado na mesma máquina do KDC. O objetivo deste trabalho é apresentar o LDAP como alternativa para esse banco de dados, o que facilita o trabalho do KDC, já que os dados não ficarão armazenados no servidor e ainda permite a integração, de maneira simples, de vários serviços.

- servidor de autenticação - é o responsável por emitir o TGT encriptado para os clientes que queiram autenticar-se no reino.
- *ticket grantig server* - diferente do servidor de autenticação, o TGS emite *tickets* específicos de cada serviço aos clientes. Ele recebe do cliente um pedido de *ticket* que inclui o nome do *principal* representando o serviço requerido e o TGT emitido pelo Servidor de Autenticação. O TGS então emite para o cliente o *ticket* relativo ao serviço requisitado.

A Figura 2.2[13] exemplifica a autenticação de um usuário em um servidor KDC. Inicialmente, o cliente se autentica e recebe o *Ticket Granting Ticket* do servidor de autenticação. O TGT é então enviado ao TGS que deve emitir o *ticket* específico para o serviço solicitado. O servidor recebe o *ticket* e deve descritpografá-lo com sua chave, obtendo dessa forma uma chave de sessão de duração limitada.

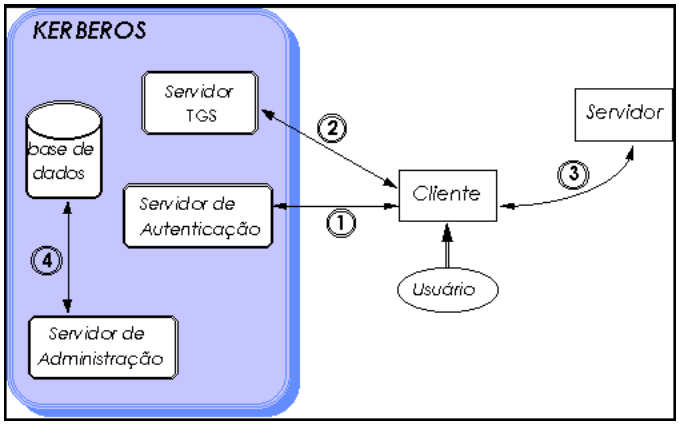

Figura 2.2: Exemplo de autenticação em um servidor Kerberos.

2.3.3 Problemas do Kerberos

Embora o grande objetivo do Kerberos seja resolver problemas de autenticação e encriptação em uma rede, ele também é dotado de falhas que devem ser observadas e previstas pelo administrador.

Como o Kerberos centraliza a autenticação de todos os serviços da rede, ele reduz toda a estrutura organizacional a um único ponto de falha. Embora esse

[13]Fonte: http://www.gta.ufrj.br/grad/02_2/kerberos/Kerberos.htm

problema possa ser minimizando através do uso de redundância de servidores, havendo qualquer dano ou defeito no(s) servidor(es) Kerberos, todo o sistema de autenticação será prejudicado.

A situação fica ainda mais grave no caso de invasão ou ataque. Obtenção de acesso de *root* ou de um *principal* que seja administrador do Kerberos, interceptação de *tickets* e ataques de negação de serviço (*Denial of Service* - DoS), são apenas alguns dos perigos aos quais o administrador deve estar atento e que podem comprometer toda a rede. O Kerberos possui mecanismos para evitar alguns desses ataques, mas ainda assim é necessária a configuração de um bom *firewall*.

Além dos problemas de segurança, a implantação do Kerberos é bastante complexa, e pode ser ainda mais em uma rede que já esteja em produção. Isso porque não há ferramentas para a migração de dados de usuários para a base do KDC. Além disso, existe a necessidade de adequar todos os serviços usados à autenticação do Kerberos, o que pode ser muito trabalhoso caso algum serviço não possua suporte nativo.

2.4 Comentários Finais

Nesse capítulo foram apresentadas as origens do Kerberos, a partir do projeto Athena, e os motivos que influenciaram o seu desenvolvimento. Também foi mostrada uma visão geral sobre o protocolo e seu funcionamento, seus principais componentes, características e alguns de seus problemas.

Uma vez que o Kerberos faz uso de *tickets* criptografados, foi necessário também definir conceitos básicos sobre criptografia e integridade de arquivos.

O próximo capítulo mostrará o sistema de diretórios LDAP, que armazenará as informações de *principals* e servirá de *backend* para o Kerberos.

Capítulo 3

Diretórios e o LDAP

Este capítulo apresenta a definição do serviço de diretórios, bem como o protocolo LDAP e sua implementação livre OpenLDAP.

3.1 Introdução ao Serviço de Diretórios

Um servidor de diretórios é um repositório que contém informações sobre diversos objetos e é organizado de forma a facilitar a sua consulta. De maneira similar a um banco de dados, em um servidor de diretórios são armazenados dados variados que poderão ser recuperados a qualquer momento. No entanto, há muitas diferenças entre um SGBD[1] e um serviço de diretórios, entre as quais se pode destacar:

- otimização para leituras - diretórios são projetados para oferecer um volume muito maior de consultas do que de inserções e por isso são organizados de forma a agilizar o processo de leitura de dados. Por outro lado, servidores de diretórios não possuem métodos avançados para controle de transações e travamentos (*locks*), ao passo que SGBDs, que são projetados para fornecer e receber dados aproximadamente com a mesma frequência, devem levar em consideração várias operações de escrita e travamento de registros.
- estrutura hierárquica - um banco de dados é organizado em tabelas e registros, enquanto que no diretório é utilizada uma estrutura em formato de árvore, similar ao modelo do DNS[2]. Essa estrutura possibilita mobilidade

[1] Sistema Gerenciador de Bancos de Dados

[2] Informações sobre o DNS podem ser vistas em (TANENBAUM, 2003)

no agrupamento de dados, além de permitir que partes do diretório sejam armazenadas em servidores fisicamente distintos.

- padronização de dados - o formato de todos os dados armazenados em um diretório deve seguir uma padronização definida em RFC[3]. Isso faz com que os tipos de dados tenham nomenclatura específica e formato consistente, além de possibilitar sua constante extensão.

- capacidades avançadas de pesquisa - em um diretório podem ser feitas buscas por tipos de registro, valores exatos, aproximados ou, até mesmo, por variações fonéticas.

3.1.1 Estrutura de um servidor de diretórios

Como dito anteriormente, em um diretório os dados são organizados de maneira semelhante a uma árvore. Essa estrutura hierárquica, onde cada ramo está ligado diretamente a um superior, facilita a representação de grupos organizacionais.

A Figura 3.1 exemplifica um modelo de diretório e a Tabela 3.1 apresenta uma explicação sobre os atributos utilizados.

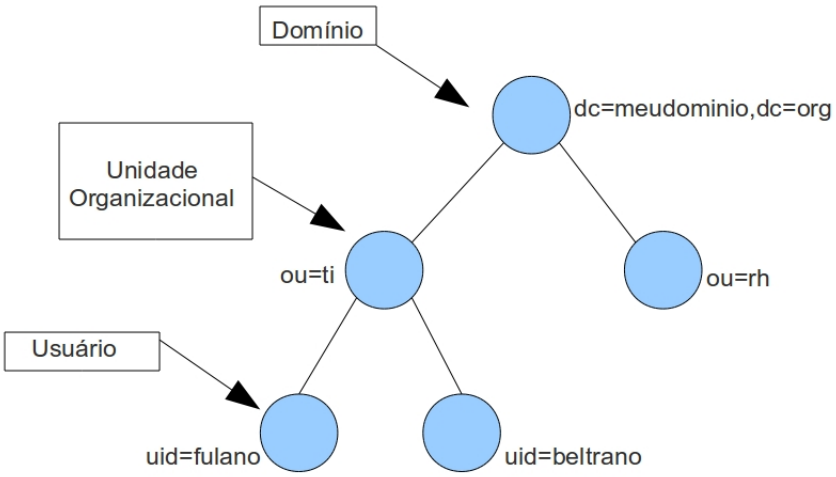

Figura 3.1: Exemplo de Diretório

[3] *Request for Comments* - http://www.ietf.org/rfc.html

Tabela 3.1: Atributos de uma entrada em um diretório

Sigla	*Objeto*	*Uso*
dc	*Domain Component*	Representa o domínio
ou	*Organizational Unit*	Representa um setor ou divisão dentro do domínio
uid	*User ID*	Identifica um usuário

O armazenamento de dados em um diretório é feito de forma a seguir a hierarquia dos objetos a que pertence a entrada, dessa forma um usuário pode ser identificado da seguinte maneira:

uid=fulano, ou=ti, dc=meudominio, dc=org

Essa identificação é chamada de DN (*Distinguished Name*). O DN sempre deve conter todos os ramos da árvore, desde o usuário (uid=fulano) até a base (dc=meudominio, dc=org), devendo ser assim único em todo o diretório.

3.2 LDAP

Em uma rede corporativa é bastante comum contar com uma gama cada vez maior de serviços, entre os quais podemos citar como exemplo um servidor DHCP que contém informações sobre endereços IP, endereços MAC, nomes de *hosts*, entre outros. Todas essas informações ou parte delas podem estar contidas também em um servidor DNS. A mesma situação se aplica a usuários, diversos serviços como *e-mail* e *proxy* armazenam *logins* e senhas muitas vezes duplicados.

Isso torna-se um problema para o administrador à medida que qualquer operação, seja ela inclusão, alteração ou remoção de dados torna-se trabalhosa e repetitiva. Obviamente, é possível desenvolver *scripts* que automatizem essas funções, mas uma solução prática e definitiva é a centralização de toda a informação redundante em um único local.

Esse é o grande potencial do LDAP: consolidar diversos serviços em um único diretório que pode ser acessado por aplicações de diferentes fornecedores.

3.2.1 Origens do Protocolo

Desde a popularização do telefone, as empresas de telecomunicações sentiam a necessidade de usar um sistema que pudesse agrupar as informações sobre números de telefone de maneira global. Em 1988, em um esforço conjunto, foi desen-

volvido pela ITU(*International Telecommunication Union*) a especificação de um padrão chamado X.500.

O serviço de diretórios X.500 era originalmente acessado através do DAP (*Directory Access Protocol*). Esse protocolo realizava a comunicação entre cliente e servidor utilizando a pilha de protocolo de sete camadas da OSI (*Open Systems Interface*). O padrão OSI é extremamente didático e amplamente utilizado no meio acadêmico para definir a base de desenvolvimento de protocolos de rede, no entanto na prática ele é muito complexo.

O LDAP surgiu originalmente como uma alternativa "leve"[4] para acesso ao serviço de diretórios X.500, através do uso da pilha de protocolo mais simples e hoje em dia amplamente utilizada TCP/IP. A Figura 3.2[5] apresenta uma comparação entre a pilha OSI e a pilha TCP/IP.

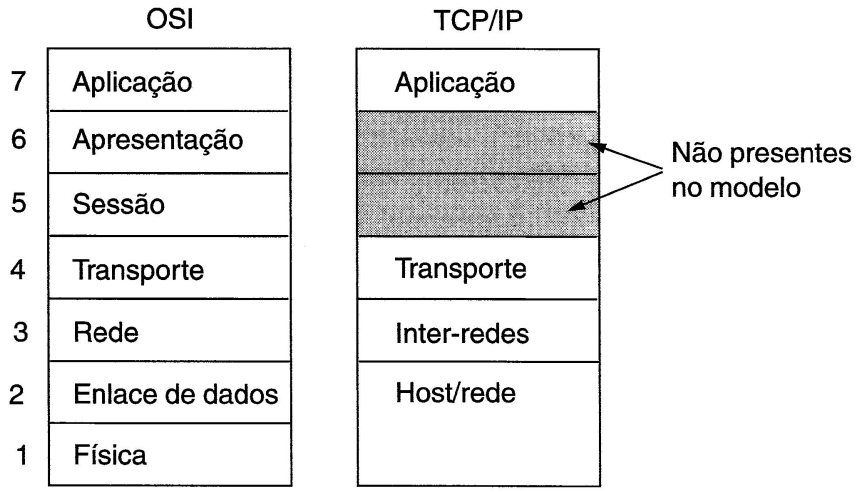

Figura 3.2: Comparaçao entre as pilhas de protocolos OSI e TCP/IP.

Não demorou para que o LDAP se tornasse um protocolo independente do X.500 e se popularizasse no meio corporativo. Inclusive, as mais recentes implementações de protocolos para o X.500 sofreram influência do LDAP e a maioria delas passou a se basear na arquitetura TCP/IP. Mais sobre o X.500 pode ser visto em (CHADWICK, 1996).

[4]Daí a nomenclatura *Lightweight*.
[5]Fonte: (TANENBAUM, 2003)

O LDAP foi originalmente desenvolvido por Tim Howes, Steve Kille e Wengyik Yeong em 1993. Em 1997, já com a ajuda da IETF (*Internet Engineering Task Force*) foi publicada a versão atual LDAPv3 que, entre outras funções adicionou a possibilidade de extensão e o suporte à SASL (*Simple Authentication and Security Layer*). É essa a versão do LDAP que será utilizada nesse estudo.

3.2.2 Visão Geral do LDAP

Entre todas os conceitos apresentado até agora, é bastante importante lembrar e destacar que o LDAP é um protocolo, um conjunto de mensagens usado para acessar dados. O protocolo não define onde os dados serão armazenados, cabendo ao administrador da rede decidir sobre qual forma de armazenamento será utilizada. Essa forma (chamada *backend*) pode variar de arquivos texto a bancos de dados relacionais extremamente complexos e escaláveis.

Por ser um protocolo "leve", o LDAP não possui mensagens para utilizar todos os recursos de um banco de dados complexo, mas também não requer que tais recursos estejam disponíveis. As seguintes operações são aceitas pelo LDAP:

- *start*_TLS - utiliza o protocolo TLS(*Transport Layer Security*) para garantir uma conexão segura.
- *bind* - estabelece a conexão, autentica o cliente e especifica a versão do LDAP.
- *search* - realiza uma busca por entradas no diretório.
- *compare* - testa se uma entrada possui determinado atributo.
- *add* - adiciona uma nova entrada.
- *delete* - remove uma entrada.
- *modify* - altera uma entrada.
- *modify_DN* - renomeia ou move uma entrada.
- *abandon* - aborta uma operação, mas não envia resposta.
- *unbind* - abandona qualquer operação pendente e fecha a conexão.

- *extended operations* - é uma operação genérica que permite criar novas operações. Podemos citar como exemplo *Password Modify* para alteração de senha e *Cancel* que funciona de maneira semelhante à *Abandon*, mas retorna mensagem de erro ou sucesso.

Independente do *backend* utilizado, o LDAP sempre apresentará uma visão organizada em árvore, ficando assim o banco de dados completamente transparente para o cliente. Isso acontece porque o protocolo segue padrões definidos de maneira rigorosa, permitindo assim a interoperabilidade entre clientes e servidores usando implementações de diferentes fornecedores. A Figura 3.3[6] ilustra essa relação de transparência.

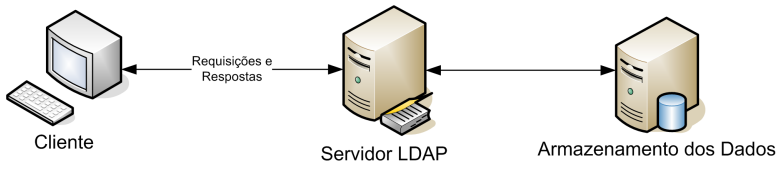

Figura 3.3: Relação entre cliente e servidor LDAP e a base de dados.

Embora seja considerado um protocolo leve, o LDAP possui diversas peculiaridades e elementos únicos, as seções seguintes apresentam uma visão geral sobre alguns deles. Informações mais aprofundadas sobre o LDAP podem ser obtidas nas RFCs 2251 a 2256, 2829, 2830 e 3377, entre outras.

3.2.2.1 *Schemas*

Schemas ou esquemas são os responsáveis por definir um conjunto de regras que determina quais tipos de dados poderão ser armazenados em um determinado diretório. Cada item adicionado ou alterado é comparado com um *schema* para validação. Caso um item não corresponda às regras definidas no *schema* ocorre uma *schema violation* (violação de esquema). Novos *schemas* podem ser adicionados e até mesmo criados pelo administrador do sistema.

[6]Fonte: (MACHADO; JUNIOR, 2006)

3.2.2.2 Arquivos LDIF

O LDAP *Interchange Format*, definido na RFC 2849[7], é um formato padrão de arquivo texto usado para armazenar configurações e itens do diretório. Arquivos LDIF são geralmente utilizados para inserir ou modificar informações no diretório. Os dados contidos em um arquivo LDIF devem estar de acordo com as regras definidas pelos *schemas* em uso. A Figura 3.4 exemplifica um arquivo LDIF que contém o topo da árvore que possui o DN dc=meudominio,dc=org.

```
# Arquivo LDIF para a entrada dn: dc=meudominio,dc=org
dn: dc=meudominio,dc=org
objectClass: domain
dc: meudominio
```

Figura 3.4: Arquivo LDIF

3.2.2.3 Atributos

Atributos são, de várias maneiras, similares a variáveis usadas em programação. Ambos armazenam valores e possuem tipos bem definidos. Essa definição é muito importante para que não haja alocação de dados incompatíveis. Não faz sentido inserir uma `string` em uma variável do tipo `integer`, da mesma forma que não é possível comparar se 2 é menor do que J. A sintaxe de um atributo do LDAP funciona de maneira parecida.

Ao contrário das variáveis, no entanto, os atributos podem armazenar mais de uma informação. Quando um novo valor é inserido em um atributo que já possui outro previamente alocado, ele é adicionado à lista de valores do atributo. Isso é útil, por exemplo, para armazenar números de telefone de usuários, já que atualmente é cada vez mais comum possuirmos dois ou mais números de contato.

Embora seja possível haver atributos com mais de um valor, alguns deles são únicos e só podem armazenar um valor de cada vez. Um exemplo de atributo único é o *uidNumber*, que identifica o ID numérico de um usuário `UNIX`.

Voltando à Figura 3.4, observamos que atributos são listados à esquerda do sinal ":"e seus valores ficam à direita do sinal, separados por um espaço.

[7] http://datatracker.ietf.org/doc/rfc2849/

Existem diversos atributos pré-definidos no LDAP, e muitos outros podem ser adicionados por *schemas*, mas é possível também criar novos atributos através de identificadores de objetos (OIDs)[8].

3.3 OpenLDAP

O LDAP é uma ideia conceitual, um modelo. Para que ele seja colocado em uso, é necessário que haja uma implementação que use o protocolo para resolver problemas reais em uma rede. Nessa seção será apresentado o OpenLDAP, uma implementação de código livre bastante popular do LDAPv3.

O projeto OpenLDAP é uma continuação do primeiro servidor LDAP desenvolvido na Universidade de *Michigan* e foi iniciado em 1998. Ele é distribuído sob a licença OpenLDAP *Public License*[9] e é um sistema multiplataforma, podendo ser executado em ambientes `Linux`, `Windows`, `Solaris` e `MAC OS`, entre outros.

Existem diversas outras implementações comerciais do LDAP, como por exemplo o *Active Directory* da *Microsoft*, no entanto o OpenLDAP será usado neste trabalho, já que ele é um projeto robusto e estável, além de possuir o código aberto e ser gratuito.

Vários serviços que vão além do escopo deste trabalho podem ser integrados ao OpenLDAP. Muitos deles são mostrados em (SUNGAILA, 2008) e em (TRIGO, 2007).

3.4 Comentários Finais

Este capítulo apresentou o sistema de diretórios como alternativa aos bancos de dados relacionais. Foram comparadas suas principais características e se conclui que para o objetivo desse estudo, um servidor de diretórios será mais eficiente.

Além de uma visão geral sobre a organização em árvore dos servidores, foram apresentados o protocolo LDAP e sua implementação OpenLDAP.

O próximo capítulo apresentará argumentos que justificam a integração de um sistema Kerberos com *backend* LDAP. Serão mostradas também as principais configurações necessárias ao funcionamento dessa integração.

[8] Mais sobre a criação de atributos pode ser visto em (CARTER, 2003) e em http://www.iana.org.
[9] http://www.openldap.org/software/release/license.html

Embora sejam apresentados configurações e comandos específicos do LDAP e do Kerberos, eles não serão extensamente analisados, uma vez que o foco desse trabalho é demonstrar as motivações e a viabilidade da integração.

Capítulo 4

Integração

4.1 Por que Integrar?

No mundo da segurança da informação, existem dois termos similares que são comumente confundidos, mas que são essencialmente diferentes e devem ser observados como tal. Esses termos são **autenticação** e **autorização**.

Autenticação é a identificação de algo ou alguém. Ela é necessária para que um usuário, por exemplo, consiga provar que ele é quem diz ser. A autenticação é normalmente realizada através de um segredo conhecido apenas pelo usuário, como uma senha ou uma chave. Por outro lado, a autorização está relacionada ao ato de conceder acesso a determinados recursos. Uma vez que um sistema verifica que um usuário é quem diz ser, é necessário que ele descubra o que esse usuário está autorizado a fazer. A autorização geralmente está ligada ao uso de grupos e permissões.

Como já foi apresentado, o protocolo Kerberos é uma ferramenta de autenticação que provê segurança a redes inseguras autenticando não só usuários, mas também os serviços que serão utilizados por eles. No entanto, o protocolo só se preocupa em garantir a identidade das partes envolvidas em uma transação. Uma vez que determinado usuário autentique-se, é necessário limitar a autorização que ele possui. Não é desejável que todos os usuários possuam acesso de administrador, por exemplo. Para definir quais são os limites de autorização que determinado usuário ou serviço possui, é necessário obter informações sobre ele. A rapidez na leitura de dados e a organização hierárquica tornam o LDAP uma ferramenta muito útil nesse processo.

Individualmente, os protocolos Kerberos e LDAP possuem um objetivo similar, a centralização. Seja ela de dados ou de autenticação, em sua essência ambos buscam diminuir a redundância de informações em uma rede. No entanto, como já foi apresentado neste trabalho, eles são ferramentas com propósitos muito distintos e que realizam tarefas complementares entre si. Embora seja a princípio um pouco complexa, a integração entre os dois serviços é viável e muito produtiva. Ela tanto facilita o trabalho do administrador de rede quanto provê segurança e comodidade ao usuário.

4.2 Pré-Requisitos

Para que se obtenha o perfeito funcionamento das implementações dos serviços LDAP e Kerberos, é necessário que certos requisitos sejam atendidos. As configurações realizadas nesse trabalho assumem que a rede não possui nenhum serviço previamente instalado. Em redes já em produção, certos requisitos já poderão estar satisfeitos, mas é provável que alterações sejam necessárias.

A organização do Kerberos e do LDAP torna necessária a configuração de um domínio na rede. Todas as máquinas e serviços da rede deverão possuir FQDNs(*Fully Qualified Domain Names*) únicos e definidos. A forma mais fácil de fazer isso é através da configuração de um servidor DNS. Tal configuração nem sempre é trivial, dessa forma foge do escopo deste trabalho a apresentação da mesma. Mais detalhes sobre a configuração de servidores DNS podem ser obtidos em (MORIMOTO, 2008) e em (UCHÔA; SICA; SIMEONE, 2004).

Outro requisito importante é a sincronização do relógio das máquinas da rede. Como os *tickets* emitidos pelo servidor kdc expiram com o tempo, é necessário que o horário esteja configurado da mesma forma em todos os *hosts*. Caso haja um servidor de hora na rede, ele pode ser usado, caso contrário basta usar o `ntp`[1]. Ele mantém os relógios sincronizados e deve ser instalado em todos os *hosts* da rede. O ntp pode ser instalado através da ferramenta `apt-get`:

```
$ sudo apt-get install ntp
```

Esses são os requisitos iniciais, caso não sejam satisfeitos, erros podem ser encontrados durante ou após a configuração dos serviços. Qualquer requisito adicional será apresentado nas seções subsequentes, durante a implementação.

[1] *Network Time Protocol* - http://support.ntp.org/

4.3 Instalando e configurando o servidor LDAP

Esta seção apresenta a configuração completa do LDAP em um servidor dedicado para executar a aplicação e armazenar os dados. A configuração do servidor LDAP será feita de forma a deixá-lo pronto para armazenar os dados dos *principals* do Kerberos.

4.3.1 Instalando e Populando a Base

O primeiro passo para a configuração do serviço de diretórios LDAP é a instalação dos pacotes `slapd` e `ldap-utils`, que pode ser feita através do `apt-get`. Alguns *schemas* básicos já vem carregados por padrão, mas será preciso carregar outros. A Figura 4.1 mostra a instalação dos pacotes e a adição dos *schemas*. A Tabela 4.1 explica alguns parâmetros dos comandos **ldapadd** e **ldapmodify** usados neste ou em outros exemplos subsequentes.

```
$ sudo apt-get install slapd ldap-utils

$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Figura 4.1: Instalando Pacotes e adicionando *schemas*

Tabela 4.1: Parâmetros usados no comando ldapadd

Valor	*Significado*
$-Y$	Força o uso da autenticação padrão da SASL(EXTERNAL).
$-H$	Especifica URI(s)[1] relacionados ao servidor LDAP.
$-f$	Indica o arquivo .ldif que será adicionado.
$-x$	Usa autenticação simples ao invés de SASL.
$-D$	Usa o DN especificado para autenticação no diretório.
$-W$	Solicita autenticação ao invés de receber a senha diretamente pela linha de comando.

[1] Uniform Resource Identifier - http://tools.ietf.org/html/rfc3986

O OpenLDAP usa um diretório **cn=config** que é o responsável por configurar dinamicamente o *daemon* `slapd`. Esse diretório *backend* possui apenas uma configuração mínima. Para que o diretório *frontend* seja populado, serão necessárias configurações adicionais. A princípio essas configurações permitirão o armazenamento de usuários capazes de se autenticar nos mais diversos serviços.

Para popular o *backend*, será criado um arquivo **backend.meudominio.org.ldif**, conforme a Figura 4.2. Esse arquivo deve ser adicionado ao diretório usando-se o comando mostrado na Figura 4.3.

```
# Carregar modulos dinâmicos do backend
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Configurações do Banco de dados
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=meudominio,dc=org
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=meudominio,dc=org
olcRootPW: senha
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=meudominio,dc=org"
write by anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=meudominio,dc=org" write by * read
```

Figura 4.2: Arquivo backend.meudominio.org.ldif

```
$ sudo ldapadd -Y EXTERNAL -H ldapi:/// -f backend.meudominio.org.ldif

SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module,cn=config"

adding new entry "olcDatabase=hdb,cn=config"
```

Figura 4.3: Adicionando Configurações ao *Backend*

Após essas configurações, o diretório está pronto para ser populado. Para tanto, deve-se usar um outro arquivo ldif, que será chamado **frontend.meudominio.org.ldif**, conforme a Figura 4.4 e a Figura 4.5. Serão criados o topo da árvore do diretório, um usuário administrador, dois nós "usuarios"e "grupos", um objeto pertencente ao nó "usuarios"e um objeto pertencente ao nó "grupos".

```
# Cria o objeto que será o topo da árvore do domínio
dn: dc=meudominio,dc=org
objectClass: top
objectClass: dcObject
objectclass: organization
o: Dominio de exemplo
dc: meudominio
description: Meu Dominio

# Cria o usuário admin.
dn: cn=admin,dc=meudominio,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: Administrador do LDAP
userPassword: senha

# Cria o nó usuarios
dn: ou=usuarios,dc=meudominio,dc=org
objectClass: organizationalUnit
ou: usuarios

# Cria o nó grupos
dn: ou=grupos,dc=meudominio,dc=org
objectClass: organizationalUnit
ou: grupos
```

Figura 4.4: Primeira parte do arquivo frontend.meudominio.org

```
# Cria o objeto fulano abaixo do nó usuarios
dn: uid=fulano,ou=usuarios,dc=meudominio,dc=org
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: fulano
sn: de Tal
givenName: Fulano
cn: Fulano de Tal
displayName: Fulano de Tal
uidNumber: 1000
gidNumber: 10000
userPassword: senha
gecos: Fulano de Tal
loginShell: /bin/bash
homeDirectory: /home/fulano
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: fulano.tal@meudominio.org
postalCode: 36200000
l: Lavras
o: Exemplo
mobile: +55 (35)1234 5678
homePhone: +55 (35)5678 1234
title: Administrador do Sistema
postalAddress:
initials: FT

# Cria o objeto ti abaixo do nó grupos
dn: cn=ti,ou=grupos,dc=meudominio,dc=org
objectClass: posixGroup
cn: ti
gidNumber: 10000
```

Figura 4.5: Segunda parte do arquivo frontend.meudominio.org

Esse arquivo também deve ser adicionado ao diretório. No entanto, ao contrário do arquivo **backend.meudominio.org.ldif**, que é inserido diretamente no diretório **cn=config**, ele servirá para popular a base principal do domínio, e portanto deve ser inserido na raiz do diretório. Essa inserção é mostrada na Figura 4.6.

```
$ sudo ldapadd -x -D cn=admin,dc=meudominio,dc=org -W -f frontend.meudominio.org.ldif

adding new entry "dc=meudominio,dc=org"

adding new entry "cn=admin,dc=meudominio,dc=org"

adding new entry "ou=usuarios,dc=meudominio,dc=org"

adding new entry "ou=grupos,dc=meudominio,dc=org"

adding new entry "uid=fulano,ou=usuarios,dc=meudominio,dc=org"

adding new entry "cn=ti,ou=grupos,dc=meudominio,dc=org"
```

Figura 4.6: Adicionando o arquivo frontend.meudominio.org.ldif

Estando o diretório populado, é possível fazer uma busca para observar se as informações foram adicionadas corretamente. A Figura 4.7 exemplifica a procura pelo usuário "fulano"e a Tabela 4.2 apresenta alguns parâmetros do comando **ldapsearch** usados neste ou em outros exemplos ao longo do trabalho.

```
$ ldapsearch -xLLL -b "dc=meudominio,dc=org" uid=fulano sn givenName cn
dn: uid=fulano,ou=usuarios,dc=meudominio,dc=org
sn: de Tal
givenName: Fulano
cn: Fulano de Tal
```

Figura 4.7: Busca no diretório.

Tabela 4.2: Parâmetros usados no comando ldapsearch

Valor	*Significado*
$-L(L)(L)$	Limita o nível de detalhamento da saída do comando.
$-b$	Usa o nó especificado como ponto inicial da busca.
$-Z(Z)$	Inicia a operação *StartTLS*, se forem usados dois "Z", o sucesso da operação será obrigatório.

4.3.2 Habilitando o Suporte à Criptografia

O Kerberos trafega *tickets* entre seus *principals* de forma criptografada. Isso garante a segurança em uma rede insegura. No entanto, essa segurança pode ser comprometida se não houver criptografia no tráfego de dados entre o servidor kdc e o servidor que contém base de dados LDAP. Para garantir que essa comunicação seja criptografada é necessário habilitar, no servidor LDAP, o suporte à TLS[2] (*Transport Layer Security*). Para isso, é preciso criar uma chave e um certificado para o servidor. Ambos devem ser então assinados por uma autoridade certificadora (*Certificate Authority - CA*) oficialmente reconhecida. Nesse trabalho serão usados certificados auto-assinados. A ferramenta `certtool` será utilizada para criar os certificados. Ela está contida no pacote `gnutls-bin` e pode ser instalada conforme mostra a Figura 4.8.

```
$ sudo apt-get install gnutls-bin
```

Figura 4.8: Instalando o pacote gnutls-bin

Todas as chaves e certificados serão armazenados no diretório /etc/ldap/ssl. Isso não é obrigatório, mas independente do local de instalação é necessário que as permissões sejam ajustadas para que o grupo openldap possa ler as chaves. É importante lembrar que apenas o usuário root e o grupo openldap devem ter

[2] http://datatracker.ietf.org/wg/tls/charter/

acesso ao diretório, uma vez que o comprometimento desses certificados pode causar problemas graves de segurança.

Antes de criar a chave e o certificado do servidor, é necessário criar uma chave privada e um certificado auto-assinado para a CA. A Figura 4.9 mostra como a chave privada da CA pode ser gerada.

```
$ sudo sh -c "certtool --generate-privkey > /etc/ldap/ssl/cakey.pem"
Generating a 2048 bit RSA private key...
```

Figura 4.9: Gerando a Chave Privada da CA

Para gerar o certificado, primeiro será criado um arquivo /etc/ldap/ssl/ca.info contendo informações sobre a CA, conforme a Figura 4.10.

```
cn = Meu Dominio
ca
cert_signing_key
```

Figura 4.10: Conteúdo do arquivo ca.info

A chave e o arquivo serão então usados para gerar o certificado auto-assinado. A Figura 4.11 mostra o comando usado e a saída obtida.

```
$ sudo certtool --generate-self-signed --load-privkey \
/etc/ldap/ssl/cakey.pem --template /etc/ldap/ssl/ca.info \
--outfile /etc/ldap/ssl/cacert.pem
Generating a self signed certificate...
X.509 Certificate Information:
        Version: 3
        Serial Number (hex): 4d693f60
        Validity:
                Not Before: Sat Feb 26 17:58:56 UTC 2011
                Not After: Sun Feb 26 17:58:56 UTC 2012
        Subject: CN=Meu Dominio
        Subject Public Key Algorithm: RSA
                Modulus (bits 2048):
                        bc:17:14:36:4b:c7:75:70:23:cd:55:a6:ce:30:f7:d4
                        c4:38:26:59:e1:66:27:9b:c2:dd:b9:f7:ec:fe:26:48
                        38:60:d8:23:8d:d2:6e:da:18:91:a1:70:41:ee:75:f2
                        21:05:6e:f2:8c:78:1a:dc:fd:fc:35:1e:a4:33:29:d1
                        f9:c1:fc:51:6d:f0:6a:c4:95:d5:83:9b:b7:3e:8a:56
                        70:78:03:8d:1a:e3:f8:a5:53:f6:20:96:0e:75:a9:fc
                        fc:9e:87:55:8b:7f:ba:14:74:7f:3f:eb:b8:ab:92:62
                        9c:d0:96:f0:cb:40:15:46:86:f7:e2:ce:4e:47:4e:8c
                        9c:b5:7a:36:00:45:e0:9b:87:6f:70:08:8c:1c:af:3a
                        1e:79:e3:39:01:b1:6e:00:c6:e8:e0:7d:ac:31:1e:3f
                        cf:82:87:6c:c4:23:eb:20:78:65:8c:d4:17:7c:51:3d
                        bb:26:07:b0:81:3e:96:16:31:5b:5d:a5:96:13:69:ab
                        6c:77:3c:c0:e3:8f:f2:7b:57:8a:61:ab:ad:fa:05:d6
                        4b:2f:eb:9f:45:1a:d9:e1:70:24:92:3c:90:3c:7c:3b
                        23:b8:fe:e9:c1:aa:00:02:7d:37:94:6d:b2:ae:2c:49
                        a9:89:8e:b3:0f:01:ac:0f:fd:5f:2a:6d:29:ad:4c:85
                Exponent (bits 24):
                        01:00:01
        Extensions:
                Basic Constraints (critical):
                        Certificate Authority (CA): TRUE
                Key Usage (critical):
                        Certificate signing.
                Subject Key Identifier (not critical):
                        71f579b327708417eef1b748a20770938d13052d
Other Information:
        Public Key Id:
                71f579b327708417eef1b748a20770938d13052d

Signing certificate...
```

Figura 4.11: Gerando o certificado auto-assinado

De posse do certificado auto-assinado da CA, é possível assinar um certificado para o servidor LDAP. No entanto, primeiramente é necessário criar também uma chave privada para ele como mostra a Figura 4.12.

```
$ sudo sh -c "certtool --generate-privkey > /etc/ldap/ssl/ldapmaster_slapd_key.pem "
Generating a 2048 bit RSA private key...
```

Figura 4.12: Gerando a Chave Privada do Servidor

Assim como o certificado da CA, o certificado do servidor requer algumas informações que serão passadas através do arquivo /etc/ldap/ssl/ldapmaster.info, mostrado na Figura 4.13.

```
organization = Meu Dominio
cn = ldapmaster.meudominio.org
tls_www_server
encryption_key
signing_key
```

Figura 4.13: Arquivo ldapmaster.info

Finalmente, o certificado do servidor pode ser gerado. A Figura 4.14 e a Figura 4.15 mostram o comando usado e a saída obtida.

```
$ sudo certtool --generate-certificate --load-privkey \
/etc/ldap/ssl/ldapmaster_slapd_key.pem --load-ca-certificate \
/etc/ldap/ssl/cacert.pem --load-ca-privkey /etc/ldap/ssl/cakey.pem \
 --template /etc/ldap/ssl/ldapmaster.info --outfile \
/etc/ldap/ssl/ldapmaster_slapd_cert.pem
Generating a signed certificate...
X.509 Certificate Information:
        Version: 3
        Serial Number (hex): 4d6940c6
        Validity:
                Not Before: Sat Feb 26 18:04:54 UTC 2011
                Not After: Sun Feb 26 18:04:54 UTC 2012
        Subject: O=Meu Dominio,CN=ldapmaster.meudominio.org
        Subject Public Key Algorithm: RSA
                Modulus (bits 2048):
                        ad:81:7d:b1:ee:4c:12:4c:73:5c:4a:34:ac:05:65:dd
                        de:a4:a7:8a:c2:06:68:82:cd:5b:52:42:57:3d:8b:62
                        88:15:44:1e:a2:cc:d4:e3:e7:4c:93:9b:44:c2:2b:ce
                        dc:e9:8c:18:b5:d0:b7:d3:57:9a:ea:4c:79:53:65:23
                        35:75:43:f2:a5:ff:00:40:b7:3d:44:7e:27:48:8f:71
                        33:68:44:2c:c6:a6:85:0e:41:2d:35:e3:8f:1b:63:66
                        0c:04:1a:77:9c:33:a3:a4:b6:03:9b:04:03:ee:c0:f1
                        6a:52:6b:46:7a:7e:68:1b:da:2c:8a:fb:cb:7c:e9:fa
                        db:e5:0b:5f:2c:50:05:a6:ec:3b:a9:9d:24:96:d3:0e
                        6d:ca:ee:72:ad:4e:94:f3:83:7a:38:d4:32:2a:60:50
                        a6:c3:14:50:fb:e0:62:d0:96:21:a3:6b:62:64:77:f8
                        68:01:1a:44:59:65:01:13:0c:fc:6f:b5:0f:9b:7b:85
                        31:c0:8c:92:50:64:b5:2d:47:ee:1c:50:6f:8c:48:9e
                        60:ba:d2:bb:5b:49:99:25:22:b2:00:b9:a4:96:7a:2f
                        7f:af:33:90:e9:f4:7a:e6:c6:71:00:82:82:34:80:d2
                        b1:27:d9:70:26:f0:6a:9c:72:04:8b:5f:44:fa:8a:95
                Exponent (bits 24):
                        01:00:01
```

Figura 4.14: Gerando o certificado do Servidor (Primeira Parte)

```
Extensions:
                Basic Constraints (critical):
                        Certificate Authority (CA): FALSE
                Key Purpose (not critical):
                        TLS WWW Server.
                Key Usage (critical):
                        Digital signature.
                        Key encipherment.
                Subject Key Identifier (not critical):
                        0cbd520affb6bada886db4c4c7c32ee43dc90829
                Authority Key Identifier (not critical):
                        71f579b327708417eef1b748a20770938d13052d
Other Information:
        Public Key Id:
                0cbd520affb6bada886db4c4c7c32ee43dc90829

Signing certificate...
```

Figura 4.15: Gerando o certificado do Servidor (Segunda Parte)

Serão necessárias ainda algumas alterações no arquivo **/etc/ldap/ldap.conf**. Tais alterações, mostradas na Figura 4.16, servem para indicar o endereço do servidor LDAP e o uso de criptografia nele, além de informar de que maneira o certificado da CA será verificado. A Tabela 4.3 mostra os valores possíveis para o parâmetro TLS_REQCERT.

```
BASE    dc=meudominio,dc=org
URI     ldap://ldapmaster.meudominio.org ldaps://ldapmaster.meudominio.org
TLS_REQCERT allow
```

Figura 4.16: Modificações no Arquivo /etc/ldap/ldap.conf

Tabela 4.3: Valores possíveis para o parâmetro TLS_REQCERT

Valor	*Significado*
never	Aceita qualquer certificado, sem verificação
allow	Aceita CAs desconhecidas e certificados auto-assinados
try	Não exige certificado, mas caso ele exista, checa todos os seus parâmetros
hard/*demand*	Requer e verifica todos os dados do certificado

O último passo na configuração da TLS é informar, na árvore cn=config, a localização das chaves e dos certificados. Para isso, será usado o comando **ldapmodify** em sua forma interativa, como mostrado na Figura 4.17. Pode ser também utilizado um arquivo .ldif. A senha usada para autenticação é a que foi configurada para o admin no arquivo **backend.meudominio.org.ldif**. Após a execução do comando, devem ser pressionadas as teclas ctrl + d e o serviço slapd deve ser reiniciado.

```
$sudo ldapmodify -Y EXTERNAL -H ldapi:/// -W

Enter LDAP Password:
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap01_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap01_slapd_key.pem

modifying entry "cn=config"
```

Figura 4.17: Usando o Comando ldapmodify

Para testar se o suporte à TLS foi habilitado, pode-se fazer uma busca na base usando o parâmetro -ZZ, que força o uso da instrução `StartTLS`. A Figura 4.18 exemplifica essa busca.

```
$ldapsearch -xLLL -D "cn=admin,dc=meudominio,dc=org" -ZZ -W uid=fulano sn givenName cn
Enter LDAP Password:
dn: uid=fulano,ou=usuarios,dc=meudominio,dc=org
sn: de Tal
givenName: Fulano
cn: Fulano de Tal
```

Figura 4.18: Efetuando uma Busca Usando Conexão Criptografada.

4.3.3 Preparando a Base para a Inserção de *Principals*

Após essa configuração inicial, o servidor LDAP está pronto para ser usado. No entanto, para que ele seja capaz de armazenar chaves de *principals*, alguns ajustes adicionais serão necessários. A instalação padrão do OpenLDAP não possui os *schemas* que contêm os modelos de dados do Kerberos, dessa forma, faz-se necessária a instalação de um *plugin* que possua tais arquivos. Essa ferramenta, no entanto, não adiciona os *schemas*, sendo necessária a extração manual dos mesmos. O processo de instalação e extração é mostrado na Figura 4.19.

```
$sudo apt-get install krb5-kdc-ldap

$sudo gzip -d /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz
$sudo cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema /etc/ldap/schema/
```

Figura 4.19: Instalando *plugin* e Extraindo *schemas*.

Todos os *schemas* usados em um sistema LDAP devem ser adicionados ao diretório cn=config. Para isso, é necessário que eles estejam em formato de arquivo ldif. Os *schemas* que estão incluídos na instalação padrão do OpenLDAP já possuem arquivos nesse formato, no entanto o *schema* do Kerberos precisa ser convertido. Isso pode ser feito usando-se a ferramenta `slapcat`. Para tal, será necessário um arquivo temporário, mostrado na Figura 4.20 que será usado na conversão.

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/kerberos.schema
```

Figura 4.20: Arquivo schema.conf.

Apenas por uma questão de organização, será criado um diretório temporário chamado tmp para armazenar os arquivos ldif. O comando para gerar o arquivo é mostrado na Figura 4.21.

```
$slapcat -f schema_convert.conf -F tmp -n0 -s \
"cn={12}kerberos,cn=schema,cn=config" > tmp/cn=kerberos.ldif
```

Figura 4.21: Convertendo *schema* para Arquivo ldif

Após gerado, o arquivo cn=kerberos.ldif deve ser alterado conforme apresentado na Figura 4.22 e adicionado ao diretório, conforme mostra a Figura 4.23.

```
dn: cn=kerberos,cn=schema,cn=config
...
cn: kerberos

# Estas linhas, localizadas no final do arquivo, devem ser removidas.
structuralObjectClass: olcSchemaConfig
entryUUID: 18ccd010-746b-102d-9fbe-3760cca765dc
creatorsName: cn=config
createTimestamp: 20090111203515Z
entryCSN: 20090111203515.326445Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20090111203515Z
```

Figura 4.22: Alterações no Arquivo kerberos.ldif

```
$sudo ldapadd -Y EXTERNAL -H ldapi:/// -f tmp/cn\=kerberos.ldif
```

Figura 4.23: Adicionando o Arquivo kerberos.ldif

O servidor OpenLDAP já está pronto para armazenar os dados dos *principals* do Kerberos, no entanto ainda são recomendáveis alguns ajustes adicionais. Primeiramente é possível adicionar o nome do *principal* aos índices de busca do diretório, isso permite que sejam feitas buscas utilizando esse parâmetro. A Figura 4.24 mostra essa configuração. O LDAP possui algumas formas características de comparação que podem ser usadas na pesquisa. Essas formas são mostradas na Tabela 4.4.

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -W
Enter LDAP Password:
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
dn: olcDatabase={1}hdb,cn=config
add: olcDBIndex
olcDBIndex: krbPrincipalName eq,pres,sub

modifying entry "olcDatabase={1}hdb,cn=config"
```

Figura 4.24: Adicionando Parâmetro de Indexação

Tabela 4.4: Tipos de pesquisas possíveis em um diretório

Tipo	*Significado*
eq	Compara o atributo como um todo
sub	Compara trechos do atributo
pres	Compara o atributo pela sua presença
approx	Compara o atributo por aproximação
none	Não faz nenhuma comparação com o atributo

Outro ajuste importante é a configuração da lista de controle de acesso ou ACL (*Access Control List*), para limitar o acesso a informações confidenciais, como as chaves dos *principals*. Isso também é feito usando o comando **ldapmodify**, e é mostrado na Figura 4.25.

Terminados os ajustes, o servidor LDAP está pronto para armazenar as informações do Kerberos. Restando ainda o processo de configuração do servidor kdc. Tal processo será explicado na próxima seção.

```
$ sudo ldapmodify -Y EXTERNAL -H ldapi:/// -W
Enter LDAP Password:
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
dn: olcDatabase={1}hdb,cn=config
replace: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange,krbPrincipalKey
by dn="cn=admin,dc=meudominio,dc=org" write
by anonymous auth
by self write
by * none
-
add: olcAccess
olcAccess: to dn.base="" by * read
-
add: olcAccess
olcAccess: to * by dn="cn=admin,dc=meudominio,dc=org" write by * read

modifying entry "olcDatabase={1}hdb,cn=config"
```

Figura 4.25: Alterando a ACL

4.4 Instalando e Configurando o Servidor Kerberos

Com o servidor LDAP pronto, é possível configurar o kdc, que será responsável por emitir os *tickets* e autenticar os usuários. Exceto quando especificado, toda a configuração feita nessa seção se aplica ao servidor do Kerberos.

Primeiramente, deve-se instalar os pacotes do Kerberos e *plugins* necessários para sua integração com o LDAP. A Figura 4.26 mostra essa instalação.

```
$sudo apt-get install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

Figura 4.26: Instalando Pacotes no Servidor Kerberos

Durante o processo de instalação, algumas informações serão solicitadas. Não há necessidade de informá-las, uma vez que serão inseridas posteriormente direto no arquivo **/etc/krb5.conf**.

Neste arquivo, pode-se observar que é possível configurar mais de um reino Kerberos. No entanto, no exemplo mostrado na Figura 4.27, será configurado apenas o reino padrão, que corresponde ao domínio e também ao topo da estrutura organizacional do LDAP. É necessário informar o endereço do servidor kdc e do servidor de administração, sejam eles separados ou não. O DN da raiz do diretório e do administrador da base também são informados. Por fim, são inseridos o endereço do servidor LDAP e o arquivo que contém a chave para autenticação no servidor.

```
[libdefaults]
        default_realm = MEUDOMINIO.ORG

...

[realms]
        MEUDOMINIO.ORG = {
                kdc = kdc.meudominio.org
                admin_server = kdc.meudominio.org
                default_domain = meudominio.org
                database_module = meudominio_ldapconf
        }

...

[domain_realm]
        .meudominio.org = MEUDOMINIO.ORG
                meudominio.org = MEUDOMINIO.ORG

...

[dbdefaults]
        ldap_kerberos_container_dn = dc=meudominio,dc=org

[dbmodules]
        meudominio_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=meudominio,dc=org"
                ldap_kadmind_dn = "cn=admin,dc=meudominio,dc=org"
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps://ldapmaster.meudominio.org
                ldap_conns_per_server = 5
        }
```

Figura 4.27: Alterações no Arquivo /etc/krb5.conf

Devem ser criados o reino e um arquivo que armazenará a senha usada na conexão com o servidor LDAP. Ambos os serão feitos usando a ferramenta **kdb5_ldap_util**, como mostrado na Figura 4.28.

```
$ sudo kdb5_ldap_util -D cn=admin,dc=meudominio,dc=org \
create -subtrees dc=meudominio,dc=org -r MEUDOMINIO.ORG \
-s -H ldap://ldapmaster.meudominio.org
Password for "cn=admin,dc=meudominio,dc=org":
Initializing database for realm 'MEUDOMINIO.ORG'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:

$ sudo kdb5_ldap_util -D cn=admin,dc=meudominio,dc=org \
stashsrvpw -f /etc/krb5kdc/service.keyfile \
cn=admin,dc=meudominio,dc=org
Password for "cn=admin,dc=meudominio,dc=org":
Re-enter password for "cn=admin,dc=meudominio,dc=org":
```

Figura 4.28: Criando reino Kerberos e Senha para Conexão com o LDAP

Para que haja conexão criptografada entre o kdc e o diretório, é necessário que o servidor Kerberos possua e reconheça o certificado auto-assinado da CA. Para tanto, o arquivo **cacert.pem**, gerado no servidor LDAP deve ser copiado para algum diretório local, no exemplo /etc/ssl/certs. Além disso, no próprio kdc, o arquivo **/etc/ldap/ldap.conf** deve ser alterado conforme mostrado na Figura 4.29.

```
TLS_CACERT /etc/ssl/certs/cacert.pem
```

Figura 4.29: Alterando o Arquivo /etc/ldap/ldap.conf no Servidor Kerberos

O servidor Kerberos já está pronto para se conectar ao LDAP e armazenar os usuários. A Figura 4.30 mostra um usuário sendo adicionado ao diretório. A adição com sucesso pode ser conferida no servidor LDAP, entre outras maneiras, usando-se o comando **slapcat**.

```
$sudo kadmin.local -q "addprinc beltrano@MEUDOMINIO.ORG"
Authenticating as principal root/admin@MEUDOMINIO.ORG with password.
WARNING: no policy specified for beltrano@MEUDOMINIO.ORG; defaulting to no policy
Enter password for principal "beltrano@MEUDOMINIO.ORG":
Re-enter password for principal "beltrano@MEUDOMINIO.ORG":
Principal "beltrano@MEUDOMINIO.ORG" created.
```

Figura 4.30: Adicionando um Usuário no Diretório

Com um usuário já cadastrado, é possível obter *tickets*, usando o comando **kinit**, e verificar sua validade com o comando **klist**, ambos mostrados na Figura 4.31

```
$ kinit beltrano
Password for beltrano@MEUDOMINIO.ORG:
$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: beltrano@MEUDOMINIO.ORG

Valid starting     Expires            Service principal
03/10/11 14:05:40  03/11/11 00:05:40  krbtgt/MEUDOMINIO.ORG@MEUDOMINIO.ORG
        renew until 03/11/11 14:05:38
```

Figura 4.31: Solicitando e Verificando um *ticket*

A configuração dos servidores Kerberos e LDAP está pronta e sua integração, funcional. O próximo passo é autenticar os usuários no sistema e explorar as diversas possibilidades de uso dessa integração.

4.5 Comentários Finais

O objetivo desse capítulo era defender a viabilidade, justificar e demonstrar a integração dos protocolos Kerberos e LDAP. Foram também apresentados e brevemente explicados os seus requisitos.

A maior parte do capítulo foi dedicada às configurações necessárias à integração. No entanto todos os comandos e arquivos usados foram abordados de maneira superficial. Informações mais detalhadas sobre eles podem ser vistas em (GARMAN, 2003), (SUNGAILA, 2008), (CARTER, 2003), e (TRIGO, 2007).

O próximo capítulo apresenta, como resultados, possibilidades de aplicações práticas de um sistema Kerberos com *backend* LDAP.

Capítulo 5

Exemplos de Aplicação

A integração do protocolo Kerberos com *backend* LDAP embora funcional, por si só, não apresenta resultados mensuráveis. A implementação serve apenas de base para o uso de outros serviços.

Neste capítulo serão apresentados exemplos de uso que mostram a autenticação de usuários em um sistema UNIX e em um sistema de compartilhamento de arquivos.

O Kerberos adiciona segurança no acesso aos serviços, enquanto o LDAP centraliza os dados dos usuários. Dessa forma, um usuário cadastrado uma única vez pode acessar quaisquer serviços que estejam integrados a essa solução.

5.1 Autenticando Usuários

Usar o servidor Kerberos para autenticar usuários em um sistema Linux é muito simples. Primeiramente é necessário instalar o pacote **libpam-krb5**, conforme mostra a Figura 5.1. Todas as configurações feitas nesta seção, exceto quando exposto o contrário, serão feitas no cliente.

```
$ sudo apt-get install libpam-krb5
```

Figura 5.1: Instalando Pacote libpam-krb5

A seguir deve ser feita a configuração no arquivo **/etc/krb5.conf** para identificar o *realm*, o domínio e o endereço do servidor kdc. As alterações, mostradas na Figura 5.2, são similares àquelas realizadas no servidor Kerberos no capítulo anterior.

```
[libdefaults]
        default_realm = MEUDOMINIO.ORG

[realms]
         MEUDOMINIO.ORG = {
             kdc = kdc.meudominio.org
             admin_server = kdc.meudominio.org
             master_kdc = kdc.meudominio.org
             default_domain = meudominio.org
         }
```

Figura 5.2: Alterações no Arquivo /etc/krb5.conf

As configurações necessárias para autenticação são adicionadas automaticamente ao PAM (*Pluggable Authentication Modules*) e o servidor kdc estará pronto para autenticar usuários. Para isso, no entanto, é necessário que os mesmos existam na base LDAP e no arquivo **/etc/passwd** do cliente. Mais sobre esse arquivo e sobre o PAM pode ser visto em (UCHÔA, 2007).

A Figura 5.3 exemplifica a adição de um usuário à base no kdc, feita diretamente no servidor Kerberos, e a Figura 5.4 mostra a adição do usuário no cliente.

```
$ sudo kadmin.local -q "addprinc beltrano@MEUDOMINIO.ORG"
Authenticating as principal root/admin@MEUDOMINIO.ORG with password.
WARNING: no policy specified for beltrano@MEUDOMINIO.ORG; defaulting to no policy
Enter password for principal "beltrano@MEUDOMINIO.ORG":
Re-enter password for principal "beltrano@MEUDOMINIO.ORG":
Principal "beltrano@MEUDOMINIO.ORG" created.
```

Figura 5.3: Adicionando um *Principal*

```
$ sudo adduser beltrano
Adicionando o usuário 'beltrano' ...
Adicionando novo grupo 'beltrano' (1001) ...
Adicionando novo usuário 'beltrano' (1001) ao grupo 'beltrano' ...
Criando diretório pessoal '/home/beltrano' ...
Copiando arquivos de '/etc/skel' ...
Current Kerberos password:
Enter new Kerberos password:
Retype new Kerberos password:
passwd: password updated successfully
Changing the user information for beltrano
Enter the new value, or press ENTER for the default
        Full Name []: Beltrano
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Esta informação está correta?[S/n] S
```

Figura 5.4: Adicionando um Usuário para Autenticação no Cliente

5.2 Compartilhando Arquivos Usando NFSv4

O NFS[1] (*Network File System*) é um protocolo que permite o compartilhamento de arquivos em uma rede. Através dele pode-se configurar um servidor que conterá um ou mais diretórios que poderão ser acessados por outras máquinas da rede, mediante autenticação. A seguir será demonstrado como configurar o NFS de modo a usar o Kerberos para autenticar os clientes.

5.2.1 Configurações no Servidor

Neste exemplo, o compartilhamento será feito a partir do próprio servidor kdc. O diretório compartilhado será o **/home/compartilhado**. É recomendado que se utilize uma estrutura separada de diretórios para o NFS. Por isso os diretórios compartilhados serão montados abaixo do diretório /export, dessa forma será necessário criar a estrutura de diretórios **/export/home/compartilhado**.

É preciso instalar os pacotes `nfs-kernel-server` e `nfs-common`, conforme mostra a Figura 5.5.

```
$ apt-get install nfs-kernel-server nfs-common
```

Figura 5.5: Instalando Pacotes Necessários para o Funcionamento do NFS

No arquivo **/etc/fstab** deverá ser informado que o diretório /home será montado em /export/home. A Figura 5.6 mostra a linha que deverá ser adicionada ao arquivo. Feita essa alteração, o diretório deve ser montado usando o comando `mount` ou o sistema deve ser reiniciado para que as configurações do fstab sejam utilizadas.

```
/home    /export/home   none    bind  0  0
```

Figura 5.6: Arquivo /etc/fstab

A seguir, o arquivo **/etc/exports** deve ser configurado para compartilhar os diretórios com todos os clientes que estejam usando Kerberos. A Figura 5.7 mostra as configurações feitas no arquivo, e a Tabela 5.1 explica os parâmetros utilizados.

[1] http://www.nfsv4.org/

```
/export         gss/krb5(rw,fsid=0,async,subtree_check,no_root_squash,crossmnt)
/export/home    gss/krb5(rw,async,subtree_check,no_root_squash,crossmnt)
/export/home/compartilhado        gss/krb5(rw,async,subtree_check,root_squash,crossmnt)
```

Figura 5.7: Arquivo /etc/exports

Tabela 5.1: Opções usadas no arquivo /etc/fstab

Valor	*Significado*
rw	Monta o sistema de arquivos com permissões de leitura e escrita
$fsid = 0$	Indica qual será o diretório *root* da exportação
async	As opereções de leitura e escrita no sistema deverão ser feitas de maneira assíncrona
subtree_check	Previne que um usuário acesse diretórios do sistema que não estão compartilhados
no_root_squash	Não permite que o usuário root acesse o sistema de arquivos
root_squash	Permite que o usuário root acesse o sistema de arquivos
crossmnt	Permite que os diretórios abaixo do diretório marcado também sejam acessíveis

É necessário ainda configurar o NFS para que ele utilize o Kerberos. Para tanto é necessário alterar o arquivo **/etc/default/nfs-common**, mostrado na Figura 5.8. Nele, as opções NEED_IDMPAD e NEED_GSSD são necessárias para o uso do NFSv4 e do Kerberos, respectivamente.

```
NEED_STATD=
STATDOPTS=
NEED_IDMAPD=yes
NEED_GSSD=yes
```

Figura 5.8: Arquivo /etc/default/nfs-common

Outro arquivo que precisa ser alterado é o **/etc/default/nfs-kernel-server**. Nele a opção NEED_SVCGSSD deve ser modificada de acordo com a Figura 5.9, para habilitar o suporte à *exports* do Kerberos.

```
RPCNFSDCOUNT=10
RPCNFSDPRIORITY=0
RPCMOUNTDOPTS=
NEED_SVCGSSD=yes
RPCSVCGSSDOPTS=
```

Figura 5.9: Arquivo /etc/default/nfs-kernel-server

Por fim, o domínio deve ser informado no parâmetro Domain, do arquivo **/etc/idmapd.conf**, conforme a Figura 5.10.

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = MEUDOMINIO.ORG

[Mapping]

Nobody-User = nobody
Nobody-Group = nogroup
```

Figura 5.10: Arquivo /etc/idmapd.conf

A versão do NFS para Ubuntu 10.04 suporta apenas o método de encriptação DES, que não está habilitado por padrão. Portanto, esse suporte deve ser habilitado alterando-se o arquivo **/etc/krb5.conf** como mostrado na Figura 5.11.

```
[libdefaults]
  allow_weak_crypto = true
  default_tgs_enctypes = des-cbc-crc
  default_tkt_enctypes = des-cbc-crc
```

Figura 5.11: Arquivo /etc/krb5.conf

A seguir, é preciso criar os *principals* para o servidor e para o cliente. Ambos serão criados no próprio servidor, mas o arquivo **pc01.keytab** deve ser copiado para o diretório /etc no cliente e nele deve ser renomeado para **krb5.keytab**. A Figura 5.12 mostra a criação dos *principals* e a Figura 5.13 mostra a saída do comando `klist -k` no servidor e no cliente, respectivamente.

```
$ sudo kadmin.local -q "addprinc -randkey nfs/nfs.meudominio.org"
$ sudo kadmin.local -q "ktadd -e des-cbc-crc:normal nfs/nfs.meudominio.org"

$ sudo kadmin.local -q "addprinc -randkey nfs/pc01.meudominio.org"
$ sudo kadmin.local -q "ktadd -e des-cbc-crc:normal -k pc01.keytab nfs/pc01.meudominio.org"
```

Figura 5.12: Adicionando *Principals* e Gerando Chave para o Cliente

```
$ sudo klist -k
Keytab name: WRFILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   2 nfs/nfs.meudominio.org@MEUDOMINIO.ORG

$ sudo klist -k
Keytab name: WRFILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   3 nfs/pc01.meudominio.org@MEUDOMINIO.ORG
```

Figura 5.13: Saída do Comando klist -k no Servidor e no Cliente

A configuração do servidor está pronta, e a sua funcionalidade pode ser testada usando-se o comando mostrado na Figura 5.14.

```
$ sudo mount -t nfs4 -o sec=krb5 nfs.meudominio.org:/home/compartilhado /mnt
```

Figura 5.14: Testando o Servidor NFS

5.2.2 Configurações nos Clientes

Os pacotes mostrados na Figura 5.15 são necessários para a configuração dos clientes.

```
$ apt-get install nfs-common krb5-user
```

Figura 5.15: Instalando Pacotes NFS no Cliente

Assim como no servidor, nos clientes é preciso configurar o Kerberos e o NFS alterando-se o arquivo **/etc/krb5.conf**, de acordo com a Figura 5.16. Os parâmetros alterados foram os mesmos do servidor, com exceção do default_realm, que deve indicar o reino Kerberos.

O arquivo **/etc/default/nfs-common** também é muito similar ao do servidor, com a exceção do parâmetro RPCGSSDOPTS, que aumenta o detalhamento dos arquivos de log. O arquivo na íntegra é mostrado na Figura 5.17.

```
[libdefaults]
  default_realm = MEUDOMINIO.ORG
  allow_weak_crypto = true
  default_tgs_enctypes = des-cbc-crc
  default_tkt_enctypes = des-cbc-crc
```

Figura 5.16: Arquivo /etc/krb5.conf no Cliente NFS

```
NEED_STATD=
STATDOPTS=
NEED_IDMAPD=yes
NEED_GSSD=yes
RPCGSSDOPTS="-vvv"
```

Figura 5.17: Arquivo /etc/default/nfs-common no Cliente NFS

Por fim, o arquivo **/etc/idmapd.conf** deve ficar igual ao seu correspondente no servidor, como mostra a Figura 5.18.

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = MEUDOMINIO.ORG

[Mapping]

Nobody-User = nobody
Nobody-Group = nogroup
```

Figura 5.18: Arquivo /etc/idmapd.conf no Cliente NFS

Terminadas as configurações, é recomendado que o cliente seja reiniciado. Após a reinicialização, o diretório pode ser montado no cliente da mesma forma usada no servidor.

5.3 Comentários Finais

O objetivo desse capítulo foi apresentar opções práticas de aplicação de um sistema Kerberos com *backend* LDAP.

Foi mostrada a autenticação simples de um usuário UNIX e também o compartilhamento de arquivos usando NFSv4 com autenticação.

O capítulo a seguir encerra este trabalho, apresenta algumas considerações finais e propostas para trabalhos futuros.

Capítulo 6

Considerações Finais

Visando à centralização da administração, à melhor conveniência do usuário e à redução dos problemas de segurança em redes de computadores, esse trabalho apresentou como solução a implementação de um sistema Kerberos com *backend* LDAP.

O objetivo principal foi apresentar e implementar os dois serviços, primeiramente de maneira separada e posteriormente integrá-los. Tanto o Kerberos quanto o LDAP foram explorados de maneira superficial, uma vez que são serviços muito complexos e, ainda que o trabalho fosse dedicado inteiramente a qualquer um deles, dificilmente o assunto seria contemplado em sua totalidade.

No entanto, embora as obras contidas na referência bibliográfica desse trabalho possam oferecer uma visão mais aprofundada em relação ao Kerberos e ao LDAP, na maioria delas a integração é apenas citada ou exposta como possibilidade. As informações, configurações e argumentos contidos no trabalho são fruto de análise, pesquisa e compilação a partir de diversas fontes.

A partir desse estudo, é possível concluir que, embora seja complexa e trabalhosa, a implementação de um servidor Kerberos com *backend* LDAP é viável e pode ser uma grande ferramenta no auxílio ao administrador da rede. Serviços "kerberizados"se tornam muito mais seguros e confiáveis, ao mesmo tempo que a organização da estrutura da rede se torna mais clara e organizada.

A integração apresentada nesse trabalho foi completa, e é totalmente funcional, no entanto ela é apenas o começo de uma vasta gama de possibilidades. O foco desse trabalho foi apresentar os dois serviços e o seu potencial de trabalhar em conjunto, entretanto ambos possuem muitas outras possibilidades a serem ex-

ploradas e problemas a serem evitados. As seções a seguir apresentam comentários sobre alguns problemas que podem afetar o desempenho da solução implementada e ideias para expansão desse trabalho.

6.1 Problemas de Segurança

O uso de um sistema Kerberos apoiado em uma base LDAP, por si só, não garante total segurança na rede. Embora ele adicione uma camada extra de segurança, através de autenticação e criptografia, o Kerberos não pode prevenir problemas de *hardware*, *software* e do principal ponto de falha em uma rede de computadores: o usuário.

Servidores são máquinas físicas, e como tal, estão sujeitas a defeitos e mal funcionamentos. Por isso é importante que haja redundância nos principais serviços de uma rede. O servidor OpenLDAP será responsável por armazenar todas as informações dos usuários da rede, e o Kerberos fará toda a autenticação. Se algum desses serviços parar devido à falha de *hardware*, toda a rede será diretamente comprometida. A solução para isso é a replicação tanto da base de dados quanto do servidor kdc, preferencialmente em máquinas fisicamente distantes.

Outro ponto a ser observado é o aspecto do *software*. Qualquer máquina ligada à Internet está sujeita a ataques. De nada adianta um servidor bem configurado e redundante se ele pode ser facilmente acessado por indivíduos mal intencionados. A configuração de um bom *firewall*, fechando-se todas as portas e desabilitando todos os serviços não essenciais, é necessária para blindar os servidores.

Por último, e não menos importante, o maior inimigo pode estar devidamente credenciado e identificado. O usuário da rede local não deve ter acesso físico aos servidores e deve ser conscientizado de suas responsabilidades em caso de perda ou divulgação de sua própria senha de acesso.

6.2 Problemas na Estrutura da Rede

Como já citado anteriormente, o uso de um servidor DNS na rede facilita em muito a configuração do Kerberos. Caso não haja tal serviço, será necessário configurar todos os clientes de forma que eles conheçam o endereço IP dos servidores kdc e LDAP. A heterogeneidade entre sistemas operacionais também pode ser um pro-

blema para o administrador da rede. Caso hajam múltiplas plataformas em uso, a integração com serviços adicionais como o Samba pode ser necessária.

6.3 Propostas para Trabalhos Futuros

Como pôde ser observado nesse capítulo, a implementação do Kerberos com *backend* LDAP ainda necessita de vários ajustes. A seguir serão listadas as possibilidades de continuação desse trabalho:

- configuração de um servidor Kerberos secundário;
- replicação da base de dados LDAP;
- integração do Kerberos a outros serviços de rede (Proxy, SMTP, etc);
- integração ao Samba e suporte a diferentes sistemas operacionais.

Referências Bibliográficas

CARTER, G. *LDAP System Administration*. 1. ed. United States of America: O'Reilly, 2003.

CHADWICK, D. W. *Understanding X.500 (The Directory)*. 1. ed. United States of America: International Thompson Publishing, 1996.

FERGUSON, N.; SCHNEIER, B. *Practical Criptography*. 1. ed. United States of America: Wiley, 2003.

GARMAN, J. *Kerberos: The Definitive Guide*. 1. ed. United States of America: O'Reilly, 2003.

JORDÃO, L. d. B. *Uso do Protocolo de Autenticação Kerberos em Redes Linux*. Lavras, 2005. Disponível em: <http://www.ginux.ufla.br/files/mono-LeonardoJordao.pdf>.

KHOL, J.; NEUMANN, C. *The Kerberos Network Authentication Service (V5)*. [S.l.], 2005. Disponível em: <http://www.ietf.org/rfc/rfc1510.txt>.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 5. ed. United States of America: Pearson Education, 2010.

MACHADO, E. S.; JUNIOR, F. d. S. M. *Autenticação Integrada Baseada em Serviço de Diretório LDAP*. São Paulo, 2006. Disponível em: <http://www.linux-.ime.usp.br/~cef/mac499-06/monografias/erich/html/index.html>.

MORIMOTO, C. E. *Servidores Linux Guia Prático*. 1. ed. Porto Alegre: Sul Editores, 2008.

NEMETH, E.; SNYDER, G.; HEIN, T. R. *Manual Completo do Linux, Guia do Administrador*. 2. ed. São Paulo: Pearson Education do Brasil, 2009.

SINCLAIR, J. T.; MERKOW, M. *Thin Clients Clearly Explained*. 1. ed. United States of America: Morgan Kaufmann, 1999.

SUNGAILA, M. *Autenticação Centralizada com OpenLDAP*. 1. ed. São Paulo: Novatec, 2008.

TANENBAUM, A. S. *Redes de Computadores*. 4. ed. Rio de Janeiro: Editora Campus, 2003.

TRIGO, C. H. *Openldap - Uma Abordagem Integrada*. 1. ed. São Paulo: Novatec, 2007.

UCHÔA, J. Q. *Gerenciamento de Sistemas Linux*. 3. ed. Lavras: UFLA/FAEPE, 2007.

UCHÔA, J. Q.; SICA, F. C.; SIMEONE, L. E. *Serviços de Redes em Linux*. 2. ed. Lavras: UFLA/FAEPE, 2004.